High precision camera calibration

Zhongwei Tang

École Normale Supérieure de Cachan

Thèse

pour obtenir le grade de

DOCTEUR DE L'ÉCOLE NORMALE SUPÉRIEURE DE CACHAN

Spécialité : Mathématiques Appliquées

Présentée par

Zhongwei TANG

Calibration de Caméra à Haute Précision

Directeurs de thèse: Pascal MONASSE et Jean-Michel MOREL

préparée à CMLA, ENS-Cachan

soutenue le 1 juillet, 2011

Jury :

Rapporteurs:	Luis ALVAREZ	-	Universidad de Las Palmas, Espagne
	Jean-Marc LAVEST	-	Université d'Auvergne
	Peter STURM	-	INRIA Grenoble - Rhône-Alpes
Directeurs:	Pascal MONASSE	-	École des Ponts ParisTech
	Jean-Michel MOREL	-	École Normale Supérieure de Cachan
Examinateurs:	Marc PIERROT-DESEILLIGNY	-	Institut Géographique National
	Guillermo SAPIRO	-	University of Minnesota, Étas-Unis

Remerciements

J'ai effectuée ce travail de thèse au Centre de Mathématiques et de Leurs Applications (CMLA), à l'Ecole Normale Supérieure de Cachan (ENS-Cachan). Je voudrais d'abord remercier Pascal Monasse et Jean-Michel Morel, qui m'ont dirigé au cours de ces années. Je n'aurais pas pu mener ce travail à bien sans leur grande disponibilité et patience. Leurs grandes qualités scientifiques et humaines m'ont beaucoup aidé à comprendre, analyser et résoudre les problèmes que j'ai recontrés tout au long de ma thèse. Pour tout cela, je leur en suis très reconnaissant. Il ne faut pas que j'oublie mon "directeur de thèse virtuel": Rafael Grompone von Gioi. Bien que il ne soit pas mon directeur de thèse selon la feuille de l'inscription, il est toujours là et prêt à m'aider. Je trouve que sa philosophie est tout utile pour la science et a orienté mes recherches vers un chemin tout logique. Sans ses aides et son logiciel magique, la thèse aurait été beaucoup plus difficile. Pour tout cela, je lui en suis très reconnaissant.

Un élément important dans cette thèse est l'interaction active avec les autres chercheurs. Pour la partie de "burst denoising", je remercie beaucoup Antoni Buades et Yifei Lou pour leur collaboration. Pour la partie de calibration, je tiens ma gratitude à Jean-Marc Lavest pour son accueil chaleureux à LASMEA et nous passer généreusement son code de calibration.

Je suis très sensible à l'honneur que m'ont fait les membres du jury de thèse: Luis Alvarez, Jean-Marc Lavest et Peter Sturm qui ont accepté de rapporter ma thèse, Guillermo Sapiro et Marc Pierrot-Deseilligny qui ont accepté de faire partie du jury. Je les remercie d'avoir apporté un intérêt à mon travail. Ce mémoire a beaucoup bénéficié des leurs commentaires et questions.

Je tiens également à exprimer toute ma gratitude à Véronique Almadovar, Micheline Brunetti, Sandra Doucet, Virginie Pauchont et Carine Saint-Prix pour leur efficacité et la bonne humeur qu'elles répandent dans le laboratoire.

Tous mes remerciements vont aux membres passés et présents de cette équipe qui m'ont soutenu tout au long de cette thèse. Merci à Adina, Aude, Bruno, Eric, Ives, Julien, Mauricio, Neus, Nicolas, Rafael et Yohann. Merci à tous les thésards, post-docs et visiteurs avec lesquels j'ai pris le déjeuner génial au Crous, le café du Pavillons des Jardins sur la pelouse et le repas gastronomiques au restaurant le soir: Adina, Alex, Aude, Ayman, Benjaman, Bruno, Eric, Frédéric, Frédérique, Gaël, Ives, Julien, Mao Yu, Marc, Mauricio, Miguel, Neus, Nicolas C., Nicolas M., Romain, Sadd, Yifei et Yohann.

Enfin, merci à tous mes amis en France qui me tiennent compagnie pendant certain périodes longues ou courts, Xu Xiao-qian, Chen Jian, Chen Ying-Ying, Gao Jia-Yi, Liu Jia-gui, Luo Ling, Pen Ya-Xin, Qu xing-tai, Ren Hua, Ruan Yi-bing, Shi Fei-fei, Sun Zhe, Wang Yao, Wu Ting, Wu Xiao, Yang Fan, Yang Yu-heng, Yi Hua, Yi Zhen-Zhen, Yu Guo-Shen, Yu Yong, Zeng tie-Yong, Zhang Fang-zhou, Zhang Min, Zheng Ding-wei, Zhou Qing pour leur

humour, leur amitié, leurs conseils et soutiens de chaque instant lors des moments difficiles et les bon moments passés ensemble.

Mes principaux remerciements s'adressent finalement à mes parents. Je vous remercie pour votre confiance que vous m'avez témoignée durant ce long parcours d'études. Merci de m'avoir donné la vie assez belle, de m'avoir aidée à croire en moi pendant toutes ces années d'études.

Contents

Abstract

The thesis focuses on precision aspects of 3D reconstruction with a particular emphasis on camera distortion correction. The causes of imprecisions in stereoscopy can be found at any step of the chain. The imprecision caused in a certain step will make useless the precision gained in the previous steps, then be propagated, amplified or mixed with errors in the following steps, finally leading to an imprecise 3D reconstruction. It seems impossible to directly improve the overall precision of a reconstruction chain leading to final imprecise 3D data. The appropriate approach to obtain a precise 3D model is to study the precision of every component.

A maximal attention is paid to the camera calibration for three reasons. First, it is often the first component in the chain. Second, it is by itself already a complicated system containing many unknown parameters. Third, the intrinsic parameters of a camera only need to be calibrated once, depending on the camera configuration (and at constant temperature).

The camera calibration problem is supposed to have been solved since years. Nevertheless, calibration methods and models that were valid for past precision requirements are becoming unsatisfying for new digital cameras permitting a higher precision. In our experiments, we regularly observed that current global camera methods can leave behind a residual distortion error as big as one pixel, which can lead to distorted reconstructed scenes. We propose two methods in the thesis to correct the distortion with a far higher precision. With an objective evaluation tool, it will be shown that the finally achievable correction precision is about 0.02 pixels. This value measures the average deviation of an observed straight line crossing the image domain from its perfectly straight regression line.

High precision is also needed or desired for other image processing tasks crucial in 3D, like image registration. In contrast to the advance in the invariance of feature detectors, the matching precision has not been studied carefully. We analyze the SIFT method (Scale-invariant feature transform) and evaluate its matching precision. It will be shown that by some simple modifications in the SIFT scale space, the matching precision can be improved to be about 0.05 pixels on synthetic tests. A more realistic algorithm is also proposed to increase the registration precision for two real images when it is assumed that their transformation is locally smooth.

A multiple-image denoising method, called “burst denoising”, is proposed to take advantage of precise image registration to estimate and remove the noise at the same time. This method produces an accurate noise curve, which can be used to guide the denoising by the simple averaging and classic block matching method. “burst denoising” is particularly powerful to recover fine non-periodic textured part in images, even compared to the best state of the art denoising method.

Résumé

Cette thèse se concentre sur les aspects de précision de la reconstruction 3D avec un accent particulier sur la correction de distorsion. La cause de l'imprécision dans la stéréoscopie peut être trouvée à toute étape de la chaîne. L'imprécision due à une certaine étape rend inutile la précision acquise dans les étapes précédentes, puis peut se propage, se amplifie ou se mélange avec les erreurs dans les étapes suivantes, conduisant finalement à une reconstruction 3D imprécise. Il semble impossible d'améliorer directement la précision globale d'une chaîne de reconstruction 3D qui conduit à données 3D imprécises. L'approche plus approprié pour obtenir un modèle 3D précis est d'étudier la précision de chaque composant.

Une attention maximale est portée à la calibration de l'appareil photo pour trois raisons. Premièrement, il est souvent le premier composant dans la chaîne. Deuxièmement, il est en soi déjà un système compliqué contenant de nombreux paramètres inconnus. Troisièmement, il suffit de calibrer les paramètres intrinsèques d'un appareil photo une fois, en fonction de la configuration de l'appareil photo (et à température constante).

Le problème de calibration de l'appareil photo est censé d'avoir été résolu depuis des années. Néanmoins, méthodes et modèles de calibration qui étaient valables pour les exigences de précision autrefois deviennent insatisfaisants pour les nouveaux appareils photo numériques permettant une plus grande précision. Dans nos expériences, nous avons régulièrement observé que les méthodes globales actuelles peuvent laisser une distorsion résiduelle en ordre d'un pixel, ce qui peut conduire à des distorsions dans les scènes reconstruites. Nous proposons deux méthodes dans la thèse pour corriger la distorsion, avec une précision beaucoup plus élevée. Avec un outil d'évaluation objective, nous montrons que la précision de correction finalement réalisable est d'environ $0,02$ pixels. Cette valeur représente l'écart moyen d'une ligne droite observée traversant le domaine de l'image à sa ligne de régression parfaitement droite.

La haute précision est également nécessaire ou souhaitée pour d'autres tâches de traitement d'images cruciales en 3D, comme l'enregistrement des images. Contrairement au progrès dans l'invariance de détecteurs des point d'intérêt, la précision de matchings n'a pas été étudiée avec soin. Nous analysons la méthode SIFT (Scale-Invariant Feature Transform) et d'évaluer sa précision de matchings. Il montre que par quelques modifications simples dans l'espace d'échelle de SIFT, la précision de matchings peut être améliorée à être d'environ $0,05$ pixels sur des tests synthétiques. Un algorithme plus réaliste est également proposé pour augmenter la précision de matchings pour deux images réelles quand la transformation entre elles est localement lisse.

Une méthode de débruitage avec une série des images, appelée "burst denoising", est proposée pour profiter des matchings précis pour estimer et enlever le bruit en même temps. Cette

méthode produit une courbe de bruit précise, qui peut être utilisée pour guider le débruitage par la moyenne simple et la méthode classique. “burst denoising” est particulièrement puissant pour restaurer la partie fine texturée non-périodique dans les images, même par rapport aux meilleures méthodes de débruitage de l’état de l’art.

Chapter 1

Introduction

The thesis focuses on precision aspects of 3D reconstruction with a particular emphasis on camera distortion correction. The causes of imprecisions in stereoscopy can be found at any step of the chain. The imprecision caused in a certain step will make useless the precision gained in the previous steps, then be propagated, amplified or mixed with errors in the following steps, finally leading to an imprecise 3D reconstruction. It seems impossible to directly improve the overall precision of a reconstruction chain leading to final imprecise 3D data. The appropriate approach to obtain a precise 3D model is to study the precision of every component.

We shall pay a maximal attention to the camera calibration for three reasons. First, it is often the first component in the chain. Second, it is by itself already a complicated system containing many unknown parameters. Third, the intrinsic parameters of a camera only need to be calibrated once, depending on the camera configuration (and at constant temperature).

The camera calibration problem is supposed to have been solved since years. Nevertheless, calibration methods and models that were valid for past precision requirements are becoming unsatisfying for new digital cameras permitting a higher precision. In our experiments, we regularly observed that current global camera methods can leave behind a residual distortion error as big as one pixel, which can lead to distorted reconstructed scenes. We propose two methods in the thesis to correct the distortion with a far higher precision. With an objective evaluation tool, it will be shown that the finally achievable correction precision is about 0.02 pixels. This value measures the average deviation of an observed straight line crossing the image domain from its perfectly straight regression line [53, 3, 4, 89].

High precision is also needed or desired for other image processing tasks crucial in 3D, like image registration. In contrast to the advance in the invariance of feature detectors, the matching precision has not been studied carefully. We analyze the SIFT method (Scale-invariant feature transform) [117] and evaluate its matching precision. It will be shown that by some simple modifications in the SIFT scale space, the matching precision can be improved to be about 0.05 pixels on synthetic tests. A more realistic algorithm is also proposed to increase the registration precision for two real images when it is assumed that their transformation is locally smooth. A multiple-image denoising method, called “burst denoising”, is finally proposed to take advantage of precise image registration.

1.1 Context

This thesis belongs to the research project CALLISTO (Calibration en vision stéréo par méthodes statistiques) sponsored by ANR (Agence Nationale de la Recherche), whose final aim is to reconstruct 3D scenes with high precision. This project relies on the collaboration between several universities and *Grandes Ecoles*: CMLA-ENS-Cachan, IMAGINE-ENPC, MAP5-Paris 6 and LCTI-Télécom Paris. Even though this project mainly processes images taken by consumer digital cameras, a significant part of the results can be used in the satellite imaging context. The work thus helped to some extent the MISS (Mathématiques de l'imageries stéréscopique spatiale) project in collaboration with CNES (Centre National d'Etudes Spatiales), whose aim it is to reconstruct a completely controlled and reliable 3D terrain models from two images taken almost simultaneously by air-bone cameras or satellite cameras.

The 3D scene reconstruction chain from two images, which is very relevant in space imaging, can be roughly divided into five components [65, 89, 71]: camera calibration, image rectification, dense image registration, 3D scene reconstruction and merging (Fig. 1.1). Some components can be replaced by other techniques to make the chain more adapted to specific applications [105, 74].

The camera calibration is the first step in the chain and thus plays a very important role. It consists of a camera internal/external parameters calibration and of a distortion model estimation [65, 89, 71, 115, 20]. By camera internal parameters we mean the intrinsic parameters of the camera, like its aspect ratio, focal length, principal point, lens distortion parameters. The camera external parameters mean the camera orientation and position in a fixed world coordinate. The distortion model is a planar deformation field modeling the geometric deviation of the real camera from a pinhole camera.

Image stereo rectification is an auxiliary step for dense image registration [89]. Starting from a stereo pair of images, it virtually rotates the cameras which took the photographs around their respective optical centers so that the two cameras planes become co-planar, and that their x-axes become parallel to the baseline. This generates two images whose corresponding epipolar lines coincide and are parallel to the x-axis of the image. A pair of rectified images is helpful for dense stereo matching algorithms. It restricts the search domain for each matching to a line parallel to the x-axis. Due to the redundant degrees of freedom, the solution to rectification is not unique and actually can lead to undesirable distortions or be stuck in a local minimum of the distortion function. This motivated us to propose a robust three-step image rectification in Chapter 7.

Dense image registration is the thesis subject of Neus Sabater [154], whose aim was to find dense correspondences between two images. Traditional "large B/H" (large baseline/height) stereo is a difficult problem because of the varying imaging conditions when taking the images, like a geometric distortion depending on the viewing angle, a non-linear lens distortion of the camera, changing lighting condition, non-static scenes, occlusions, etc [120, 71]. As the mentioned thesis shows, these problems can be to some extent alleviated by using a pair of images with "low B/H" (low baseline/height), which are taken by a satellite almost simultaneously. But this raises a higher demand on the precision of the correspondences.

Once a camera is precisely calibrated and a pair of images is accurately and reliably registered, a 3D model of a scene can be easily reconstructed on the same order of precision

with some classic methods, up to a 3D similarity transformation [89, 120, 71, 115, 65]. Of course one pair of images only permits us to reconstruct a partial 3D model. To have a more complete model, it is necessary to take photos with different angles of view around the 3D scene. By merging many pair-wise partial 3D models, a more complete 3D model can be obtained.

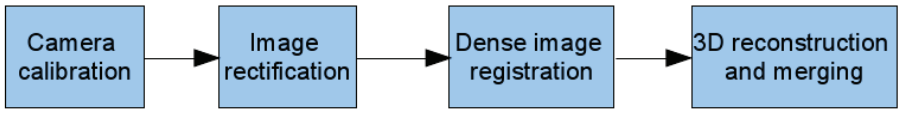


Figure 1.1: The 3D reconstruction chain.

1.2 Background

The 3D scene reconstruction preexisted to computer vision. Before the "computer" and digital camera came into history, at the end of the seventies, scene reconstruction was already a classic problem in photogrammetry, where it had the different name "stereo-photogrammetry" [138, 178]. Its aim was always to determine the geometric properties of objects from photographs. At that time, more attention was paid to the methods in optics and metrology, due to the lack of computational power. The distances and angles were directly measured manually from photographs, objects in scene and cameras separated by a fixed baseline. The precision is not ensured in the measurement, which can lead to inaccurate final result. The lack of imprecision of manual work limited the practical application of early photogrammetry techniques.

With the advent of digital cameras and high-performance computers, the 3D reconstruction became more mature with fewer or no manual measurement. The prompt perhaps came from the remarkable advance in camera calibration techniques. The ground breaking numerical calibration methods, Guang and Long's automatic calibration method [82], Sturm's plane-based calibration [163, 83], Zhang's flexible pattern based method [191], the Lavest *et al.* method [95], the Devernay and Faugeras plumb-line method [53], Hartley's pattern-free method [85], Bouguet's dual-space geometry based method [21, 22], Bouguet's calibration toolbox [20] or Pierrot-Deseilligny's automatic bundle adjustment calibration software [147], make it possible to calibrate a camera and reconstruct a 3D scene on site with little human intervention.

Another possibility to obtain 3D reconstructions is to use a 3D laser or LIDAR (LIght Detection And Ranging) [23]. Unlike the multi-view geometric methods, this kind of methods is active. The position of object in space is measured by the time delay between the transmission of a pulse and the detection of the reflected signal. Thus a dense 3D points can be directly obtained by some numerical post-processing, mainly filtering and merging. (An intermediate technique exists with triangulation scanners, where a laser comb is coupled with a camera). It is commonly admitted that 3D stereo reconstruction from images is not as precise as the result obtained by a 3D laser scanner. A high quality 3D triangulation scanner can for example provide 3D point clouds with a precision about 20 μm at a 40 cm distance. This cannot

be achieved by state-of-art image-based 3D stereo reconstruction algorithm and camera technology. Yet, accurate 3D laser scanners are expensive sophisticated machines, which have to be set up carefully and cannot be easily transported for on site 3D reconstruction tasks, while a camera, even a reflex camera of good quality, costs almost nothing compared to a scanner machine and can be flexibly transported anywhere to take photos (Fig. 1.2). In addition, it is not feasible to install a huge laser scanner on a satellite to perform 3D reconstruction. So it is still of interest to use camera photos to reconstruct 3D scenes if the precision can achieve or surpass that of 3D scanners.

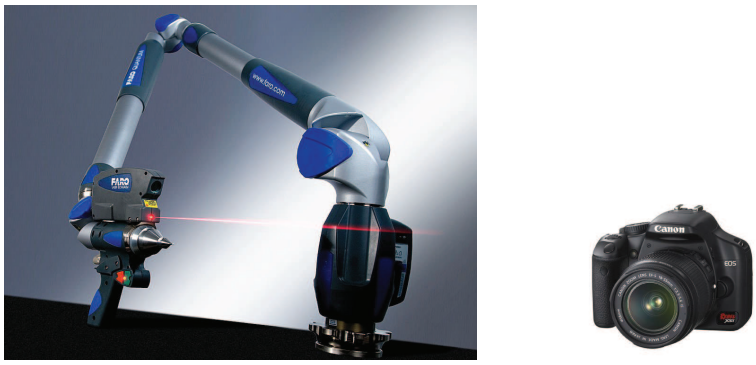


Figure 1.2: Left: 3D laser scanner. Right: Canon EOS Reflex camera.

1.3 The Precision Challenge

To have a precise final 3D reconstruction from 2D photos, each component should produce a precise result. This gives the first importance to the camera calibration because its imprecision will be inevitably propagated to the other components and usually cannot be corrected later. With a precise camera calibration, it is already sufficient to determine the relative position of two cameras and reconstruct a sparse model of 3D scene from the correspondences between two images. The common camera model has the form

$$\mathbf{P} = \mathcal{D}\mathbf{K}\mathbf{R}[\mathbf{I}| - \mathbf{C}] \tag{1.1}$$

with $\mathcal{D}$ non-linear operator of lens distortion, $\mathbf{K}$ calibration matrix, $\mathbf{R}$ camera orientation matrix and vector $\mathbf{C}$ the camera optical center in a fixed world frame. Given a 3D point, it is first transformed into the camera-based frame by the translation $-\mathbf{C}$ then the rotation $\mathbf{R}$. Then it is projected onto the image plane by $\mathbf{K}$, followed by the non-linear lens distortion represented by $\mathcal{D}$. According to this model, camera calibration consists of two parts: internal parameters calibration ($\mathcal{D}$ and $\mathbf{K}$) and external parameters calibration ($\mathbf{R}$ and $\mathbf{C}$). The internal calibration is to retrieve the intrinsic parameters of camera, which should be constant once the camera has fixed its configuration. But in the experiments, it so happens that the internal parameters vary from one experiment to the other, even if the same camera was used with a fixed configuration. This means that the internal calibration is not reusable for the other data sets. Similarly, for the external calibration, the camera position and orientation also varies by using two data sets sharing a common camera position. This led us to think

over again the whole camera calibration system. The conjecture about the phenomenon can be two-fold:

1. the distortion model cannot capture the real physical aspect of real distortion
2. the error in external calibration for $\mathbf{R}$ and $\mathbf{C}$ compensates the error in internal calibration for $\mathcal{D}$ and $\mathbf{K}$. So the whole camera model can be precise in the sense that the observed point is close to the point computed by the camera model. But none of the components ($\mathcal{D}$, $\mathbf{K}$, $\mathbf{R}$, $\mathbf{C}$) is precise.

This is in fact the common drawback of many global camera calibration methods, which perform the internal and external calibration together. Typical global calibration algorithms were tested to verify the problem. In particular, the state of the art Lavest *et al.* algorithm [95] shows a very small re-projection error of about 0.02 pixels, which confirms the precision of the global camera model. But the residual distortion error can be 10 times bigger. Since the Lavest *et al.* algorithm corrects also the non-flatness of the pattern and estimates its shape, this 10 factor can only be explained by an error compensation between the various estimates. In our opinion this error compensation cannot be corrected in the framework of a global method. This is why we chose to address the distortion correction separately, actually as the preliminary step to any calibration. We shall explain why and how this leads to use a slightly new and probably cleaner camera model, abstracted from any physical contingency, which we call the *virtual pinhole* camera.

1.4 The Virtual Pinhole Camera

It seems to us that an important contribution of this thesis is the proposition and use of a virtual pinhole camera, which succeeds in avoiding the model inaccuracies of physical models. The first aim of camera calibration is to recover a pinhole camera by correcting the distortion. But as explained, due to the error compensation, it is risky to use global calibration methods to correct lens distortion. The error compensation can be avoided by separating the distortion correction from the global calibration. So the camera calibration is decomposed into two steps. The first step is to correct lens distortion to obtain a pinhole camera; the second step is to calibrate the pinhole camera. As for the pinhole camera, the fundamental theorem must be cited [89, 53]:

Theorem 1 *A camera follows the pinhole model if and only if the projection of every line (not passing through the optical center) in space onto the camera is a line.*

This theorem can be understood in three different ways. First, the distortion can be corrected by rectifying the distorted lines in image. Second, the distortion correction can be evaluated by measuring the straightness of corrected lines in image. Third, the pinhole camera is not unique because a straight line in the image remains straight under any 2D homography. This means that the distortion can be corrected only up to an arbitrary homography. Assume the arbitrary homography is $\mathbf{H}$, then the estimated distortion is $\tilde{\mathcal{D}} = \mathcal{D}\mathbf{H}$ with $\mathcal{D}$ the absolute distortion introduced by camera lens system. By applying the inverse of $\tilde{\mathcal{D}}$ on camera, we obtain a new camera

$$\tilde{\mathbf{P}} = \tilde{\mathcal{D}}^{-1}\mathbf{P} = \mathbf{H}^{-1}\mathcal{D}^{-1}\mathcal{D}\mathbf{K}\mathbf{R}[\mathbf{I}| \ -\mathbf{C}] = \mathbf{H}^{-1}\mathbf{K}\mathbf{R}[\mathbf{I}| \ -\mathbf{C}]. \tag{1.2}$$

$\mathbf{H}$, $\mathbf{K}$ being invertible, the decomposition $\mathbf{H}^{-1}\mathbf{K} = \tilde{\mathbf{K}}\mathbf{R}'$ is unique by QR decomposition with the constraint that $\tilde{\mathbf{K}}$ is an upper-triangle 3×3 matrix and $\mathbf{R}'$ is a 3×3 rotation matrix. The new camera $\tilde{\mathbf{P}}$ becomes $\tilde{\mathbf{P}} = \tilde{\mathbf{K}}\mathbf{R}'\mathbf{R}[\mathbf{I}| \ -\mathbf{C}] = \tilde{\mathbf{K}}\tilde{\mathbf{R}}[\mathbf{I}| \ -\mathbf{C}]$. We call it the *virtual (or mathematic) camera after distortion correction* because the calibration matrix $\tilde{\mathbf{K}}$ and the rotation matrix $\tilde{\mathbf{R}}$ do not match the physics of the actual camera, but yield a virtual pinhole camera that can be used to the very same purposes. Indeed, consider several positions of the physical camera inducing as many camera models $\mathbf{P}_i = \mathcal{D}\mathbf{K}\mathbf{R}_i[\mathbf{I}| \ -\mathbf{C}_i]$. Applying the correction $\tilde{\mathcal{D}}^{-1}$ to all images obtained from these camera positions yields virtual pinhole cameras $\tilde{\mathbf{P}}_i = \tilde{\mathbf{K}}\tilde{\mathbf{R}}_i[\mathbf{I}| \ -\mathbf{C}_i]$, which maintains the same relative orientations: $\tilde{\mathbf{R}}_i^{-1}\tilde{\mathbf{R}}_j = \mathbf{R}_i^{-1}\mathbf{R}_j$. From these cameras the whole 3D scene can be reconstructed by standard methods, up to a 3D similarity.

1.5 Image Registration and Denoising

Invariance and precision are two key problems in image registration. Invariance means whether an image matching algorithm can find reliable correspondences under critical geometric or photometric transformations. Precision means whether the matchings between two images are precise. The two problems are crucial for the success of many applications, like super-resolution, image mosaicing, camera calibration, etc. Many efforts have been recently dedicated to obtaining feature detectors more invariant to geometry or photometric transformations, while the matching precision of feature points is always considered enough and has not been carefully studied. In fact, some state-of-art feature detectors combining a robust descriptor give enough invariance for many applications. So it seems to be time to evaluate and improve the matching precision. We took interest in the matching precision of SIFT [117], the most popular scale-invariant feature detector. The matching precision of the SIFT method is studied and improved. We show that the matching precision can achieve a precision better than 0.05 pixel if the transformation between the two matched images is locally smooth.

Precise image registration inspired us a new image denoising algorithm, which we called "burst denoising". Basically, this algorithm aligns a burst of images to a reference image and performs the average operation to reduce the noise level. This method is extended to be a mixed algorithm by combining the block denoising when the two images are not related by a rigid transformations.

1.6 The Thesis Chapter by Chapter

Chapter 2: Camera Model and Projective Geometry

Many algorithms in multi-view geometry are based on the assumption that the camera is ideal pinhole. But in practice, the camera is deviated from a pinhole model by lens distortion. In this chapter, some basic concepts about pinhole camera model and distortion model are introduced. A typical bundle adjustment camera calibration method is also explained in detail to show the problems we shall meet in distortion correction.

Once the distortion is removed and the camera becomes pinhole, projective geometry is the perfect tool to solve many problems in multi-view geometry, like image rectification and mosaicing. The geometric constraint becomes more complicated when the number of views

increases. Here we concentrate on the simplest two-view geometry because it is difficult for three or more images to share enough stable feature points in practice. The relation between corresponding points in two views is described by epipolar geometry. From the algebraic viewpoint, the epipolar geometry is coded by a 3×3 matrix, called fundamental matrix, $\mathbf{F}$. The fundamental matrix only depends on the relative position of two cameras and the intrinsic parameters of cameras, but not on the 3D scene. Two important observations of epipolar geometry are:

- Given one point $\mathbf{x}$ in the left image, its corresponding point in the right image must be on the line called epipolar line, which can be explicitly computed as $\mathbf{Fx}$. This provides a necessary condition to test whether two points correspond to the same 3D point, see Fig. 1.3.
- Given a set of corresponding points in two images, the 3D scene can be reconstructed up to a 3D projective transformation, as well as the camera position and orientation.

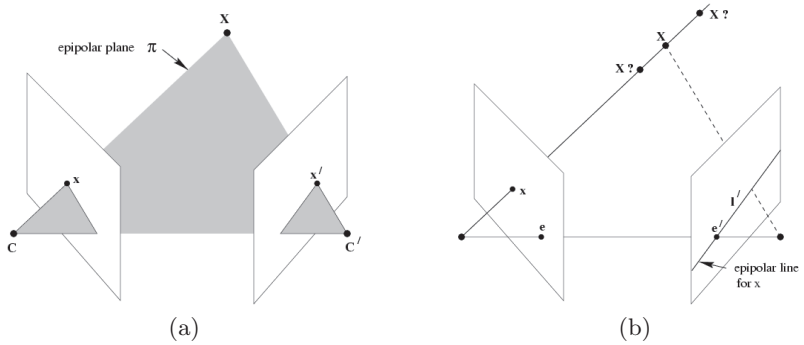


Figure 1.3: (a) The optical center of two cameras and a 3D point form a epipolar plane which intersects two image planes at the epipolar line. (b) Given one point $\mathbf{x}$ in the left image, its correspondence in the right image must be on the corresponding epipolar line.

Chapter 3: Calibration Harp: A Measurement Tool of Lens Distortion Correction Precision

Lens distortion is a non-linear deformation which deviates a pinhole camera from perspective projection. The alignment is the only property preserved in the perspective projection. So it is reasonable to measure the straightness of the projection of 3D straight lines to evaluate the lens distortion correction precision. To this aim one can photograph tightened good quality strings. It is relatively easy to ensure a high straightness by tightening the strings on a frame, while it is more delicate to choose an appropriate type of string. We tried four types of strings and found that the opaque fishing string is the best choice for our purpose. An evaluation pattern made of several parallel tightly stretched opaque fishing strings with a translucent paper as background, called "calibration harp" is thus built (see Fig. 1.4). The Devernay sub-pixel precision edge detector [52] is used to extract the edge points in image, which are

then associated to the line segments detected by LSD (Line Segment Detector) [174]. Finally, the distortion correction is evaluated as the root-mean-square (RMS) distance from the edge points belonging to a same line segment to their corresponding linear regression line.

(a) (b)

Figure 1.4: (a) The harp of opaque fishing strings with a translucent paper as background. (b) A close-up of (a).

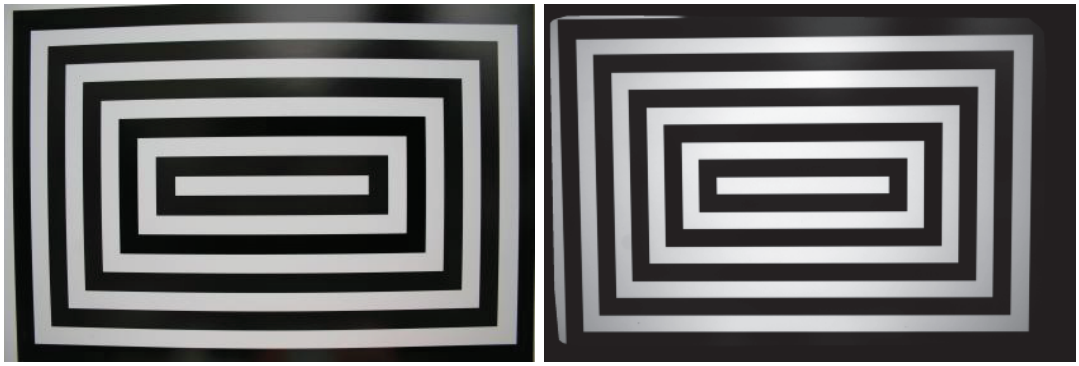
Chapter 4: Non-parametric lens distortion correction

This chapter presents a first technique to correct the distortion with high precision. By high precision, we mean that the residual error between the camera and its numerical model obtained by calibration should be far smaller than the pixel size. At first sight, this problem seemed to have been solved adequately by recent global calibration methods. The celebrated Lavest *et al.* method [95] measures the non-flatness of a pattern and yields a remarkably small re-projection error of about 0.02 pixels, which outperforms the precision of other methods. For the goals of computer vision, this precision would be more than sufficient. Yet, this chapter describes a seriously discrepant accuracy measurement contradicting this hasty conclusion. According to the measurement tool of distortion correction precision developed in Chapter 3, the only objective and correct criterion is straightness of corrected lines.

Following this tool, the accuracy criterion used herewith directly measures the straightness of corrected lines. We shall see that this straightness criterion gives a RMSE as large as 0.2 pixel, which contradicts the 0.02 pixel re-projection accuracy. This significant discrepancy means that, in the global optimization process, errors in the external and internal camera parameter are being compensated by opposite errors in the distortion model. Thus, an inaccurate distortion model can pass undetected. Such facts raise a solid objection to global calibration methods, which estimate simultaneously the lens distortion and the camera parameters. This chapter reconsiders the whole calibration chain and examines an alternative way to guarantee a high accuracy. A useful tool toward this goal will be proposed and carefully tested. It is a direct non-parametric, non-iterative, and model-free distortion correction method. By non-parametric and model-free, we mean that the distortion model allows for any diffeomorphism.

This non-parametric method requires a flat and textured pattern. By using the dense matchings between the pattern and its photo, a distortion field can be obtained by trian-

gulation and local affine interpolation. The obtained precision compares favorably to the distortion given by state of the art global calibration and reaches a RMSE of 0.08 pixels (see Fig. 1.5 for a correction example). The non-flatness of the pattern is a limitation of this method and can introduce a systematic error in the distortion correction. Nonetheless, we also show that this accuracy can still be improved in the next two chapters.



(a) (b)

Figure 1.5: A correction example. (a) distorted image. (b) corrected image.

Chapter 5: Self-Consistency and Universality of Camera Lens Distortion Models

This chapter is a preparation for the next chapter. Due to the difficulty to control the flatness of a pattern, we will go back to the parametric method in the next chapter to obtain a higher correction precision. For any parametric method, an appropriate distortion model is necessary. Even though there exist many distortion models in the literature, it is not clear which one is more appropriate than the others. In addition, the role of distorted points and undistorted points seems to be interchangeable in literature, which makes the notion of distortion model ambiguous.

In this chapter, the concepts of "self-consistency" and "universality" are introduced to evaluate the validity and precision of camera lens distortion models. Self-consistency is evaluated by the residual error when distortion generated with a certain model is corrected (using the model in reverse way) by the best parameters for the same model. Analogously, universality is measured by the residual error when a model is used to correct distortion generated by a family of other models.

Five classic camera lens distortion models are reviewed and compared for their degree of self-consistency and universality. *The realistic synthetic experiments shows clearly that the polynomial model is self-consistent and more universal than the other models.* Indeed the polynomial model, with order from 8 to 19, permits to approximate any other four models, and the inverse of any other four models including itself, at the precision about 1/100 pixel. The high order of the model is more than compensated by its linearity and its translation invariance, which makes it independent of the distortion center. A real experiment shows that the polynomial model of degree 6 can approximate a real distortion field between a textured

pattern and its photo at a 1/100 pixel precision (see Fig. 1.6). So the polynomial model will be chosen in the next chapter as distortion model to improve the correction precision.

Chapter 6: High Precision Camera Calibration with a Harp

This chapter is a continuation of the last chapter to improve the precision of lens distortion correction. Even though non-parametric pattern based methods do not depend on the *a priori* choice of a distortion model with a fixed number of parameters, to achieve a high precision, they require a very flat non deformable plate with highly accurate patterns printed on it. It is shown in this chapter that a relatively small $100\mu m$ flatness error can nonetheless introduce a 0.3 pixels error in distortion estimation. But to fabricate sizeable very flat patterns (a half meter or more is necessary for camera calibration), or even to validate their flatness, is a very difficult physical and technical problem. This is why, as suggested in "plumb line methods" the easiest physical object for which we can ensure some serious straightness are tightened strings. This is why we resort to a real plumb-line method to correct the distortion.

Based on the well-known fact that a camera follows the pinhole model if and only if the projection of every line in space onto the camera is a line, it is sufficient to correct the distorted lines to obtain the pinhole camera [53, 3, 4, 89]. The "calibration harp" built in Chapter 3 to evaluate the precision of distortion correction can be directly used here. But this time, it is used both as a distortion correction tool and as a validation tool. The hardware problem of plumb-line methods is easily solved. But we still need a good distortion model to associate with the plumb-line method to treat different types of realistic lens distortion. According to our discussion of the self-consistency and universality of the main models, the polynomial model seems more adapted to correct real distortion. In addition, it is invariant to any translation to the distortion center. So the distortion center can be fixed anywhere without being estimated. This is a big advantage compared to other models.

Photographs of different orientations were taken to estimate the best coefficients of the polynomial model to correct the distortion (see Fig. 1.7). Real experiments show that no artificial bias is created on the corrected strings and higher precision is attained compared to non-parametric pattern based method. Both the harp of sewing strings and the harp of opaque fishing strings were tested in the experiments. With the sewing strings harp, the correction precision is better than the non-parametric pattern based method and no global artificial bias is observed. With the opaque fishing strings harp, the residual oscillation due to the braid pattern of sewing strings is largely reduced (see Fig. 1.8). We do gain a precision factor about 2 over sewing strings harp and achieve a average correction precision of about 0.02 pixels. This precision is much better than the result given by global camera calibration, which is not stable and varies with the parameters used in the distortion model.

Chapter 7: Three-Step Image Rectification

The epipolar geometry is particularly easy to deal with when the two camera planes are co-planar and parallel with the line connecting the optical center of both cameras. In this case, the corresponding epipolar lines also coincide and align with the x-axis of the image. This means that one point has the same y-coordinate as its corresponding point. This special geometry can be achieved by rotating two cameras without changing their optical center. This is equivalent to applying a homography on each image respectively. This process is

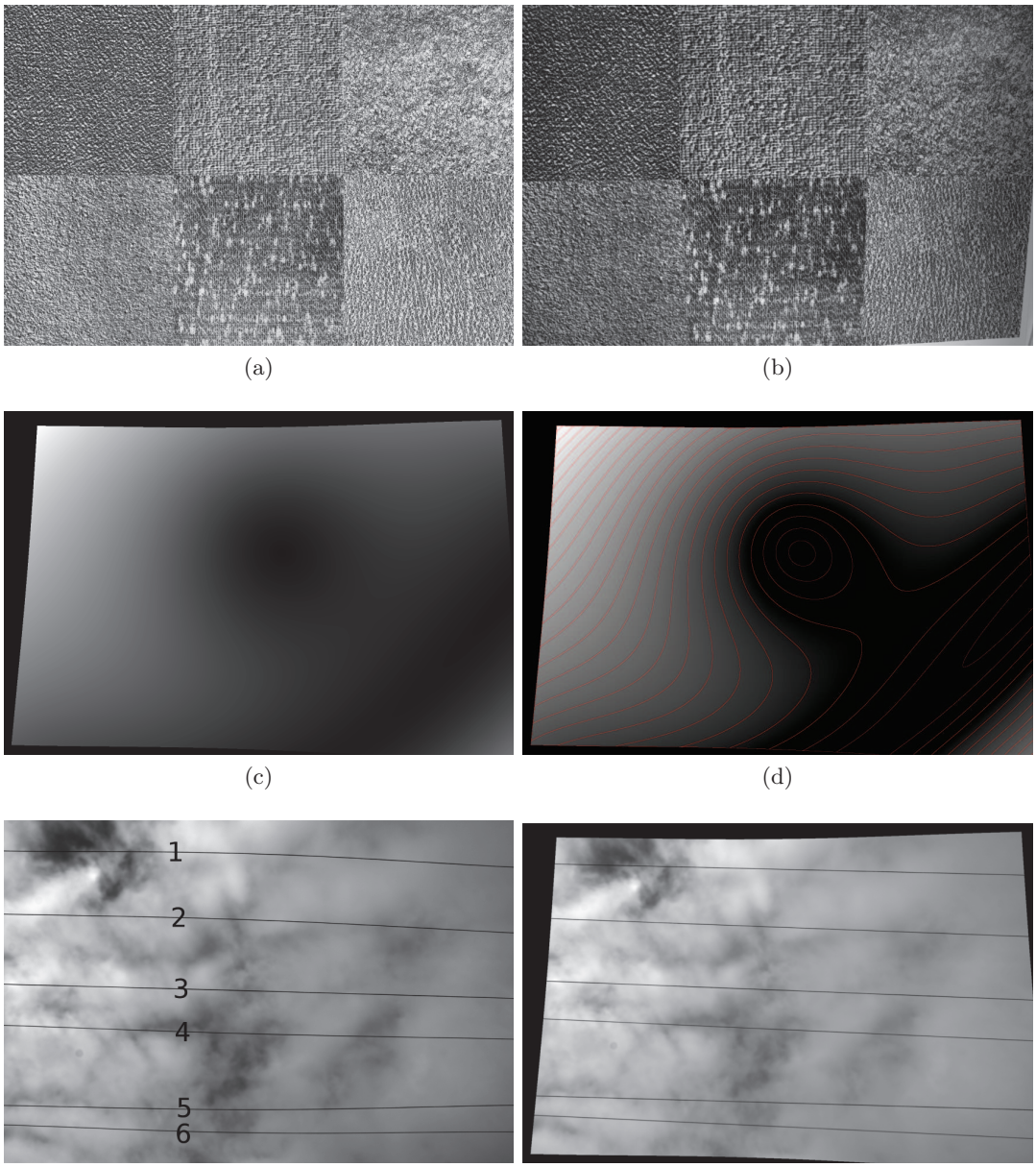


Figure 1.6: (a) digital pattern. (b) a photo of the digital pattern. (c) the distortion field constructed by the estimated parameters of the polynomial model. (d) level lines of (c) with quantization step of 20. (e) distorted images of tightly stretched lines. (f) corrected image by polynomial model.

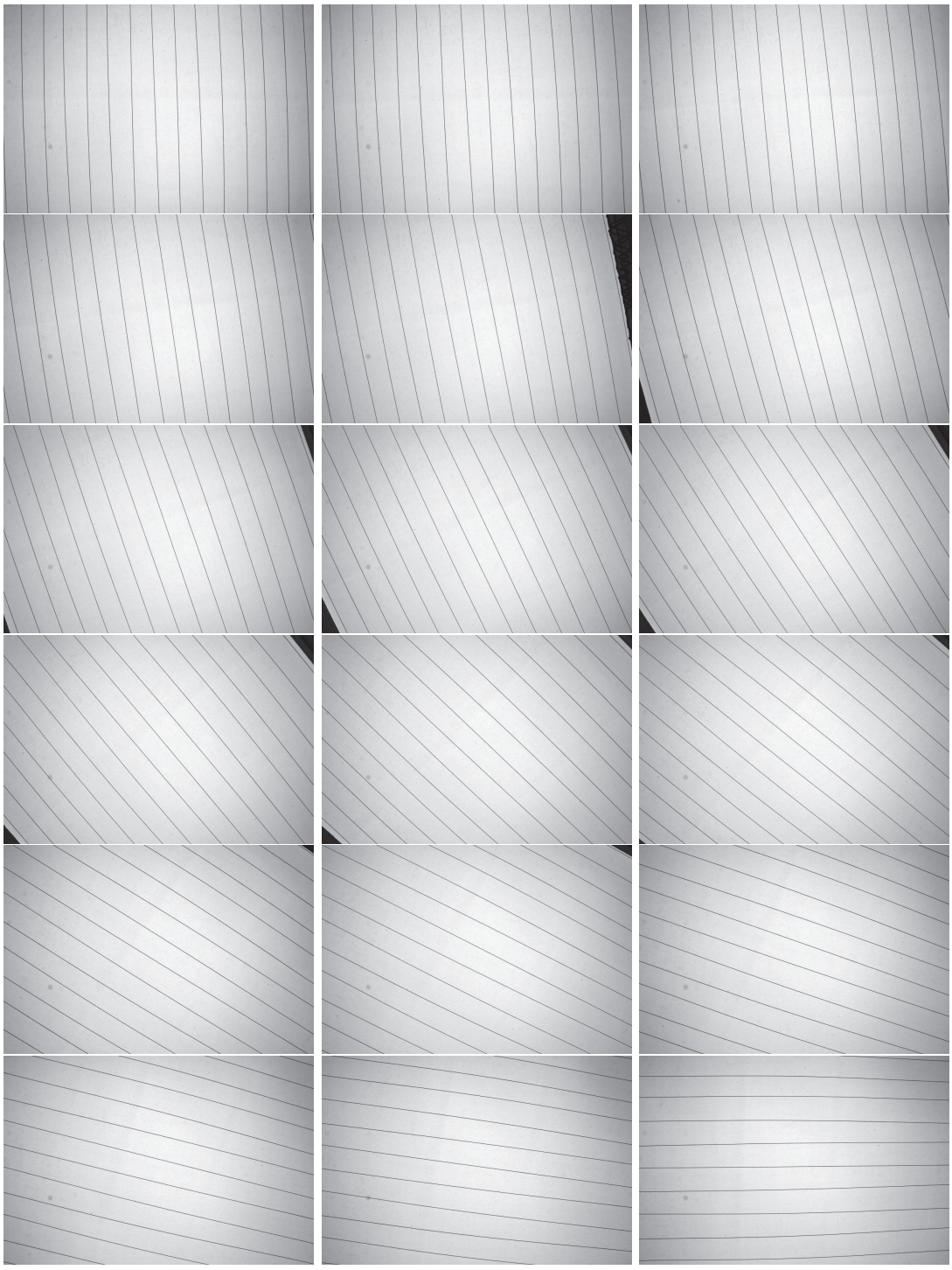
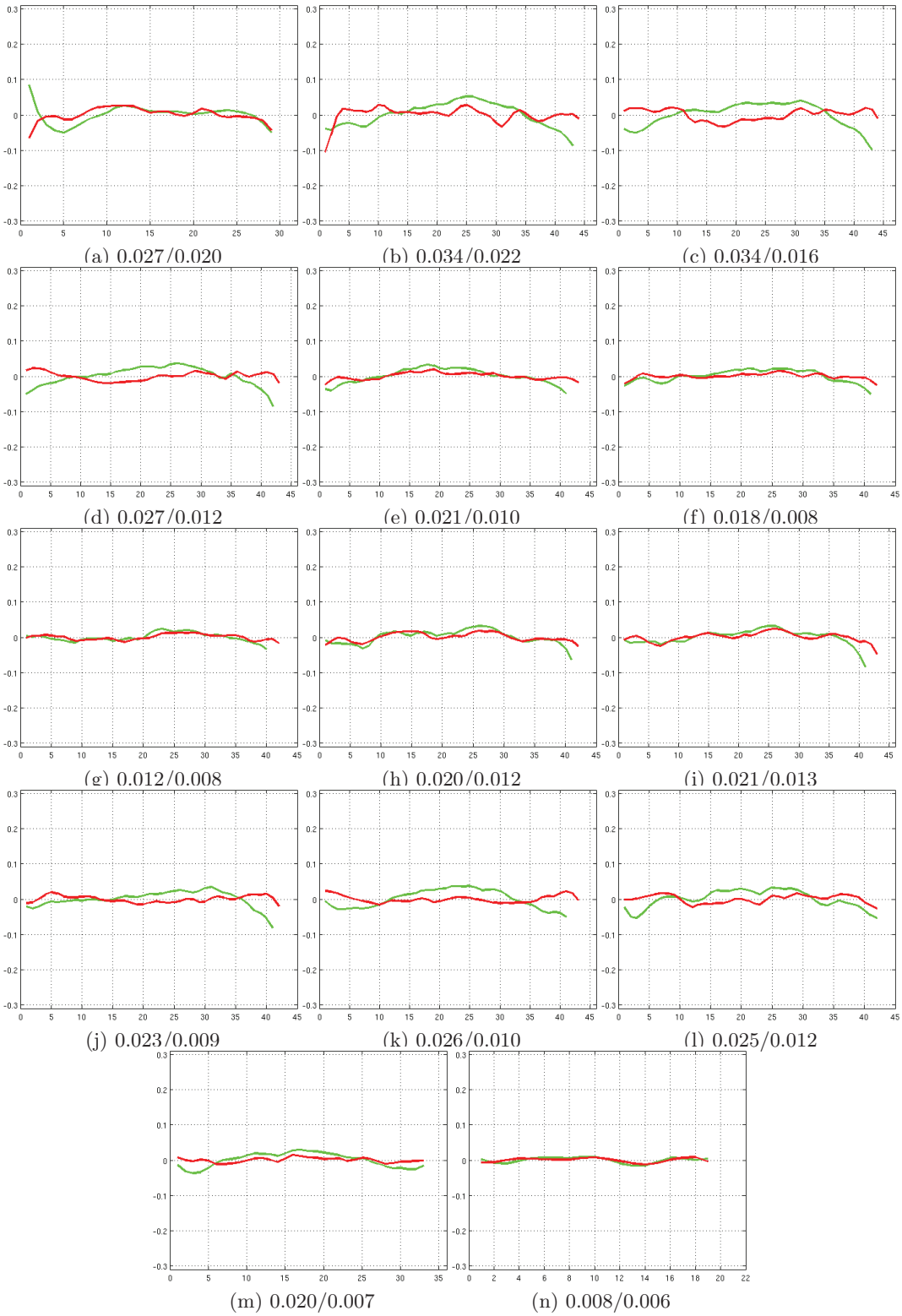


Figure 1.7: Distorted fishing strings taken by the camera fixed on a tripod with different orientations.



(a) 0.027/0.020 (b) 0.034/0.022 (c) 0.034/0.016

(d) 0.027/0.012 (e) 0.021/0.010 (f) 0.018/0.008

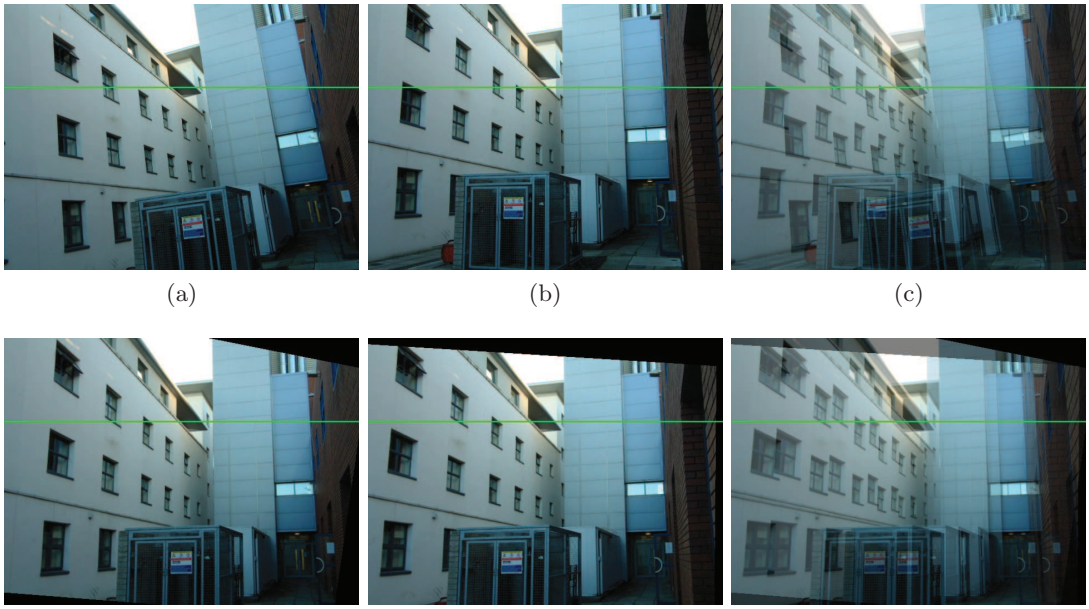
(g) 0.012/0.008 (h) 0.020/0.012 (i) 0.021/0.013

(j) 0.023/0.009 (k) 0.026/0.010 (l) 0.025/0.012

(m) 0.020/0.007 (n) 0.008/0.006

Figure 1.8: Correction performance of the proposed plumb-line method with a harp made up of fishing strings. The distance (in pixels) from the edge point to the corresponding linear regression line is plotted. The x-axis is the index of edge points. The range of y-axis is from -0.3 pixels to 0.3 pixels. The straightness error (in pixels) measured as root mean square distance from the edge points to their linear regression line is given below each figure. Note that each figure contains two curves because there are two sides for one string. The camera focal length is fixed 55 mm and the distance between camera and object is about 100 cm.

called image rectification in two-view geometry (Fig. 1.9 shows a pair of images before and after rectification). A pair of stereo-rectified images is helpful for dense stereo matching algorithms. It restricts the search domain for each match to a line parallel to the x-axis and also restricts the deformation of the blocs in bloc matching. Due to the redundant degrees of freedom, the solution to stereo-rectification is not unique and actually can lead to undesirable projective distortions or be stuck in a local minimum of the distortion function. Many rectification methods reduce the distortion by different explicit measures. But it is not clear which measure is the most appropriate. We propose a rectification method by a three steps of camera rotation. In each step, the distortion is explicitly reduced by minimizing the rotation angle. For un-calibrated cameras, this method can be formulated as an efficient minimization algorithm by optimizing only one natural parameter, the focal length. This is in contrast with many methods which optimize between 3 and 6 parameters.



(a) (b) (c)

(d) (e) (f)

Figure 1.9: Rectification example. (a) and (b): two original images. (c) the blend of two original images. (d) and (e): two rectified images. (f) the blend of two rectified images. A horizontal line is added to images to check the rectification.

Chapter 8: Matching Precision of SIFT

Image features detection and matching is a fundamental step in many computer vision tasks. Many methods have been proposed in recent years, with the aim to extract image features fully invariant to any geometric and photometric transformation. Even though the state-of-art has not achieved the full invariance, many methods, like SIFT [117], Harris-affine [128] and Hessian-affine [127] combining a robust and distinctive descriptor, give sufficient invariance for many practical applications. In contrast to the advance in the invariance of feature detectors, the matching precision has not been paid enough attention to, even though the repeatability

and stability are extensively studied. Matching precision is evaluated on a pair of images and reflects to some extent the *average relative localization precision* between two images. It depends on the localization precision of feature detector, the scale change between two images, the descriptor construction and matching protocol. In this chapter, we focus on the SIFT method and measure its matching precision by average residual error under different geometric transformations. For the scale invariant feature detector, the matching precision decreases with the scale of the features. This drawback can be avoided by canceling the sub-sampling in SIFT scale space. This first scheme improves the matching precision only when there is no scale change between both images. An iterative scheme is thus proposed to address the scale changes. For real images, a local filtering technique is used to improve the matching precision if the transformation between two image is locally smooth.

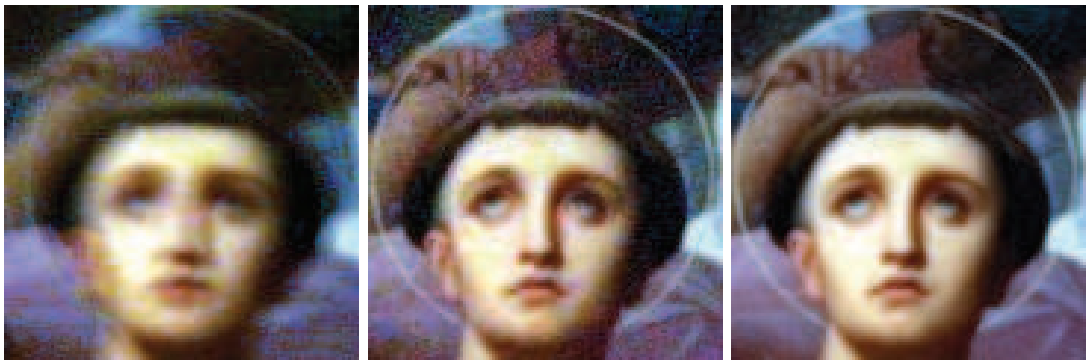
The applications of precise SIFT matching can be envisaged in three directions, two of which are considered in this thesis:

- panorama: the aim is to perform a photo montage seamlessly from several photos. These photos are taken at large distance from the object and the overlapping between two adjacent photos is important.
- burst denoising: the aim is to obtain an image with better SNR from a burst of short exposure images.
- global camera calibration: the aim is to calibrate camera model parameters by several photos of a flat pattern.

Chapter 9: Burst Denoising

Denoising is one of the most important image enhancement techniques. Even though denoising algorithms have been largely developed since years, most of them concentrate on the single image denoising. However, with the increase of memory size and data storage speed, today it is possible for cameras to take a burst of images in a few seconds. This opens a new possibility to do image denoising, in particular under dim light conditions. Many of us had the frustrating experience of taking photos in a museum under low light conditions, where the flash of camera and tripod are forbidden. In such a situation, taking photographs with a hand-held camera is problematic. If the camera is set to a long exposure time, the photograph gets motion blur. If it is taken with short exposure, the image is noisy. This dilemma can be solved by taking a burst of images, each with short-exposure time, as shown in Fig. 9.1. But then, as classical in video processing, an accurate registration technique is required to align the images. Denote by $u(x)$ the ideal non noisy image color at a pixel x. Such an image can be obtained from a still scene by a camera in a fixed position with a long exposure time. The observed value for a short exposure time τ is a random Poisson variable with mean $\tau u(x)$ and the standard variation proportional to $\tau u(x)$. Thus the SNR increases with the exposure time proportionally to τ. The core idea of the burst denoising method is a slight extension of the same law. The only assumption is that the various values at a cross-registered pixel obtained by a burst are i.i.d. Thus, averaging the registered images amounts to averaging several realizations of these random variables. An easy calculation shows that this increases the SNR by a factor proportional to $\sqrt{n}$ where n is the number of shots in the burst. (We call SNR of a given pixel the ratio of its temporal standard deviation to its temporal mean).

Fig. 1.10 summarizes the possibilities offered by an image burst. A long exposure image is exposed to motion blur. The short exposure image is noisy, but sharp. Finally, the image obtained by averaging the images of the burst after registration is both sharp and noiseless. In this real example the burst taken in a gallery had 32 images. The noise standard deviation was therefore divided by approximately 5.6.



(a) (b) (c)

Figure 1.10: (a) one long-exposure image (time = 0.4 sec, ISO=100). (b) one of 16 short-exposure images (time = 1/40 sec, ISO = 1600). (c) the average after registration. All images have been color-balanced to show the same contrast. The long exposure image is blurry due to camera motion. The middle short-exposure image is noisy, and the third one is some 5.6 times less noisy, being the result of averaging 32 short-exposure images.

Even though the denoising power of burst denoising is eventually hemmed by the low growth of the square root, dividing the noise by the mentioned factors and getting an artifact free image is in no way a negligible ambition. Indeed, even the best state of the art denoising methods can create slightly annoying artifacts. If a fine non-periodic texture is present in an image, it is virtually indistinguishable from noise, and actually contains a flat spectrum part which has the same Fourier spectrum as the white noise. Such fine textures can be distinguished from noise only if several samples of the same texture are present in other frames and can be accurately registered.

Yet, this method rises serious technical objections. The main technical objection is: how to register globally the images of a burst? Fortunately, there are several situations where the series of snapshots are indeed related to each other by a homography, and we shall explore these situations first. The homography assumption is actually valid if one of the assumptions is satisfied:

- the only motion of the camera is an arbitrary rotation around its optical center;
- the photographed objects share the same plane in the 3D scene;
- the whole scene is far away from the camera.

In those cases, image registration is equivalent to computing the underlying image homography. But this registration should be sub-pixel accurate. To this aim we will use the precise

SIFT in Chapter 8 and a generalization of ORSA (Optimized Random Sampling Algorithm, [67]) to register all the images together. Yet, in general, the images of 3D scene are not related by a homography, but by an epipolar geometry. Even if the camera is well-calibrated, 3D point-to-point correspondence is impossible to obtain without knowing the depth of the 3D scene. Therefore, we should not expect that a simple homography will work everywhere in the image, but only on a significant part. On this part, we shall say that we have a dominant homography. At each pixel that is well-registered, the registered samples are i.i.d. samples of the same underlying Poisson model. As a result, a signal dependent noise model will be accurately estimated for each colour channel. This model simply is a curve of image intensity versus the standard deviation of the noise.

Averaging does not work at the mis-registered pixels, and block matching methods are at risk on the fine image structures. Thus they will be combined. The simple combination used here will be a convex combination of them, the weight function being based on the noise curve and on the observed standard deviation of the values for the accumulation at a certain pixel. If this standard deviation is compatible with the noise model, the denoised value will be the mean of the samples. Otherwise, the standard deviation test will imply that the registration at this point is inaccurate and a conservative denoising will be applied.

1.7 Main Contributions

In our opinion, the main contributions of this thesis are:

- the concept of a virtual pinhole camera and the proof that it can be used as a real camera;
- a tool to evaluate the precision of distortion correction with a calibration harp;
- a non-parametric pattern based distortion correction method;
- a concept of "self-consistency" and "universality" of distortion models permitting to discuss them thoroughly;
- a distortion correction method with a calibration harp which reconstructs a highly precise virtual pinhole camera;
- a robust three-step image rectification;
- an evaluation and improvement of the SIFT matching precision;
- the "burst denoising" algorithm;

Published or submitted articles linked to the content of this thesis:

- R. Grompone von Gioi, P. Monasse, J.M. Morel and Z. Tang. Lens distortion correction with a calibration harp. *IEEE International Conference on Image Processing*, 2011
- R. Grompone von Gioi, P. Monasse, J.M. Morel and Z. Tang. Self-consistency and universality of camera lens distortion models. Submitted, 2011

- R. Grompone von Gioi, P. Monasse, J.M. Morel and Z. Tang. Correction de distorsion optique avec une harpe de calibration. Submitted, 2011
- A. Buades, Y. Lou, J.M. Morel and Z. Tang. Multi image noise estimation and denoising. Preprint, 2010
- P. Monasse, J.M. Morel and Z. Tang. Three-step image rectification. *British Machine Vision conference*, 2010
- R. Grompone von Gioi, P. Monasse, J.M. Morel and Z. Tang. Towards High-precision Lens Distortion Correction. *IEEE International Conference on Image Processing*, 2010
- A. Buades, Y. Lou, J.M. Morel and Z. Tang. A Note on multi-image denoising. *International Workshop on Local and Non-Local Approximation in Image Processing*, 2009

Chapter 2

Camera Model and Projective Geometry

Contents

2.1 Introduction

This chapter is dedicated to the introduction of camera model and projective geometry. Some basic concepts will be reviewed to make the following chapters easier to read. The ideal pinhole camera model is first introduced due to its simplicity and linearity in projective geometry. By adding the lens distortion to the pinhole model, a more realistic camera model is obtained. A typical bundle adjustment camera calibration algorithm [171] is then introduced to show how to estimate the internal and external parameters of a camera. The other part of this chapter treats the geometry deduced from two images produced by pinhole cameras. The most important one is perhaps the epipolar geometry, which is the intrinsic projective geometry between two views. It is independent of scene structure, and only depends on the cameras' internal parameters and relative pose. From the algebraic view point, the epipolar geometry is represented by a 3×3 fundamental matrix $\mathbf{F}$, which can be computed by more than 7 correspondences between two images. Finally the image rectification, as an application of the fundamental matrix, is presented.

2.2 Camera Model

A real camera can be modeled as a pinhole camera deviated by lens distortion. Some basic concepts about the pinhole camera are shown in Fig. 2.1:

- camera center (or optical center, or focal point): the point through which all relevant light rays pass.
- image plane: the camera CCD plane where the image is formed. This plane does *not* contain the camera center.
- principal axis: the line from the camera center perpendicular to the image plane.
- principal plane: the plane containing the camera center and parallel to the image plane.
- principal point: the intersection point of the principal axis and the image plane.
- normalized image plane: the plane parallel to the image plane and the principal plane, but at one unit distance from the camera center.
- focal length f: the distance from the camera center to the image plane.
- camera frame: the coordinate frame linked to the camera with origin at the camera center and with Z-axis equal to the principal axis. The X and Y coordinates are the same as those of the image plane.
- world frame: a pre-fixed coordinate frame for 3D points.

Note that there is a rotation and translation between the world frame and the camera frame.

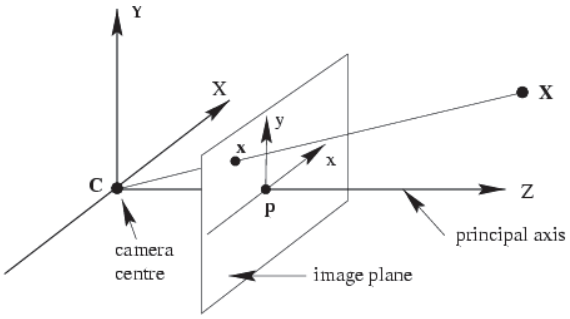


Figure 2.1: Pinhole camera model.

2.2.1 Perspective Projection

In the pinhole camera model, all the relevant light rays pass through the camera center. So it is called perspective projection. To project a 3D point on the camera plane, the first step is to represent it in the camera frame by a translation and a rotation from the world frame

$$\hat{\mathbf{X}}_c = \mathbf{R}(\hat{\mathbf{X}} - \mathbf{C}) \tag{2.1}$$

with $\hat{\mathbf{X}} = (X, Y, Z)^T$ and $\hat{\mathbf{X}}_c = (X_c, Y_c, Z_c)^T$ the coordinate of a point in the world frame and in the camera frame respectively; $\mathbf{C} = (X_o, Y_o, Z_o)^T$ represents the camera center in the world frame. $\mathbf{R} = [\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3]$ represents the rotation matrix between the world frame and the camera frame. $\mathbf{r}_1$, $\mathbf{r}_2$ and $\mathbf{r}_3$ are respectively the X-axis, Y-axis and Z-axis of the world frame represented in the camera frame.

By Thales theorem, the projection of $\hat{\mathbf{X}}_c$ on the camera plane is

$$x_u = fX_c/Z_c \tag{2.2}$$
$$y_u = fY_c/Z_c \tag{2.3}$$

This can also be represented more succinctly in matrix form by using homogeneous coordinates:

$$\mathbf{x}_u = \mathbf{G}\hat{\mathbf{X}}_c = \begin{pmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{pmatrix} \hat{\mathbf{X}}_c \tag{2.4}$$

where $\mathbf{G}$ is the focal length matrix:

$$\mathbf{G} = \begin{pmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{pmatrix}. \tag{2.5}$$

In homogeneous coordinates, $\mathbf{x}_u = (fX_c, fY_c, Z_c)^T$ is equivalent to the 2D point $(fX_c/Z_c, fY_c/Z_c)^T$ by dividing the first two coordinates by the third coordinate. By concatenating the frame change and the perspective projection, a 3D point is projected on a 2D point

$$\mathbf{x}_u = \mathbf{G}\hat{\mathbf{X}}_c = \mathbf{GR}\left(\mathbf{I} \mid -\mathbf{C}\right)\begin{pmatrix} \hat{\mathbf{X}} \\ 1 \end{pmatrix} = \begin{pmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{pmatrix} \mathbf{R}\left(\mathbf{I} \mid -\mathbf{C}\right)\mathbf{X} \tag{2.6}$$

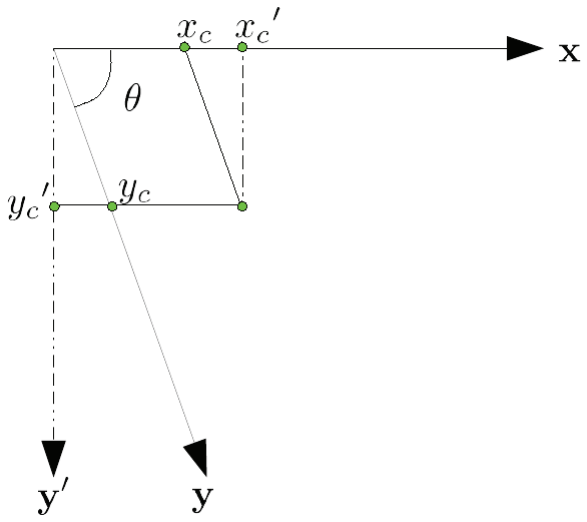


Figure 2.2: The effect of skew on the pixel.

where $\mathbf{X} = (X, Y, Z, 1)^T$ are the homogeneous coordinates of the 3D point $\hat{\mathbf{X}} = (X, Y, Z)^T$.

By inverting the focal length matrix $\mathbf{G}$ on $\mathbf{x}_u$, we obtain the point $\hat{\mathbf{x}}_u = (\hat{x}_u, \hat{y}_u, 1)^T$ with

$$\hat{x}_u = X_c/Z_c \tag{2.7}$$
$$\hat{y}_u = Y_c/Z_c. \tag{2.8}$$

The point $\hat{\mathbf{x}}_u$ is the projection of $\hat{\mathbf{X}}_c$ on the normalized image plane, which is the plane parallel to the image plane situated at unit distance from the optical center.

2.2.2 Internal Parameters

The above obtained 2D point $\mathbf{x}_u = (x_u, y_u, 1)^T$ can have a meter or millimeter or inch unit. But any digital image will be measured in the pixel unit. In addition, the projected point $\mathbf{x}_u$ has the principal point as the origin, while the convention is to take the top-left corner of the image as origin. Due to some manufacturing imprecision, the pixel in a CCD array is not necessarily a square, but may be a rectangle or even a parallelogram. In the skew coordinate system with skewness angle θ, $\mathbf{x}_u$ is represented by $\mathbf{x}'_u = (x'_u, y'_u, 1)^T$ (Fig. 2.2):

$$x'_u = x_u - y_u \cot\theta$$
$$y'_u = \frac{y_u}{\sin\theta}.$$

Putting everything in matrix form, we obtain

$$\begin{aligned}
\mathbf{x} &= \mathbf{LSx}_u = \begin{pmatrix} m_x & 0 & u_0 \\ 0 & m_y & v_0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & -\cot\theta & 0 \\ 0 & \frac{1}{\sin\theta} & 0 \\ 0 & 0 & 1 \end{pmatrix} \mathbf{x}_u \\
&= \begin{pmatrix} m_x & 0 & u_0 \\ 0 & m_y & v_0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & -\cot\theta & 0 \\ 0 & \frac{1}{\sin\theta} & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{pmatrix} \mathbf{R}\left(\mathbf{I} \mid -\mathbf{C}\right)\mathbf{X} \\
&= \begin{pmatrix} m_x f & -m_x f\cot\theta & u_0 \\ 0 & \frac{m_y f}{\sin\theta} & v_0 \\ 0 & 0 & 1 \end{pmatrix} \mathbf{R}\left(\mathbf{I} \mid -\mathbf{C}\right)\mathbf{X} \\
&= \begin{pmatrix} \alpha_x & s & u_0 \\ 0 & \alpha_y & v_0 \\ 0 & 0 & 1 \end{pmatrix} \mathbf{R}[\mathbf{I} \mid -\mathbf{C}]\mathbf{X} = \mathbf{KR}[\mathbf{I} \mid -\mathbf{C}]\mathbf{X} = \mathbf{PX}
\end{aligned} \tag{2.9}$$

where m_x and m_y are the number of pixels per unit length for the skew x-axis direction and for the skew y-axis direction in the image plane respectively; f is the focal length of camera; u_0 and v_0 are the coordinates of the principal point, represented in the skew image frame in pixels; s is the skewness factor which is 0 when the pixel is rectangular; θ is the skewness angle between two sides of image CCD plane.

$$\mathbf{S} = \begin{pmatrix} 1 & -\cot\theta & 0 \\ 0 & \frac{1}{\sin\theta} & 0 \\ 0 & 0 & 1 \end{pmatrix}, \tag{2.10}$$

is called the skewness matrix and

$$\mathbf{L} = \begin{pmatrix} m_x & 0 & u_0 \\ 0 & m_y & v_0 \\ 0 & 0 & 1 \end{pmatrix}. \tag{2.11}$$

is called pixelation matrix. The calibration matrix

$$\mathbf{K} = \mathbf{LSG} = \begin{pmatrix} m_x f & -m_x f\cot\theta & u_0 \\ 0 & \frac{m_y f}{\sin\theta} & v_0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} \alpha_x & s & u_0 \\ 0 & \alpha_y & v_0 \\ 0 & 0 & 1 \end{pmatrix} \tag{2.12}$$

depends only on the camera settings, not on its position. Remark that the entries in $\mathbf{K}$ are not all positive: θ is generally in the range $[0, \pi]$. The entry $-m_x f\cot\theta$ will be positive if $\theta > 90°$, negative if $\theta < 90°$ and 0 when $\theta = 90°$. The entry $\frac{m_y f}{\sin(\theta)}$ is always positive. So the determinant of $\mathbf{K}$ will always be positive.

2.2.3 Projection Matrix

In conclusion, the perspective projection from 3D to 2D performed by a pinhole camera can be represented by a 3×4 matrix $\mathbf{P} = \mathbf{KR}\,[\mathbf{I} \mid -\mathbf{C}]$, which is called the camera projection matrix. This matrix contains all the parameters of the camera: the calibration matrix as *internal parameters*; the camera orientation and camera center as *external parameters*. Remark that the projection matrix has rank 3 and that the right null vector of $\mathbf{P}$ is the vector representing the camera center in homogeneous coordinates: $\mathbf{PC} = \mathbf{0}$.

2.2.4 Lens Distortion

The pinhole camera is only an ideal camera model. Various effects deviate the real camera from the pinhole camera, like optic blur, lens distortion, vignetting, chromatic aberration, etc. However, from the geometric viewpoint, the lens distortion is the main difference between a real camera and an ideal pinhole camera. The lens distortion is the result of several types of imperfections in the design and assembly of lenses composing the camera optical system. The distortion produces a space-varying displacement of each point in the image domain from its ideal pinhole position. This displacement can be decomposed into a radial component and a tangential component. The main kinds of distortion considered in the literature are radial distortion, decentering distortion and thin prism distortion [25, 177]. Radial distortion only contributes to the radial component, while decentering distortion and thin prism distortion contribute to both the radial and tangential components.

Lens distortion is applied on the undistorted point $\mathbf{x}_u$ in Eq. (2.6). Denote (x_d, y_d) the distorted point, $(\bar{x}_u, \bar{y}_u)$ the radial undistorted point and $(\bar{x}_d, \bar{y}_d)$ the radial distorted point where $\bar{x}_u = x_u - x_c$, $\bar{y}_u = y_u - y_c$, $\bar{x}_d = x_d - x_c$ and $\bar{y}_d = y_d - y_c$ with (x_c, y_c) the distortion center, which does not undergo the distortion. The distorted radius is $r_d = \sqrt{\bar{x}_d^2 + \bar{y}_d^2}$. The distortion model can be written as [25]:

$$\begin{aligned} \bar{x}_d &= \bar{x}_u \left(k_0 + k_1 r_u + k_2 r_u^2 + \cdots\right) \\ &+ \left[p_1 \left(r_u^2 + 2\bar{x}_u^2\right) + 2p_2 \bar{x}_u \bar{y}_u\right] \left(1 + p_3 r_u^2\right) + s_1 r_u^2 \\ \bar{y}_d &= \bar{y}_u \left(k_0 + k_1 r_u + k_2 r_u^2 + \cdots\right) \\ &+ \left[p_2 \left(r_u^2 + 2\bar{y}_u^2\right) + 2p_1 \bar{x}_u \bar{y}_u\right] \left(1 + p_3 r_u^2\right) + s_2 r_u^2 \end{aligned} \tag{2.13}$$

where p_1, p_2, p_3 are parameters of the decentering distortion, s_1, s_2 parameters of the thin prism distortion, and $k_0, k_1, k_2, \cdots$ parameters of the radial distortion.

The skewness matrix $\mathbf{S}$ in Eq. (2.10) is applied on the distorted point $\mathbf{x}_d = (x_d, y_d, 1)^T$, followed by the pixelation matrix $\mathbf{L}$ in Eq. (2.11). So the final observed point $\mathbf{x}$ in image is

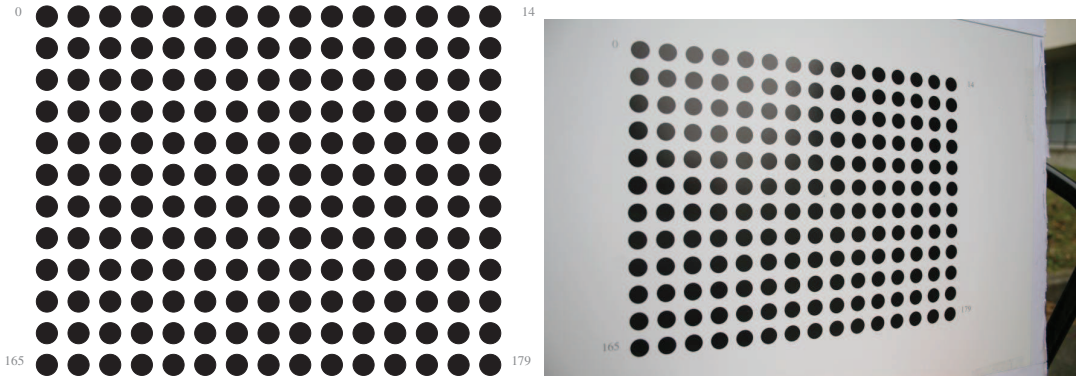
$$\begin{aligned} \mathbf{x} &= \mathbf{LSx}_d = \mathbf{LS}\mathcal{D}\mathbf{x}_u = \mathbf{LS}\mathcal{D}\mathbf{GR}[\mathbf{I} \mid -\mathbf{C}]\mathbf{X} \\ &= \mathbf{LS}\mathcal{D} \begin{pmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{pmatrix} \mathbf{R}[\mathbf{I} \mid -\mathbf{C}]\mathbf{X} \\ &= \mathbf{LS} \begin{pmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{pmatrix} \tilde{\mathcal{D}}\mathbf{R}[\mathbf{I} \mid -\mathbf{C}]\mathbf{X} \\ &= \mathbf{LSG}\tilde{\mathcal{D}}\mathbf{R}[\mathbf{I} \mid -\mathbf{C}]\mathbf{X} = \mathbf{K}\tilde{\mathcal{D}}\mathbf{R}[\mathbf{I} \mid -\mathbf{C}]\mathbf{X} \end{aligned} \tag{2.14}$$

with $\mathcal{D}$ the non-linear distortion represented in Eq. (2.13), and $\tilde{\mathcal{D}}$ another form of distortion model by putting the distortion operator after the focal length matrix. The model $\tilde{\mathcal{D}}$ is similar to $\mathcal{D}$ except that the distortion parameters and the distortion center is normalized by the focal length f: $\tilde{k}_0 = k_0$, $\tilde{k}_1 = fk_1$, $\tilde{k}_2 = f^2 k_1$, $\cdots$, $\tilde{p}_1 = fp_1$, $\tilde{p}_2 = fp_2$, $\tilde{p}_3 = fp_3$, $\tilde{s}_1 = fs_1$, $\tilde{s}_2 = fs_2$, $\tilde{x}_c = x_c/f$ and $\tilde{y}_c = y_c/f$. By including the distortion, the camera model becomes complete:

$$\mathbf{P} = \mathbf{K}\tilde{\mathcal{D}}\mathbf{R}\left[\mathbf{I} \mid -\mathbf{C}\right]. \tag{2.15}$$

2.3 The Lavest *et al.* Method: a Bundle Adjustement

In this section we introduce the Lavest *et al.* method [95], probably the most achieved global calibration method from flat patterns. It will be used for comparison in all experiments. This method is complete in the sense that, in contrast to other global calibration methods, it also estimates the 3D position of the pattern feature points. Thus it is simply a pattern based camera *bundle adjustment* method [171]. The feature points on the pattern are supposed to be known. But due to manufacturing error and mechanical instabilities, the pattern itself can undergo physical deformation. A typical pattern and one of its photographs are shown in Fig. 2.3. The feature points used in the method are the disk centers.



(a) (b)

Figure 2.3: (a) the ideal disk pattern used in the Lavest *et al.* method. (b) one photograph of the pattern.

The basic idea of this method is to find the parameters which, on the one hand, correct the observed feature points in the normalized image plane; on the other hand project the 3D feature points on the pattern onto the normalized image plane, and minimize the distance between corrected points and the re-projected ones.

The distortion model is similar to the one in Eq. (2.13), but with fewer parameters:

$$\begin{aligned} \tilde{x}_u &= f_x(\tilde{x}_d, \tilde{y}_d) = \tilde{x}_d + \tilde{x}_d \left(k_1 \tilde{r}_d^2 + k_2 \tilde{r}_d^4 + k_3 \tilde{r}_d^6\right) \\ &+ p_1 \left(\tilde{r}_d^2 + 2\tilde{x}_d^2\right) + 2p_2 \tilde{x}_d \tilde{y}_d \qquad (2.16) \\ \tilde{y}_u &= f_y(\tilde{x}_d, \tilde{y}_d) = \tilde{y}_d + \tilde{y}_d \left(k_1 \tilde{r}_d^2 + k_2 \tilde{r}_d^4 + k_3 \tilde{r}_d^6\right) \\ &+ p_2 \left(\tilde{r}_d^2 + 2\tilde{y}_d^2\right) + 2p_1 \tilde{x}_d \tilde{y}_d \qquad (2.17) \end{aligned}$$

Remark that the distortion model is used in the inverse direction, that is, for distortion correction. The distorted point on the normalized image plane is denoted by $\tilde{\mathbf{x}}_d = (\tilde{x}_d, \tilde{y}_d, 1)^T$, obtained by applying $\mathbf{K}^{-1}$ on the observed feature point $\mathbf{x}$:

$$\tilde{\mathbf{x}}_d = \mathbf{K}^{-1}\mathbf{x}. \qquad (2.18)$$

The skewness coefficient in the calibration matrix being zero and the distortion center being

at the principal center, we have

$$\tilde{x}_d = (x - u_0)/\alpha_x \quad (2.19)$$
$$\tilde{y}_d = (y - v_0)/\alpha_y \quad (2.20)$$
$$\tilde{r}_d = \sqrt{\tilde{x}_d^2 + \tilde{y}_d}. \quad (2.21)$$

The corresponding 3D point $\mathbf{X}$ is projected to $\hat{\mathbf{x}}_u = (\hat{x}_u, \hat{y}_u, 1)^T$ in the normalized image plane like Eq. (2.7). The error vector $(e_x, e_y)^T$ (in pixels) is

$$e_x = \alpha_x(\tilde{x}_u - \hat{x}_u) \quad (2.22)$$
$$e_y = \alpha_y(\tilde{y}_u - \hat{y}_u). \quad (2.23)$$

The re-projection error ${e_x}^2 + {e_y}^2$ is minimized to estimate the best parameters in the least square sense.

Given m cameras observing a pattern containing n circles, there are $9+3n+6m$ unknown parameters: 5 parameters for the distortion model (3 for radial distortion and 2 for decentering distortion, the distortion center is supposed to the same as the principal point in the calibration matrix), 4 parameters for the calibration matrix (the skewness coefficient being assumed to be 0), $3n$ parameters for the 3D position of n pattern features and $6m$ parameters for the rotation and translation of m cameras.

Denoting by $({e_x}_i^j, {e_y}_i^j)$ the error vector for point j in image i, and the total error vector by $\mathbf{E}(\mathbf{\Phi}) = ({e_x}_1^1, {e_y}_1^1, {e_x}_1^2, {e_y}_1^2, \cdots, {e_x}_m^n, {e_y}_m^n)^T$, with $\mathbf{\Phi}$ the vector of all the unknown parameters. The total error S to be minimized is

$$S = \sum_{i=1}^{m}\sum_{j=1}^{n} {e_x}_i^{j\,2} + {e_y}_i^{j\,2}. \quad (2.24)$$

This is a non-linear optimization problem, even without the distortion. The Levenberg-Marquardt (LM) algorithm can be used to estimate the unknown parameters (see Appendix A.3). LM algorithm is in fact a strategy to decompose a non-linear minimization problem into an iterative step-wise linear problem, from an initialization solution $\mathbf{\Phi}_0$. At iteration k, the augmented normal equation to be solved is

$$\big(\mathbf{J}(\mathbf{\Phi}_k)^T\mathbf{J}(\mathbf{\Phi}_k) + \lambda \mathrm{diag}(\mathbf{J}(\mathbf{\Phi}_k)^T\mathbf{J}(\mathbf{\Phi}_k))\big)\Delta\mathbf{\Phi}_k = -\mathbf{E}(\mathbf{\Phi}_k) \quad (2.25)$$

where $\mathbf{J}(\mathbf{\Phi}_k)$ is the Jacobian of $\mathbf{E}$ with respect to the unknown parameters vector $\mathbf{\Phi}$ at the point $\mathbf{\Phi}_k$, $\Delta\mathbf{\Phi}_k$ the update step to $\mathbf{\Phi}_k$ and $\mathbf{E}(\mathbf{\Phi}_k)$ the residual error vector at the point $\mathbf{\Phi}_k$. LM algorithm is a mixture of Gauss-Newton method (when λ is small) and gradient descent (when λ is large). When λ is large, the coefficient matrix in Eq. (2.25) is close to be diagonal and there is always a solution. When λ is small, the coefficient matrix degenerates to be $\mathbf{J}(\mathbf{\Phi}_k)^T\mathbf{J}(\mathbf{\Phi}_k)$. The Jacobian matrix $\mathbf{J}(\mathbf{\Phi}_k)$ having size $2mn \times (9 + 3n + 6m)$, there is an unique solution to $\Delta\mathbf{\Phi}_k$ if $2mn \geqslant 9 + 3n + 6m$. This condition is easily satisfied. For the pattern in Fig. 2.3 containing 180 points ($n = 180$), 2 photographs ($m = 2$) are sufficient to satisfy $2mn \geqslant 9 + 3n + 6m$. Then the updated step $\Delta\mathbf{\Phi}_k$ can be computed and $\mathbf{\Phi}_k$ can be updated as $\mathbf{\Phi}_{k+1} = \mathbf{\Phi}_k + \Delta\mathbf{\Phi}_k$.

2.3.1 Initialization

The initialization is important for the LM algorithm to converge to the global minimum. The smart linear initialization method proposed in Zhang's flexible method [191] is used here for initialization. Assume the pattern is planar and no distortion is introduced by the camera, then Eq. (2.14) can be written as

$$\begin{aligned} \mathbf{x} &= \mathbf{KR}[\mathbf{I} \mid -\mathbf{C}]\mathbf{X} = \mathbf{K}[\mathbf{R} \mid -\mathbf{RC}]\mathbf{X} \\ &= \mathbf{K}[\mathbf{r}_1\ \mathbf{r}_2\ \mathbf{r}_3\ \mathbf{t}] \begin{pmatrix} X \\ Y \\ 0 \\ 1 \end{pmatrix} = \mathbf{K}[\mathbf{r}_1\ \mathbf{r}_2\ \mathbf{t}] \begin{pmatrix} X \\ Y \\ 1 \end{pmatrix} = \mathbf{H} \begin{pmatrix} X \\ Y \\ 1 \end{pmatrix}, \end{aligned} \tag{2.26}$$

with $\mathbf{R} = [\mathbf{r}_1\ \mathbf{r}_2\ \mathbf{r}_3]$ and $\mathbf{t} = -\mathbf{RC}$. The pattern plane is on the plane $Z = 0$ in the world frame, so the homography can be written

$$\mathbf{H} = \mathbf{K}[\mathbf{r}_1\ \mathbf{r}_2\ \mathbf{t}] \tag{2.27}$$

The homography can be computed by the linear method from more than 4 pairs of correspondences between the pattern and the image or by the non linear method which minimizes the geometric error [89].

Denote $\mathbf{H} = [\mathbf{h}_1\ \mathbf{h}_2\ \mathbf{h}_3]$. Remark that $\mathbf{H}$ is estimated up to a scale factor. So Eq. (2.27) can be rewritten

$$[\mathbf{h}_1\ \mathbf{h}_2\ \mathbf{h}_3] = \lambda\mathbf{K}\ [\mathbf{r}_1\ \mathbf{r}_2\ \mathbf{t}] \tag{2.28}$$

with λ the scale factor equating the left side and the right side in the above equation. Since $\mathbf{r}_1$ and $\mathbf{r}_2$ are orthonormal, we have the following two constraints

$$\mathbf{h}_1^T\mathbf{K}^{-T}\mathbf{K}^{-1}\mathbf{h}_2 = 0 \tag{2.29}$$

$$\mathbf{h}_1^T\mathbf{K}^{-T}\mathbf{K}^{-1}\mathbf{h}_1 = \mathbf{h}_2^T\mathbf{K}^{-T}\mathbf{K}^{-1}\mathbf{h}_2 \tag{2.30}$$

Let

$$\mathbf{B} = \mathbf{K}^{-T}\mathbf{K}^{-1} = \begin{pmatrix} \mathbf{B}_{11} & \mathbf{B}_{12} & \mathbf{B}_{13} \\ \mathbf{B}_{12} & \mathbf{B}_{22} & \mathbf{B}_{23} \\ \mathbf{B}_{13} & \mathbf{B}_{23} & \mathbf{B}_{33} \end{pmatrix} = \begin{pmatrix} \frac{1}{f_x^2} & 0 & \frac{-x_0}{f_x^2} \\ 0 & \frac{1}{f_y^2} & \frac{-y_0}{f_y^2} \\ \frac{-x_0}{f_x^2} & \frac{-y_0}{f_y^2} & \frac{x_0^2}{f_x^2} + \frac{y_0^2}{f_y^2} + 1 \end{pmatrix}.$$

Remark that $\mathbf{B}$ is symmetric and defined by a 5-vector:

$$\mathbf{b} = [\mathbf{B}_{11}\ \mathbf{B}_{13}\ \mathbf{B}_{22}\ \mathbf{B}_{23}\ \mathbf{B}_{33}]^T. \tag{2.31}$$

Let the i^{th} column vector of $\mathbf{H}$ be $\mathbf{h}_i = [h_{1i}\ h_{2i}\ h_{3i}]^T$. Then we have

$$\mathbf{h}_i^T\mathbf{B}\mathbf{h}_j = \mathbf{v}_{ij}^T\mathbf{b} \tag{2.32}$$

with $\mathbf{v}_{ij} = [h_{1i}h_{1j}, h_{1i}h_{2j} + h_{2i}h_{1j}, h_{2i}h_{2j}, h_{3i}h_{1j} + h_{1i}h_{3j}, h_{3i}h_{2j} + h_{2i}h_{3j}, h_{3i}h_{3j}]^T$. So the two constraints in Eq. (2.30) and (2.30), from a homography, can be written as two homogeneous equations in $\mathbf{b}$,

$$\begin{pmatrix} \mathbf{v}_{12}^T \\ (\mathbf{v}_{11} - \mathbf{v}_{22})^T \end{pmatrix} = \mathbf{b}. \tag{2.33}$$

A linear system can be established if there are k images of the pattern (k homographies):

$$\mathbf{V}\mathbf{b} = \mathbf{0} \tag{2.34}$$

where $\mathbf{V}$ is a $2k \times 5$ matrix. If there are $k \geqslant 3$ images, then there is an unique solution to $\mathbf{b}$ up to scale. Once $\mathbf{b}$ is obtained, the calibration matrix $\mathbf{K}$ and the camera rotation and translation can also be computed (see more details in [191]). These parameters are then used for the initialization for LM minimization.

2.3.2 Distortion Correction

One of the aims of camera calibration is to estimate the distortion model and correct the distortion of the images taken by the same camera. In the Lavest *et al.* method, the correction model (Eq. (2.17) and (2.17)) is estimated. Given a distorted image, Eq. (2.17) and (2.17) can be used to send each pixel in the distorted image to its corresponding undistorted point in the corrected image. But the undistorted point has not necessarily integer coordinates and there can be some zones receiving no distorted point.

The distortion model (Eq. (2.14) and (2.14)) is more adapted for distortion correction. Given an integer position in the corrected image, the distortion model allows one to find its corresponding distorted point at sub-pixel position in the distorted image. Any appropriate interpolation technique (like high-order spline interpolation) can be used to recover the intensity. So the corrected image can be obtained without the problem of "unfilled holes".

So, from Eq. (2.17) and (2.17), given an undistorted point $\tilde{\mathbf{x}}_u = (\tilde{x}_u, \tilde{y}_u)$, we want to find the corresponding distorted point $\tilde{\mathbf{x}}_d = (\tilde{x}_d, \tilde{y}_d)$. It is again a non-linear problem and LM can be used. Denote the error vector $\mathbf{E}(\tilde{x}_d, \tilde{y}_d) = (f_x(\tilde{x}_d, \tilde{y}_d) - \tilde{x}_u, f_y(\tilde{x}_d, \tilde{y}_d) - \tilde{y}_u)^T$. The initialization of $(\tilde{x}_d, \tilde{y}_d)$ can be simply the undistorted point $(\tilde{x}_{d_0}, \tilde{y}_{d_0}) = (\tilde{x}_u, \tilde{y}_u)$. The Jacobian $\mathbf{J} = \frac{\partial \mathbf{E}}{\partial \tilde{\mathbf{x}}_d}$ is a 2×2 matrix. So the updated step $\Delta\tilde{\mathbf{x}}_d = (\Delta\tilde{x}_d, \Delta\tilde{x}_d)^T$ can be uniquely determined at each iteration.

Remark that Eq. (2.17) and (2.17) are performed on the normalized image plane. So a normalization step is necessary to obtain the distorted point $\mathbf{x} = (x, y)^T$ on the image plane

$$x = \tilde{x}_d \alpha_x + u_0 \tag{2.35}$$

$$y = \tilde{y}_d \alpha_y + v_0. \tag{2.36}$$

2.3.3 Ellipse Center

The correspondences used in the Lavest *et al.* method are the centers of disks on the pattern (see Fig. 2.3a) and the centers of ellipses on the photo of pattern (see Fig. 2.3b). Of course, the assumption that the circles on the pattern are projected as ellipses in the photo is true only if the pattern is completely planar. There is no difficulty to satisfy the assumption because even if the pattern is not globally planar due to some small mechanic instability or manufacturing error, it is locally planar at the scale of each disk.

A systematic error is introduced by the correspondences, due to the fact that the projection of the center of a circle is not the center of the projected ellipse. This is always true except when the pattern plane is parallel to the image plane or the radius of circle is zero [91].

Assume a circle is projected to the image plane. The camera frame is established as convention (see Fig. 2.1). The circle lies on a plane Π in the world frame. The world frame

has the same origin as the camera frame. The X and Y axis of the world frame are identical with those of the plane where the circle is on; and the Z axis is orthogonal to the plane. In the world frame, the plane is $Z = Z_0$. On the plane Π, the circle equation has the form

$$(X - X_0)^2 + (Y - Y_0)^2 = R_0^2 \tag{2.37}$$

with $(X_0, Y_0)^T$ the circle center and R_0 the radius. The circle can also be represented in matrix form

$$\mathbf{C} = \begin{pmatrix} 1 & 0 & -X_0 \\ 0 & 1 & -Y_0 \\ -X_0 & -Y_0 & -R_0^2 \end{pmatrix}. \tag{2.38}$$

Apply the camera projection matrix (without the calibration matrix $\mathbf{K}$) on any point in the plane Π:

$$\begin{aligned} \mathbf{x} &= \mathbf{R}[\mathbf{I} \mid -\mathbf{0}]\mathbf{X} = \mathbf{R}[\mathbf{I} \mid -\mathbf{0}] \begin{pmatrix} X \\ Y \\ Z_0 \\ 1 \end{pmatrix} \\ &= \mathbf{R} \begin{pmatrix} X \\ Y \\ Z_0 \end{pmatrix} = \mathbf{R} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & Z_0 \end{pmatrix} \begin{pmatrix} X \\ Y \\ 1 \end{pmatrix} = \mathbf{RTX} = \mathbf{HX} \end{aligned} \tag{2.39}$$

with $\mathbf{H}$ the homography between the image and the plane, $\mathbf{X} = (X, Y, 1)^T$ the point on the plane in the world frame.

Under the homography $\mathbf{H}$, the circle $\mathbf{C}$ is transformed to the conic $\mathbf{E}$:

$$\mathbf{E} = \mathbf{H}^{-T}\mathbf{C}\mathbf{H}^{-1} \tag{2.40}$$

The equation $\mathbf{x}^T\mathbf{E}\mathbf{x} = 0$ can be explicitly computed by replacing $\mathbf{X}$ by $\mathbf{H}^{-1}\mathbf{x} = (r_{11}x + r_{21}y + r_{31}, r_{12}x + r_{22}y + r_{32}, \frac{r_{13}x + r_{23}y + r_{33}}{Z_0})^T$ in Eq. (2.37), then we have

$$\begin{aligned} &(k^2 + n^2 - r^2)x^2 + 2(kl + np - rs)xy + (l^2 + p^2 - s^2)y^2 + \\ &2(km + nq - rt)x + 2(lm + pq - st)y + m^2 + q^2 - t^2 = 0 \end{aligned} \tag{2.41}$$

with

$$\begin{aligned} k &= Z_0 r_{11} - X_0 r_{13} \\ l &= Z_0 r_{21} - X_0 r_{23} \\ m &= Z_0 r_{31} - X_0 r_{33} \\ n &= Z_0 r_{12} - Y_0 r_{13} \\ p &= Z_0 r_{22} - Y_0 r_{23} \\ q &= Z_0 r_{32} - Y_0 r_{33} \\ r &= R_0 r_{13} \\ s &= R_0 r_{23} \\ t &= R_0 r_{33}. \end{aligned} \tag{2.42}$$

E represents a quadratic curve, which can be a circle, hyperbola, parabola, or ellipse. In practice, due to the limited field of view, it is usually a circle or ellipse. The center of the ellipse is

$$\begin{aligned} x_e &= \frac{(kp-nl)(lq-pm)-(ks-lr)(tl-ms)-(ns-pr)(tp-qs)}{(kp-nl)^2-(ks-lr)^2-(ns-pr)^2} \\ y_e &= \frac{(kp-nl)(mn-kq)-(ks-lr)(mr-kt)-(ns-pr)(qr-nt)}{(kp-nl)^2-(ks-lr)^2-(ns-pr)^2}. \end{aligned} \tag{2.43}$$

The projection of the circle center can be obtained by setting the radius $R_0 = 0$, leading to $r = s = t = 0$. Then the position of the projection of the circle center $\mathbf{x}_c = (x_c, y_c)^T$

$$\begin{aligned} x_c &= \frac{lq-pm}{kp-nl} \\ y_c &= \frac{mn-kq}{kp-nl}. \end{aligned} \tag{2.44}$$

The center of the ellipse never coincides with the projected circle center if the radius $R_0 \neq 0$ and if the rotation **R** axis is different from the Z-axis of the camera frame.

In the Lavest *et al.* method, the ellipse fitting method gives the center of ellipses. However, what we really need is the projection of the circle centers. So a systematic error could be introduced by the Lavest method if this bias is not corrected. The first strategy is to integrate the correction $(x_c - x_e, y_c - y_e)$ in the non-linear minimization problem. This gives the best solution in the least squares sense with a much lower convergence rate. Another strategy is to find the solution recursively. Once the Lavest method has given the intrinsic and extrinsic parameters of camera, the correction $(x_c - x_e, y_c - y_e)$ can be computed to correct the detected ellipse centers (by considering the calibration matrix **K**)

$$x' = x + \alpha_x(x_c - x_e) \tag{2.45}$$

$$y' = y + \alpha_y(y_c - y_e) \tag{2.46}$$

where the point $(x, y)^T$ is the detected ellipse center and $(x', y')^T$ the corrected ellipse center. Then the Lavest method is run again to estimate the new camera parameters. The process can be iterated until that the correction $(x_c - x_e, y_c - y_e)^T$ is stable. This strategy is simpler and faster and does not modify the core of Lavest method.

2.4 Projective Geometry

Once the distortion is removed and the camera becomes pinhole, the projective geometry is a useful tool to solve different problems in multi-view geometry. Some basic concepts are introduced here and more details can be seen in [89].

2.4.1 Homogeneous Coordinates

Homogeneous coordinates are very useful in multi-view geometry, as they represent many fundamental relationships in vector or matrix form and reduce them to linear algebra. We first introduce the homogeneous notation for points and lines on a plane. Then the homogeneous

notation for 3D space is just evident. A convention in multi-view geometry is that all the geometric entities are represented by column vectors by default.

A line in the plane can be represented by an equation $ax + by + c = 0$ with $(x, y)^T$ a point on the line. It is natural to represent the equation in vector form: $\mathbf{x}^T\mathbf{l} = 0$ with $\mathbf{x} = (x, y, 1)^T$ and $\mathbf{l} = (a, b, c)^T$. But the vector $(x, y, 1)^T$ and $(a, b, c)^T$ are not the only vectors which satisfy the line equation. Any vector $m(x, y, 1)^T$ or $n(a, b, c)^T$ satisfies also the line equation for any $m \neq 0$ and $n \neq 0$. So two vectors related by an overall non-zero scaling are considered as being equivalent. An equivalence class of vectors under this equivalence relationship is known as a homogeneous vector. For a point in the plane, its homogeneous coordinates have the form $\mathbf{x} = (x_1, x_2, x_3)^T$, representing the point inhomogeneous coordinates $(x_1/x_3, x_2/x_3)^T$ $(x_3 \neq 0)$ in $\mathcal{R}^2$. Even if the homogeneous coordinates of points and lines in the plane are 3-vectors, by homogeneity the real degrees of freedom (DOF) are still 2.

Given two lines $\mathbf{l} = (a, b, c)^T$ and $\mathbf{l}' = (a', b', c')^T$, the homogeneous coordinates of the intersection point are $\mathbf{x} = \mathbf{l} \times \mathbf{l}'$ with $\times$ the cross product (see Appendix A.1). On the other hand, the line passing through two points $\mathbf{x}$ and $\mathbf{x}'$ has the form $\mathbf{l} = \mathbf{x} \times \mathbf{x}'$. Note that the simplicity of the expressions is a direct consequence of the use of the homogeneous vector representation of lines and points.

Now consider two parallel lines $ax + by + c = 0$ and $ax + by + c' = 0$. They are represented by vectors $\mathbf{l} = (a, b, c)^T$ and $\mathbf{l}' = (a, b, c')^T$. Note that the first two coordinates are the same because they are parallel. The intersection point of the two lines is $\mathbf{l} \times \mathbf{l}' = (c' - c)(b, -a, 0)^T$. By ignoring the scale $c' - c$, the intersection point is $(b, -a, 0)$. If we try to compute the inhomogeneous coordinates of this point, we have $(b/0, -a/0)^T$, which makes no sense, except to suggest that this point has infinitely large coordinates. In fact, a point with homogeneous coordinates in the form $(x, y, 0)^T$ does not correspond to any finite point in $\mathcal{R}^2$. This agrees with the usual idea that parallel lines meet at infinity.

Homogeneous vectors $\mathbf{x} = (x_1, x_2, x_3)^T$ such that $x_3 \neq 0$ correspond to finite points in $\mathcal{R}^2$. By augmenting $\mathcal{R}^2$ with points having a last coordinate $x_3 = 0$, the resulting space is the set of all homogeneous 3-vectors, namely the 2D projective space $\mathcal{P}^2$. The points with last coordinate $x_3 = 0$ are called ideal points or points at infinity. Each ideal point represents a direction determined by the ratio $x_1 : x_2$ $(x_2 \neq 0)$ or $x_2 : x_1$ $(x_1 \neq 0)$. In addition, all the ideal points lie on a line at infinity, denoted by $\mathbf{l}_\infty = (0, 0, 1)^T$ since $(x_1, x_2, 0)(0, 0, 1)^T = 0$, $\forall x_1, x_2$. Each line $\mathbf{l}$ intersects $\mathbf{l}_\infty$ at an ideal point, which corresponds to the direction of $\mathbf{l}$. So the line at infinity $\mathbf{l}_\infty$ can also be considered as the set of all directions of lines in the plane.

2.4.2 Projective Plane

The set of equivalence classes of vectors in $\mathcal{R}^3 - (0, 0, 0)^T$ forms the projective space $\mathcal{P}^2$ (the vector $(0, 0, 0)^T$ makes no sense in projective space). We can also think of $\mathcal{P}^2$ as a set of rays in $\mathcal{R}^3$. The set of vectors $k(x_1, x_2, x_3)^T$, when the scalar k varies, forms a ray through the origin. Such a ray may be thought of as representing a single point in $\mathcal{P}^2$. In this model, the lines in $\mathcal{P}^2$ are planes passing through the origin. Two non-identical rays lie on exactly one plane, and any two planes intersect in one ray. This is the analogue of two distinct points uniquely defining a line, and two lines always intersecting in a point. Points and lines may be obtained by intersecting this set of rays and planes by the plane $x_3 = 1$. In Fig. 2.4, the rays representing ideal points and the plane representing $\mathbf{l}_\infty$ are parallel to the plane $x_3 = 1$.

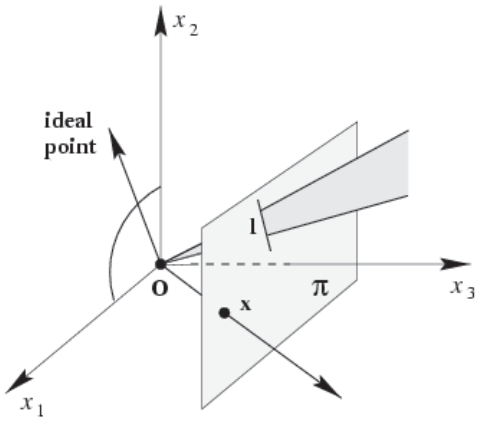


Figure 2.4: Projective plane model.

Notice that in the projective plane, two lines always intersect (at an ideal point if they are parallel).

2.4.3 Transformations

The most important transformation in the projective plane is the projective transformation (or homography), which simply is a non-singular 3×3 matrix, denoted by $\mathbf{H}$. The 2D planar projective transformation preserves the collinearity: If $\mathbf{x}_1$, $\mathbf{x}_2$ and $\mathbf{x}_3$ are on the line $\mathbf{l}$, then $\mathbf{Hx}_1$, $\mathbf{Hx}_2$ and $\mathbf{Hx}_3$ are also on the line $\mathbf{H}^{-T}\mathbf{l}$. (This translates the fact that if three vectors are coplanar, so are their images by a 3D linear transformation). Note that the computation here is in homogeneous coordinates,

$$\begin{pmatrix} x'_1 \\ x'_2 \\ x'_3 \end{pmatrix} = \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}. \tag{2.47}$$

$\mathbf{H}$ in the above equation may be changed by multiplication by an arbitrary non-zero factor without changing the result. The inhomogeneous coordinate of the point $(x'_1, x'_2, x'_3)^T$ is

$$\begin{aligned} \frac{x'_1}{x'_3} &= \frac{h_{11}x_1 + h_{12}x_2 + h_{13}x_3}{h_{31}x_1 + h_{32}x_2 + h_{33}x_3} \\ \frac{x'_2}{x'_3} &= \frac{h_{21}x_1 + h_{22}x_2 + h_{23}x_3}{h_{31}x_1 + h_{32}x_2 + h_{33}x_3}. \end{aligned} \tag{2.48}$$

Note that the multiplication of H itself by a scalar factor yields the same transform. So $\mathbf{H}$ is also a homogeneous geometric entity and it has 8 degrees of freedom. A homography is uniquely determined by the knowledge of 4 correspondences (with no more than 2 of them on any line). With $h_{31} = h_{32} = 0 \neq h_{33}$, a homography degenerates to an affine transformation, which keeps the line at infinity $\mathbf{l}_\infty$ globally invariant. A point $\mathbf{x}$ is transformed to point $\mathbf{Hx}$ under the homography H, while a line $\mathbf{l}$ is transformed to a line $\mathbf{H}^{-\mathbf{T}}\mathbf{l}$. More details about projective transformations can be found in [89].

2.5 Camera Rotation

A particular 2D projective transformation can be induced by a pure camera rotation without changing its optical center (Fig. 2.5). Given a 3D point $\mathbf{X}$, its projected image by rotating the camera is

$$\begin{aligned} \mathbf{x}_1 &= \mathbf{K}_1\mathbf{R}_1[\mathbf{I} \mid -\mathbf{C}]\mathbf{X} \\ \mathbf{x}_2 &= \mathbf{K}_2\mathbf{R}_2[\mathbf{I} \mid -\mathbf{C}]\mathbf{X}. \end{aligned} \tag{2.49}$$

By a simple substitution, we find that $\mathbf{x}_1$ and $\mathbf{x}_2$ are related by the homography

$$\mathbf{x}_2 = \mathbf{K}_2\mathbf{R}_2\mathbf{R}_1^{-1}\mathbf{K}_1^{-1}\mathbf{x}_1 = \mathbf{H}\mathbf{x}_1. \tag{2.50}$$

A camera rotation is not the only situation which induces the homography. When the 3D scene is a plane, the relationship between two images taken by a camera is also a homography. The third situation inducing homography is that the scene is very far away from the camera.

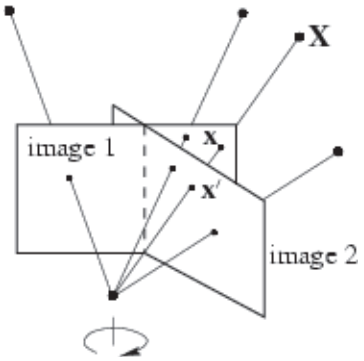


Figure 2.5: 2-D projective transformation (homography) induced by a pure camera rotation without changing camera center.

2.6 Fundamental Matrix

The epipolar geometry is the intrinsic projective geometry between two views. It is independent of the observed scene structure and of the world frame. It depends only on the camera internal parameters and relative pose. The fundamental matrix $\mathbf{F}$ encapsulates this intrinsic geometry. It is a 3×3 matrix of rank 2. If $\mathbf{x}$ and $\mathbf{x}'$ is a pair of corresponding points in two views, then they satisfy the scalar equation $\mathbf{x}'^T\mathbf{F}\mathbf{x} = 0$. The fundamental matrix can be computed from correspondences of imaged scene points alone, without requiring knowledge of the camera internal parameters or relative pose.

2.6.1 Epipolar Constraint

In Fig. 2.6, we can see that a 3D point $\mathbf{X}$ is projected to $\mathbf{x}$ and $\mathbf{x}'$ in two views. These three points $\mathbf{X}$, $\mathbf{x}$ and $\mathbf{x}'$ form the epipolar plane, which intersects the two image planes by two epipolar lines respectively. The line connecting the two camera centers is called baseline and

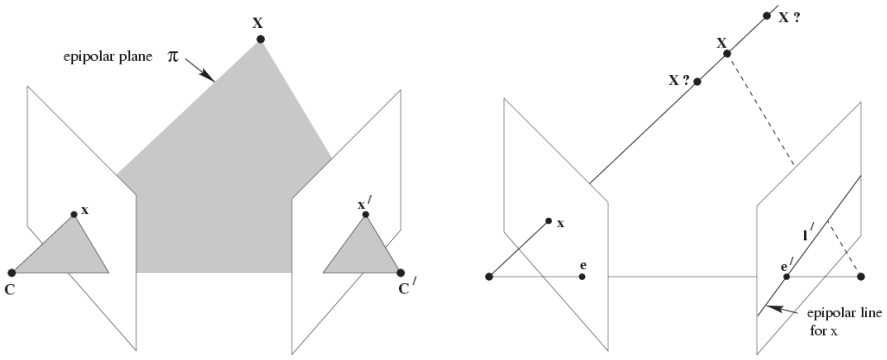


Figure 2.6: Left: The epipolar plane is formed by two camera centers and the 3-D point $\mathbf{X}$ Right: The 3-D point $\mathbf{X}$ must be on the back-projected ray defined by the left camera center and $\mathbf{x}$. This ray is imaged as a line $\mathbf{l}'$ called epipolar line in the second view and the image of $\mathbf{X}$ must lie on $\mathbf{l}'$.

intersects the two image planes at the two epipoles respectively. The well-known equation for epipolar geometry writes

$$\mathbf{x}'^T\mathbf{F}\mathbf{x} = 0 \tag{2.51}$$

with $\mathbf{F}$ the fundamental matrix which is the algebraic representation of the epipolar geometry. We should remark that this equation is only a necessary condition for two corresponding points since $\mathbf{F}$ projects a point in one image to a corresponding epipolar line in the other image (Fig. 2.6). In Fig. 2.6, it is clear that the epipolar line is obtained by projecting the ray, which passes the optical center $\mathbf{C}$ and $\mathbf{x}$, to the other image. Since the position of the 3-D scene point $\mathbf{X}$ is not determined on the ray, the image point $\mathbf{x}'$ could be anywhere on the epipolar line $\mathbf{l}'$. So the mapping from a point to its epipolar line: $\mathbf{x} \mapsto \mathbf{l}'$ is a projection represented by $\mathbf{F}$, which can be written as

$$\mathbf{l}' = \mathbf{F}\mathbf{x}. \tag{2.52}$$

In geometry, the points in the left image represent a 2-D projective space, and the epipolar lines in the right image represent a 1-D projective space since all the epipolar lines have a common intersection point, the epipole $\mathbf{e}'$. So $\mathbf{F}$ represents a projection from a 2-D projective space to a 1-D projective space. From this viewpoint, it is natural to derive that $\mathbf{F}$ has rank equal to 2. For any point $\mathbf{x}'$ on the epipolar line $\mathbf{l}'$, we have the equation $\mathbf{x}'^T\mathbf{l}' = 0$, which is exactly (2.51) by using (2.52).

When the position of the 3-D scene point $\mathbf{X}$ varies, the epipolar plane rotates about the baseline (Fig. 2.6). The family of planes is known as an epipolar pencil, which intersects the two images at two pencils of epipolar lines. Each pencil of epipolar lines intersects at the corresponding epipole.

To get the explicit form of the $\mathbf{F}$ matrix, we can write the projections of a 3D point $\mathbf{X} = (X, Y, Z, 1)^T$ in the two cameras, expressed in the first camera coordinate frame!

$$\lambda\mathbf{x} = \mathbf{K}\,[\mathbf{I}\ |\mathbf{0}]\,\mathbf{X} = \mathbf{K}\hat{\mathbf{X}} \tag{2.53}$$

$$\lambda'\mathbf{x}' = \mathbf{K}'\,[\mathbf{R}\ |\mathbf{t}]\,\mathbf{X}. \tag{2.54}$$

$\hat{\mathbf{X}} = (X, Y, Z)^T$ is the non-homogeneous coordinate of $\mathbf{X}$, $\mathbf{x} = (x, y, 1)^T$ and $\mathbf{x}' = (x', y', 1)^T$ are the projections of $\mathbf{X}$ in camera 1 and camera 2 respectively. $\mathbf{R}$ is the rotation of the camera 2 frame relative to the camera 1 frame and $\mathbf{t}$ is the coordinate of camera 1's optical center in the camera 2 frame. These 6 scalar equations have 5 parameters depending on the scene structure: $\hat{\mathbf{X}}$, λ and λ'. So it is expected that by eliminating these from the system we shall get one scalar equation. The first equation, $\hat{\mathbf{X}} = \lambda K^{-1}\mathbf{x}$ substituted in the second equation, yields

$$\lambda'\mathbf{K}'^{-1}\mathbf{x}' = \lambda\mathbf{R}\mathbf{K}^{-1}\mathbf{x} + \mathbf{t}. \tag{2.55}$$

The cross product (see Appendix A.1) of each side with vector $\mathbf{t}$ gives

$$\lambda'[\mathbf{t}]_\times\mathbf{K}'^{-1}\mathbf{x}' = [\mathbf{t}]_\times\mathbf{R}\mathbf{K}^{-1}\mathbf{x} \tag{2.56}$$

and since the left hand side is orthogonal to $\mathbf{K}'^{-1}\mathbf{x}'$, we get

$$(\mathbf{K}'^{-1}\mathbf{x}')^T[\mathbf{t}]_\times\mathbf{R}\mathbf{K}^{-1}\mathbf{x} = 0, \tag{2.57}$$

which is (2.51) with

$$\mathbf{F} = \mathbf{K}'^{-T}[\mathbf{t}]_\times\mathbf{R}\mathbf{K}^{-1}. \tag{2.58}$$

Remark that the 3D point $\hat{\mathbf{X}}$ is not the only point satisfying the above deduction. In fact, the ray passing the optical center of camera 1 (the origin) and $\hat{\mathbf{X}}$ is projected as an epipolar line in camera 2, and any point $\mathbf{x}''$ on the epipolar line and the point $\mathbf{x}$ are related the equation: $\mathbf{x}''^T\mathbf{F}\mathbf{x} = 0$. So this equation is just a necessary (not sufficient) condition for that $\mathbf{x}$ corresponds to $\mathbf{x}''$.

Some properties of the fundamental matrix are summarized here (the details could be found in chapter 9 of [89])

- F is a 3×3 rank-2 homogeneous matrix with 7 freedom degrees
- $\mathbf{x}'^T\mathbf{F}\mathbf{x} = 0$ for a pair of corresponding image points $\mathbf{x}$ and $\mathbf{x}'$
  - Given a point $\mathbf{x}$ in the left image, the corresponding epipolar line in the right image is $\mathbf{l}' = \mathbf{F}\mathbf{x}$
  - Given a point $\mathbf{x}'$ in the right image, the corresponding epipolar line in the left image is $\mathbf{l} = \mathbf{F}^T\mathbf{x}'$
- $\mathbf{F}\mathbf{e} = \mathbf{0}$ and $\mathbf{F}^T\mathbf{e}' = 0$: the epipoles are the null vectors of $\mathbf{F}$ and $\mathbf{F}^T$ and all epipolar lines contain the epipoles;
- Correspondence between epipolar lines: $\mathbf{l}' = \mathbf{F}[\mathbf{e}]_\times\mathbf{l}$ and $\mathbf{l} = \mathbf{F}^T[\mathbf{e}']_\times\mathbf{l}'$.

$\mathbf{F}$ only depends on projective properties of the cameras $\mathbf{P}$, $\mathbf{P}'$. The camera projection matrices relate 3-space measurements to image measurements and so depend on both the image coordinate frame and the choice of the world coordinate frame. On the contrary $\mathbf{F}$ does not depend on the choice of the world frame. More precisely, if $\mathbf{H}$ is a 4×4 matrix representing a projective transformation of the 3D-space, then the fundamental matrix corresponding to the pairs of camera matrix $(\mathbf{P}, \mathbf{P}')$ and $(\mathbf{P}\mathbf{H}, \mathbf{P}'\mathbf{H})$ are the same. This is because $\mathbf{x} = \mathbf{P}\mathbf{X} = (\mathbf{P}\mathbf{H})\left(\mathbf{H}^{-1}\mathbf{X}\right)$ and $\mathbf{x}' = \mathbf{P}'\mathbf{X} = (\mathbf{P}'\mathbf{H})\left(\mathbf{H}^{-1}\mathbf{X}\right)$. So the matched points $\mathbf{x} \leftrightarrow \mathbf{x}'$ are not

changed under the 3-D projective transformation $\mathbf{H}$ even if the pair of camera matrix and the 3-D point are changed. Then the fundamental matrix also remains unchanged.

When $\mathbf{K}$ and $\mathbf{K}'$ are known (called the calibrated case), the constraint (2.51) but involving $\mathbf{K}^{-1}\mathbf{x}$ and $\mathbf{K}'^{-1}\mathbf{x}'$ can be rewritten as

$$\left(\mathbf{K}'^{-1}\mathbf{x}'\right)^T \mathbf{E} \left(\mathbf{K}^{-1}\mathbf{x}\right) = 0 \tag{2.59}$$

with

$$\mathbf{E} = [\mathbf{t}]_\times \mathbf{R}. \tag{2.60}$$

$\mathbf{E}$ is called the essential matrix; its discovery was published in 1981 by H.C. Longuet-Higgins in *Nature*. It anticipated by 10 years the discovery of the fundamental matrix!

2.6.2 Computation

Two algorithms are usually used to compute the $\mathbf{F}$ matrix: the 7-point algorithm and the 8-point algorithm. The 7-point algorithm is used when there are only 7 correspondences, which is the minimal number of correspondences needed to determine a finite number of $\mathbf{F}$ (one or three solutions to $\mathbf{F}$). The 8-point algorithm requires 8 or more correspondences. In the rectification we will mention later, the 8-point algorithm is used because there are usually more than 8 correspondences.

The 7-point Algorithm

The equation $\mathbf{x}'^T\mathbf{F}\mathbf{x} = 0$ gives us one linear equation in the unknown entries of $\mathbf{F}$. More explicitly, given $\mathbf{x} = (x, y, 1)^T$, $\mathbf{x}' = (x', y', 1)^T$, write $\mathbf{f}$ as the vector made of entries of $\mathbf{F}$ in row-major order

$$\mathbf{f} = \left(\mathbf{F}_{11}, \mathbf{F}_{12}, \mathbf{F}_{13}, \mathbf{F}_{21}, \mathbf{F}_{22}, \mathbf{F}_{23}, \mathbf{F}_{31}, \mathbf{F}_{32}, \mathbf{F}_{33}\right)^T. \tag{2.61}$$

Then the equation can be written as

$$\left(x'x, x'y, x', y'x, y', x, y, 1\right) \mathbf{f} = 0. \tag{2.62}$$

If we have n image point correspondences, the n linear equations can be stacked in a linear system

$$\mathbf{A}\mathbf{f} = \mathbf{0} \tag{2.63}$$

where $\mathbf{A}$ is a $n \times 9$ coefficient matrix. Since $\mathbf{F}$ has 7 degrees of freedom, 7 correspondences are enough to compute $\mathbf{F}$. In this minimum case (7 correspondences), the solution space has dimension 2 of the form $\lambda_1\mathbf{F}_1 + \lambda_2\mathbf{F}_2$ with $\mathbf{F}_1$ and $\mathbf{F}_2$ corresponding to two independent null vectors of $\mathbf{A}$. Since $\mathbf{F}$ is a homogeneous entity, the solution does not change by multiplying a non-zero scalar, for example, $1/(\lambda_1 + \lambda_2)$. So the solution becomes $\alpha\mathbf{F}_1 + (1-\alpha)\mathbf{F}_2$ ($\alpha = \lambda_1/(\lambda_1 + \lambda_2)$).

Remember that $\det(\mathbf{F}) = \det\left(\alpha\mathbf{F}_1 + (1-\alpha)\mathbf{F}_2\right) = 0$ since $\mathbf{F}$ has rank 2. This leads to a cubic polynomial equation in α. It has either 3 real solutions or 1 real solution and 2 complex conjugate solutions. Only the real solutions make sense for $\mathbf{F}$. So $\mathbf{F} = \alpha\mathbf{F}_1 + (1-\alpha)\mathbf{F}_2$ has 1 or 3 possible solutions. The geometric interpretation of 1 or 3 solutions is in chapter 22 of [89]. From the point of view of critical surfaces, the seven points and two camera centers must lie on a quadric surface (since 9 points lie on a quadric). If this quadric is ruled, then there

will be three solutions. On the other hand, if it is not ruled quadric (for instance an ellipsoid) then there will be only one solution. The 7-point algorithm requires the minimum number of correspondences to solve $\mathbf{F}$ and it is often integrated in the outlier detection algorithm to extract the good $\mathbf{F}$ like [67, 129].

The 8-point Algorithm

The 8-point algorithm is a simpler computation method of $\mathbf{F}$. In this case, the solution space has only dimension 1 and the $\mathbf{F}$ is uniquely determined up to scale. But because of noise in point coordinates, $\det(\mathbf{F})$ is not equal to 0. The convenient method to enforce the determinant constraint is to use the SVD (Singular Value Decomposition, see Appendix A.2) and to replace the smallest singular value by 0. The obtained $\mathbf{F}'$ is optimal in the sense of minimizing the Frobenius norm $\|\mathbf{F} - \mathbf{F}'\|$. A key point of the 8-point algorithm is the normalization which makes the points more concentrated around their centroid. In Hartley's original paper [87], he proposed to translate the points so that the origin is at the centroid of the points, and to scale the points so that the average distance from the points to their centroid is equal to $\sqrt{2}$. This normalization improves dramatically the conditioning of the coefficient matrix $\mathbf{A}$ in (2.62) and make all entries of $\mathbf{F}$ contribute approximately equally to the error term.

Algorithm 1 (8-point normalization algorithm)
Objective *Given $n \geqslant 8$ image point correspondences $\{\mathbf{x}_i \leftrightarrow \mathbf{x}'_i\}$, determine the fundamental matrix $\mathbf{F}$ such that $\mathbf{x}'^T_i \mathbf{F} \mathbf{x}_i = 0$*

1. ***Normalization**: Transform the image coordinates according to $\hat{\mathbf{x}}_i = \mathbf{T}\mathbf{x}_i$ and $\hat{\mathbf{x}}'_i = \mathbf{T}'\mathbf{x}'_i$, where $\mathbf{T}$ and $\mathbf{T}'$ are normalization transformations consisting of a translation and scaling:*

$$\mathbf{T} = \begin{pmatrix} 1/\alpha & 0 & -u/\alpha \\ 0 & 1/\alpha & -v/\alpha \\ 0 & 0 & 1 \end{pmatrix}, \quad \mathbf{T}' = \begin{pmatrix} 1/\alpha' & 0 & -u'/\alpha' \\ 0 & 1/\alpha' & -v'/\alpha' \\ 0 & 0 & 1 \end{pmatrix}$$

   with $(u, v)^T$ and $(u', v')^T$ the centroid of the points in the first and second image respectively; $1/\alpha$ and $1/\alpha'$ the scale to make average distance from the points to the centroid equal to $\sqrt{2}$.

2. *Find the fundamental matrix $\hat{\mathbf{F}}$ corresponding to the matches $\{\hat{\mathbf{x}}_i \leftrightarrow \hat{\mathbf{x}}'_i\}$ by*

   (a) ***Linear solution**: Determine $\hat{\mathbf{F}}$ from the singular vector corresponding to the smallest singular value of $\hat{\mathbf{A}}$, where $\hat{\mathbf{A}}$ is the coefficient matrix composed from the matches $\{\hat{\mathbf{x}}_i \leftrightarrow \hat{\mathbf{x}}'_i\}$.*

   (b) ***Constraint enforcement:** Replace $\hat{\mathbf{F}}$ by $\hat{\mathbf{F}}'$ such that $det(\hat{\mathbf{F}}') = 0$ by setting the smallest singular value to be 0.*

3. ***Denormalization:** Set $\mathbf{F} = \mathbf{T}'^T \hat{\mathbf{F}}' \mathbf{T}$. Matrix $\mathbf{F}$ is the fundamental matrix corresponding to the original data $\{\mathbf{x}_i \leftrightarrow \mathbf{x}'_i\}$.*

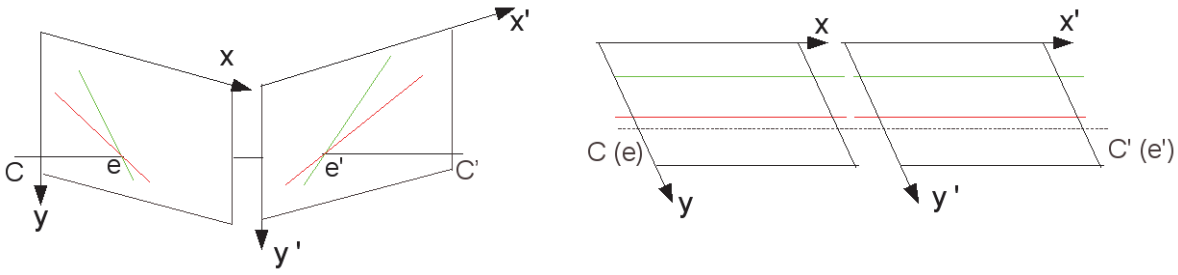


Figure 2.7: Rectification illustration. Left: original cameras configuration. Right: Camera configuration after rectification. Image planes are coplanar and their x-axis is parallel to the baseline $\mathbf{CC}'$.

2.7 Rectification

Image rectification is the process of re-sampling pairs of stereo images in order to produce a pair of "matched epipolar" projections. The rectification makes the corresponding epipolar lines coincide and be parallel to the x-axis. Consequently, the disparity between two images is only in the x-direction. A pair of stereo-rectified images is helpful for dense stereo matching algorithms. It restricts the search domain for each match to a line parallel to the x-axis.

2.7.1 Special Form of F

From the geometric viewpoint, the rectification is achieved when both cameras have their image planes coplanar and the x-axis of the image planes parallel to the baseline. This means that the motion between both cameras is a pure translation with no rotation. One can assume that the rectified camera matrices are $\mathbf{P} = \mathbf{K}\,[\mathbf{I} \mid \mathbf{0}]$ and $\mathbf{P}' = \mathbf{K}\,[\mathbf{I} \mid -\lambda\mathbf{i}]$ with $\mathbf{i} = \begin{pmatrix}1 & 0 & 0\end{pmatrix}^T$ (λ is the distance between the camera centers, called the baseline). Using (2.58) with $\mathbf{R} = \mathbf{I}$ and $\mathbf{T} = -\lambda\mathbf{i}$ we get

$$\mathbf{F} = -\lambda\mathbf{K}^{-T}[\mathbf{i}]_\times\mathbf{K}^{-1} \tag{2.64}$$

and after simplification (and ignoring the scale $-\lambda$)

$$\mathbf{F} = [\mathbf{i}]_\times\ . \tag{2.65}$$

Putting this special fundamental matrix in (2.51), we have $y = y'$, that is, the epipolar lines are corresponding raster lines and the disparity is in the x-direction.

2.7.2 Invariance

Note that the solution to the rectification is not unique. Once the rectification is achieved, we can rotate two cameras together around the baseline and the resulting images remain rectified. The ideal is to achieve the rectification by introducing a projective distortion as small as possible. A robust three-step image rectification method which implicitly minimizes the projective distortion, will be presented in Chapter 7.

Chapter 3

The Calibration Harp: a Measurement Tool of the Lens Distortion

Contents

Abstract *This chapter introduces a general tool for measuring the lens distortion of any camera, and for evaluating the performance of any camera calibration algorithm. Lens distortion is a non-linear deformation which deviates a pinhole camera from perspective projection. Since the preservation of point alignments characterizes a perspective projection, it is reasonable to measure the straightness of the projections of 3D straight lines to evaluate the lens distortion. To have a precise evaluation in practice, we need some very straight strings of good quality. It is relatively easy to ensure the straightness of strings by tightening them on a frame, while it is more delicate to choose an appropriate type of string. We tried four types of strings and found that the opaque fishing string was the best choice for our purpose. We therefore built an evaluation pattern made of several parallel opaque strings tightened on a wooden frame, which we called "calibration harp". The Devernay sub-pixel precision edge detector are used to extract the edge points in image, which are then associated to the line segments detected by LSD (Line Segment Detector). Finally, the distortion is evaluated as the root-mean-square (RMS) distance from the edge points belonging to a same line segment to their corresponding linear regression line.*

3.1 Introduction

For precise 3D stereo applications, lens distortion correction is a crucial step. Once a camera is calibrated, triangulation techniques permit a direct reconstruction of 3D scenes. But if an imprecise distortion model is used to correct the images, the residual distortion will be directly back-projected to the reconstructed 3D scene and distort globally the 3D scene. This imprecision can be a serious hindrance in remote sensing applications such as the early warnings of geology disasters, or in the construction of topographic maps from stereographic pairs of aerial photographs. Despite its importance, there is to the best of our knowledge no standard measurement for camera distortion correction. However, such measurements are implicit in the energy functionals minimized for distortion correction. In the literature, three kinds of distortion correction methods coexist by minimizing different error terms:

- classic pattern-based methods minimize the re-projection error;
- plumb-line methods minimize the straightness error of corrected lines;
- enlarged epipolar methods minimize the algebraic error in the estimate of the enlarged fundamental matrix.

In most classic pattern-based methods the lens distortion is estimated together with the camera internal and external parameters [159, 172, 191, 95, 177]. Hence their name of global camera calibration methods. These methods are mostly not pattern-free: they use photographs of a known planar or non-planar pattern containing simple geometric shapes. The corners or the centroids of these shapes are used as accurate control points. The global process finds the camera parameters minimizing the distance between the observed position of these points in the real image, and their position in the image simulated by retro-projection of the pattern model using the camera model. This is a non-linear problem with many parameters. So the result will be precise only if the model parameters capture the correct physical camera properties of cameras, and if the minimization algorithm finds a global minimum.

The second kind is the "plumb-line" method, which rectifies the distorted lines in images which contain the projection of 3D straight lines. The first paper proposing a "plumb-line" method seems to be Brown (1971, [25]). This idea has been applied to most distortion models: the radial model [4, 151, 145], the FOV (Field Of View) model [53], or the rational function model [38]. These methods minimize the straightness error of the corrected lines.

Recently, more attention has been paid to pattern-free methods (or self-calibration methods) where the estimation of the distortion is obtained without using any specific pattern. The distortion is estimated from the correspondences between two or several images in absence of any camera information. The main tool is the so-called enlarged epipolar constraint, which incorporates lens distortion into the epipolar geometry. Some iterative [161, 190] or non-iterative methods, for example the quadratic-eigenvalue problem (QEP) in [126, 68], the lifting method in [10], the companion matrix method in [108], the radial trifocal tensor in [168], the quadrifocal tensor in [168], the ratio function model in [39], the Gröbner basis method [104, 144, 32, 103, 97] are used to estimate the distortion and correct it. All of these methods minimize the algebraic error in the estimate of the enlarged fundamental matrix.

As we mentioned the minimized error depends on the method. This means that there is no common evaluation for the correction precision of the different methods. It seems desirable to propose an absolute distortion correction measurement, provided its simplicity and completeness proves faultless. Such an evaluation will be proposed here. It is based on the theorem in section 3.2. In section 3.3, we discuss how to build a pattern called "calibration harp" for this precision verification. Section 3.4 shows in practice how to use the photos of the calibration harp to compute the correction precision.

3.2 From Straight Lines to Straight Lines

In this section, we treat briefly the mathematical theory on which the evaluation tool is based. It is well known that the alignment is the only property preserved in the perspectivel projection. Even though the following theorem has been widely cited in computer vision papers [25, 4, 151, 145, 53], there is no proof for it in the literature.

Theorem 2 *Let* $\mathbf{T}$ *be a continuous transform from* $\mathcal{P}^3$ *to* $\mathcal{P}^2$ *(from 3D projective space to 2D projective plane). If there is a point* $\mathbf{C}$ *such that:*

(a) the images of any three point belonging to a line in $\mathcal{P}^3$ *not containing* $\mathbf{C}$*, are aligned points in* $\mathcal{P}^2$*;*

(b) the images of any two points belonging to a line in $\mathcal{P}^3$ *containing* $\mathbf{C}$*, are the same point in* $\mathcal{P}^2$*;*

(c) there are at least four points belonging to a plane not containing $\mathbf{C}$*, such that any group of three of them are non aligned, and their images are non aligned either;*

then $\mathbf{T}$ *is a perspective projection with center* $\mathbf{C}$*.*

This theorem provides us with a fundamental tool to verify that a camera (or virtual camera) follow the pinhole model. Some comments about the hypotheses are pertinent. Hypothesis

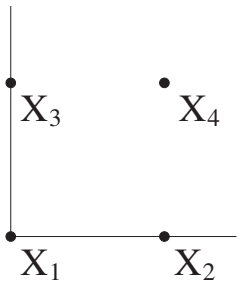
(a) is the main one: the transform maps lines into lines. However, this is limited to lines not passing through the camera center. Lines described by light rays entering the camera map into just one point on the image plane (when in focus), as required by hypothesis (b). Finally, hypothesis (c) is just needed to discard the degenerate case that brings the whole space $\mathcal{P}^3$ into one line.

Before giving the proof of the theorem, an auxiliary lemma needs to be stated and proved:

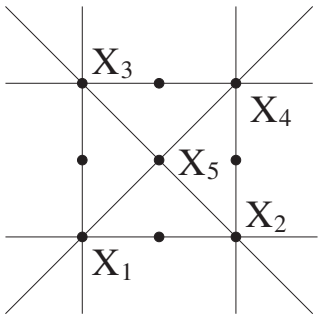
Lemma 1 *Let* $\mathbf{H}$ *be a continuous transform from* $\mathcal{P}^2$ *to* $\mathcal{P}^2$ *(projective plane). Assume that the image of any three aligned points are still aligned points, and that there are at least four points such that any group of three of them are non aligned and their images are non aligned either. Then* $\mathbf{H}$ *is a homography.*

Proof

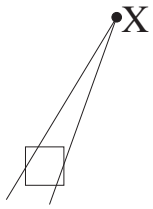
1. Assume, without loss of generality (by composing $\mathbf{H}$ with the adequate homography), that the four considered points $\mathbf{X}_1, \ldots, \mathbf{X}_4$ have homogeneous coordinates $(0,0,1)^T$, $(1,0,1)^T$, $(0,1,1)^T$ and $(1,1,1)^T$ (vertices of a unit square) and consider their images by $\mathbf{H}$. Any group of three of the four image points $\mathbf{H}(\mathbf{X}_1)$ to $\mathbf{H}(\mathbf{X}_4)$ are not aligned by hypothesis. Let $\mathbf{H}_0$ be the homography mapping $\mathbf{H}(\mathbf{X}_i)$ to $\mathbf{X}_i$ for all i. We want to show that $\mathbf{H}_0 \circ \mathbf{H} = \mathbf{I}$ (identity), proving that $\mathbf{H} = \mathbf{H}_0^{-1}$ is a homography. We will note $\mathbf{G} = \mathbf{H}_0 \circ \mathbf{H}$ in the rest of the proof.



2. By construction of $\mathbf{H}_0$, we have $\mathbf{G}(\mathbf{X}_i) = \mathbf{X}_i$. Since $\mathbf{G}$ preserves aligned points, the lines $x = 0$ and $x = 1$ are globally invariant through $\mathbf{G}$. So their intersection $\mathbf{Y}_\infty = (0,1,0)^T$ is fixed by $\mathbf{G}$. The same holds for $\mathbf{X}_\infty = (1,0,0)^T$. By intersecting the diagonals of the square, we see that the point $\mathbf{X}_5 = (\mathbf{X}_1 + \mathbf{X}_4)/2 = (\mathbf{X}_2 + \mathbf{X}_3)/2$ is also fixed. Now the horizontal and vertical lines through this point are globally invariant (since x_∞ and y_∞ are fixed) and intersecting with the edges of the square, we see that all points $(\mathbf{X}_i + \mathbf{X}_j)/2$ are fixed.



3. By the same reason, considering the four squares of respective diagonals $\mathbf{X}_i\mathbf{X}_5$ ($i = 1, \ldots, 4$) whose vertices are fixed as shown above, we see that all points $(3\mathbf{X}_i + \mathbf{X}_j)/4$ are fixed. By a recursion argument, all points of coordinates $(x, y, 1)^T$ with $0 \leq x, y \leq 1$ and x and y of finite binary expression are fixed. These points being dense in the unit square and by continuity of $\mathbf{G}$, we see that all points of this unit square are fixed.



4. Consider any point $\mathbf{X} = (x, y, z)^T \in \mathcal{P}^2$. $\mathbf{X}$ can always be seen as the intersection of two lines intersecting the *open* unit square. Each of this line containing thus two fixed points of $\mathbf{G}$, they are globally invariant by $\mathbf{G}$ and so $\mathbf{G}(\mathbf{X}) = \mathbf{X}$ by intersection. □

Proof of the theorem

1. Let us call π the plane in $\mathcal{P}^3$ containing the four points of hypothesis (c). Note that by hypothesis, π does not contain the point $\mathbf{C}$.

2. We construct an auxiliary perspective projection with center $\mathbf{C}$ defined by

$$\mathbf{Q} = \begin{pmatrix} 1 & 0 & 0 & -x_C \\ 0 & 1 & 0 & -y_C \\ 0 & 0 & 1 & -z_C \end{pmatrix}.$$

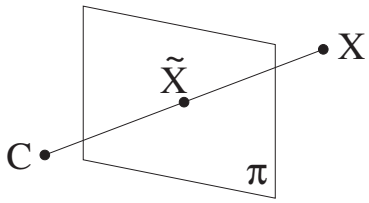
   We will call $\overline{\mathbf{Q}}$ the restriction of $\mathbf{Q}$ to the plane π. By construction, $\overline{\mathbf{Q}}$ is a continuous and bijective transformation from $\mathcal{P}^2$ to $\mathcal{P}^2$, and it maps aligned points into aligned points.

3. We will call $\overline{\mathbf{T}}$ the restriction of $\mathbf{T}$ to the plane π. By hypotheses, $\overline{\mathbf{T}}$ is continuous, transforms aligned points into aligned points, and there are at least four points such that any group of three of them are non aligned and their images are non aligned either.

4. Let us call $\mathbf{H}$ the transformation from $\mathcal{P}^2$ to $\mathcal{P}^2$ defined by $\mathbf{H} = \overline{\mathbf{T}} \circ \overline{\mathbf{Q}}^{-1}$. By construction, $\mathbf{H}$ satisfies the hypotheses of the lemma. Then, $\mathbf{H}$ is a homography.

5. Given a point $\mathbf{X}$ of $\mathcal{P}^3$, we call $\tilde{\mathbf{X}}$ the intersection of line $\overline{\mathbf{CX}}$ with plane π. By hypothesis (b), we know that $\mathbf{T}(\mathbf{X}) = \mathbf{T}(\tilde{\mathbf{X}})$. Then,

$$\mathbf{T}(\mathbf{X}) = \mathbf{T}(\tilde{\mathbf{X}}) = \overline{\mathbf{T}}(\tilde{\mathbf{X}}) = \mathbf{H} \circ \overline{\mathbf{Q}}(\tilde{\mathbf{X}}).$$



But, as $\mathbf{Q}$ is a perspective projection of center $\mathbf{C}$, and $\tilde{\mathbf{X}}$ belongs to the line $\overline{\mathbf{CX}}$,

$$\overline{\mathbf{Q}}(\tilde{\mathbf{X}}) = \mathbf{Q}(\mathbf{X}).$$

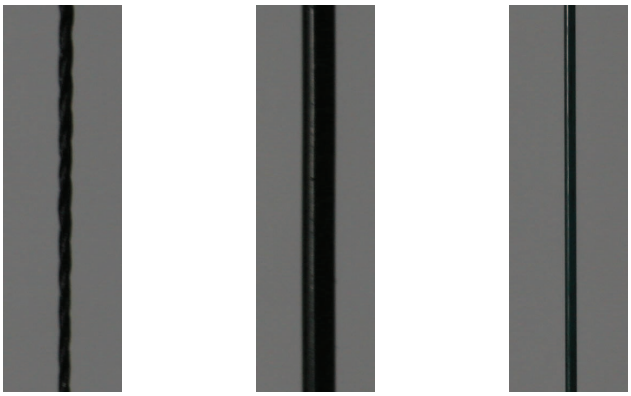
Finally, $\mathbf{T} = \mathbf{H} \circ \mathbf{Q}$ and is a perspective projection with center $\mathbf{C}$. □

The aim of distortion correction is exactly to bring a real camera back to a pinhole camera. Since the above theorem presents a necessary and sufficient condition, *it is enough to rectify all the observed distorted lines coming from real straight lines to obtain a pinhole camera.* Thus, compared to other methods, plumb-line methods seem to minimize the correct error. Yet, in the literature on plumb-line methods, there is surprisingly no detail about experiment setups: what kind of physical lines, how to ensure their straightness, how to photograph them, how to detect and extract them etc. Since all of these points are crucial to devise a precise evaluation tool, we will present them in the following sections.

3.3 Building the Calibration Harp

According to the above theorem, it is necessary to photograph physically guaranteed straight lines in the scene. But there are seldom absolutely straight lines in a physical scene. Probably the best opportunity would be given by a cable-stayed bridge. Its cables, which support the weight of the bridge, are therefore very tightly stretched. Nonetheless, it is still more convenient to dispose of a homemade pattern containing very straight lines. As a matter of fact, we discovered that it is relatively easy to build this kind of pattern. Any solid frame, for example a wooden frame like the one shown in Fig. 3.2 will do. Two dozens screws were planted on two opposite sides of the frame. Finally we tightened (manually) the strings and fixed them with the aid of the screws to ensure the straightness of the strings. This pattern looks like the musical instrument harp, hence the "calibration harp" name. Such a frame can be built in less than one hour with only the wooden frame, the screws, a screwdriver and the strings. Nevertheless the quality of the strings matters. Four different strings were tried: sewing string, tennis racket string, a transparent fishing string and an opaque fishing string. The sewing strings are not very smooth and their thickness oscillates in a braid

pattern along the strings, due to their twisted structure (see Fig. 3.1a). Among the four types of strings, the tennis racket string (Fig. 3.1b), the transparent fishing string and the opaque fishing string (Fig. 3.1c) are apparently more smooth than the sewing string and have a more uniform thickness. But, as we shall see, the tennis racket strings are rigid and would require an extreme tension to become straight. The fishing strings are both smooth and flexible, thus can be easily tightened to be very straight on the wooden frame (see Fig. 3.2). But the transparent fishing string behaves like a lens and is therefore not adequate. So the opaque fishing string turns out to be the best choice to build the calibration harp.



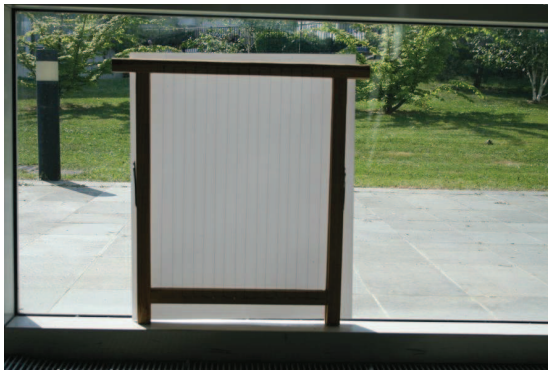
(a) (b) (c)

Figure 3.1: The quality of lines. (a) sewing line. (b) tennis racket line. (c) opaque fishing line.

To ensure the precision of the edge detection in the string images, a uniform background whose color contrasts well with the string color must be preferred. The first idea was to use a uniform wall as background. However, the projected shadows of the strings on the wall were a nuisance which could cause the edge detector failure (see Fig. 3.2a and 3.2c). The uniform wall would therefore need to be be far away from the harp, which is not easy to realize. Next to a uniform wall, the sky is the only distant and uniform background that comes into mind. Yet, in the experiments, we found that it was difficult to take photographs of the harp against the sky. When the angle of view of the camera is large, it is difficult to photograph only the sky and to avoid the interference of buildings, trees, etc. in the photos. In addition, even if at first sight the sky looks uniform, it turns out to be often inhomogeneous: see Fig. 3.3a). The final solution was to fix a translucent paper on the back of the harp and to place the frame back in front of any natural or artificial back light (see Fig. 3.2b and 3.2d for the harp with the translucent paper). This setup permits to take photos anywhere, provided the back of the harp is sufficiently lit.

Remark that the above mentioned theorem only addresses the geometric aspects of image acquisition, and does not consider other necessary aspects of image formation: lens blur, motion blur, aliasing, noise, vignetting, etc. In reality, the photographs of strings are always perturbed by such effects. To lessen the optical aberration as much as possible, the photo must be taken under a controlled condition. In our experimental setups, a Canon EOS 30D reflex camera was installed on a tripod with 10 seconds timer to avoid hand shakes and motion blur. The camera could be rotated on the tripod to take photos with varying orientations, but

(a) The harp with an uniform opaque object as background



(b) The harp with a translucent paper as background

(c) A close-up of the harp with an uniform opaque object as background

(d) A close-up of the harp with a translucent paper as background

Figure 3.2: The harp with an opaque object or a translucent paper as background. (a) The harp with an uniform opaque object as background (see a close-up in (c)). (b) The harp with a translucent paper as background (see a close-up in (d)). Shadows can be observed in (a) and (c), while there is no shadow in (b) or (d).

keeping the camera plane parallel to itself and the same distance to the harp. Compared to the photos of sewing strings taken by hand against the sky (Fig. 3.3a), the photos of opaque fishing strings with a translucent paper (Fig. 3.3b) have a more uniform background. In addition, the images taken by hand (Fig. 3.3a) suffer from inhomogeneous blur or variation of strings thickness caused by the inconstant distance from camera to the harp or the hand motion, while the images taken by tripod (Fig. 3.3b) have a better quality.

(a) The photo of the harp is taken by hand against the sky

(b) The photo of the harp is taken against a translucent paper by a tripod

Figure 3.3: The quality of photos depends on the harp, its background and also the stability of camera for taking photos.

3.4 Line Segment Detection and Edge Points Extraction

In this section, we describe how to evaluate numerically the correction precision. Assume the distortion in the photos of calibration harp is estimated and corrected by a certain distortion correction method. According to the above theorem, whether a camera is a pinhole camera or not can be evaluated by judging whether the corrected lines are straight or not. The straightness of a line is defined as the root-mean-square (RMS) distance from the edge points to their corresponding linear regression line. To compute the RMS distance, we need to extract the edge points of the distorted lines from the images in sub-pixel precision. Briefly, the lines are first detected by the LSD algorithm which groups the pixels having coherent gradient direction into line support regions [174]. In each validated line support region, Devernay's algorithm [52] is used to extract the edge points at sub-pixel precision. Finally, a 1D Gaussian convolution followed by a sub-sampling is performed on the extracted edge points to reduce the detection and aliasing noise left by this detection, without altering significantly the global distortion to be corrected.

3.4.1 Line Detection

LSD is a linear-time line segment detector that gives accurate results, controls its own false detection rate, and requires no parameter tuning [174]. The algorithm starts by computing the gradient direction at each pixel to produce a level-line field, i.e., a unit vector field such that all vectors are tangent to the level line going through their base point. Then, this field is

segmented into connected regions of pixels that share the same level-line angle up to a certain tolerance (see Fig. 3.4). These connected regions are called line support regions. Each line support region (a set of pixels) is a candidate for a line segment, which is then validated by *a contrario* approach and the Helmholtz principle proposed in [50, 51].

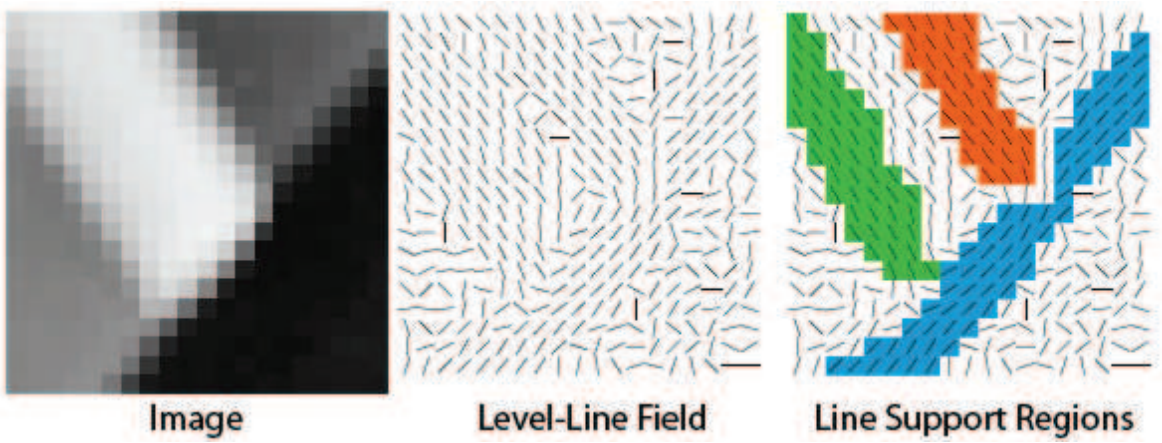


Figure 3.4: LSD algorithm.

3.4.2 Devernay's Detector

The LSD algorithm gives a validated line support region associated to a line segment, which groups a set of pixels sharing the same gradient orientation up to some toleration. Devernay's detector [52] is then used to extract the edge points of the line segments with sub-pixel precision in each validated line support region. On good quality images (SNR larger than 100), Devernay's detector can attain a precision of about 0.05 pixels. The implementation of Devernay's detector is very simple since it is derived from the well-known Non-Maxima Suppression method [33, 49]. It can be recapitulated in the following steps.

1. Let a point (x, y), where x and y are integers and $I(x, y)$ the intensity of pixel (x, y).
2. Calculate the gradient of image intensity and its magnitude in (x, y).
3. Estimate the magnitude of the gradient along the direction of the gradient in some neighborhood around (x, y).
4. If (x, y) is not a local maximum of the magnitude of the gradient along the direction of the gradient then it is not an edge point.
5. If (x, y) is a local maximum then estimate the position of the edge point in the direction of the gradient as the maximum of an interpolation on the values of the gradient norm at (x, y) and the neighboring points.

In step 3, the magnitude of the gradient along the direction of the gradient at points (x, y) is computed by linearly interpolating the closest points in the 3×3 neighborhood of the point (x, y) (see Fig. 3.5). In step 5, if the gradient magnitude of (x, y) is a local maximum,

it is considered as a good edge point. Then its position is refined by a simple quadratic interpolation of the values of the gradient magnitude between the 3 values in the gradient direction. The quadratic function of gradient magnitude along the gradient direction can be written as

$$f(l) = al^2 + bl + c \tag{3.1}$$

with l the distance to the point (x, y) and a, b and c unknown parameters. In Fig. 3.5, three points A, B and C is sufficient to compute a, b and c. Then the offset l_0 of the refined edge point to the point (x, y) can be obtained by computing the derivative of $f(l)$ and setting it to zero: $\frac{df(l)}{dl} \mid_{l=l_0} = 0$.

Remark that the sub-pixel refinement of Devernay's detector is similar to the one of the SIFT method [117] except that SIFT works on the Laplacian value and uses a two-dimension quadric interpolation, while Devernay's detector works on the magnitude of gradient and uses a one-dimensional quadric interpolation in the gradient direction.

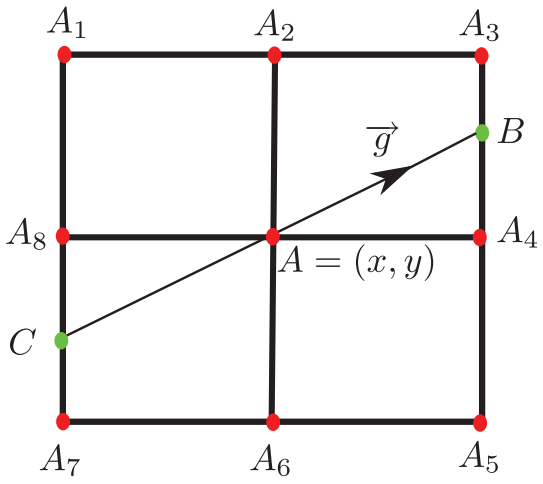


Figure 3.5: Devernay sub-pixel precision edge detector. The vector $\overrightarrow{g}$ is the gradient direction at the point (x, y). The gradient magnitude at point B is linearly interpolated by the gradient magnitude at point A_3 and A_4. Similarly, the gradient magnitude at point C is linearly interpolated by the gradient magnitude at point A_7 and A_8. If point A has a gradient magnitude larger than B and C, it is considered as a good edge point. Its position is refined by computing an offset through a quadric interpolation along the direction of $\overrightarrow{g}$.

3.4.3 Convolution and Sub-Sampling of Edge Points

For the photos of strings, almost every pixel along each side of one string is detected as an edge point at sub-pixel precision. So there are about 1000 edge points detected for a line of length about 1000 pixels. This large number of edge points opens the possibility to further reduce the detection and aliasing noise left by the detection through a convolution followed by a sub-sampling. A Gaussian blur of about $0.8 \times \sqrt{t^2 - 1}$ is needed before a t-subsampling to avoid aliasing [132]. We have two one-dimensional signals (x-coordinate and y-coordinate

of edge points) along the length of the line. The Gaussian convolution is performed both one-dimension signals, parameterized by the length along the edge. This is done by a uniform re-sampling by length along the line. To ensure a high accuracy, the sampling step is m times smaller than the average distance between two adjacent edge points. A linear interpolation is used to accelerate the re-sampling (see Fig. 3.6). Assume the distance between two adjacent edge points (x_1, y_1) and (x_2, y_2) is l and the re-sampling step is $d \simeq 1/m$. Then the re-sampled point (x', y') can be expressed as

$$\begin{aligned} x' &= \frac{d}{l}(x_2 - x_1) + x_1 \\ y' &= \frac{d}{l}(y_2 - y_1) + y_1. \end{aligned}$$

Once the line is re-sampled, the Gaussian blur $0.8 \times \sqrt{t^2 - 1}$ can be applied and is followed by a sub-sampling with factor mt on the x and y coordinates separately (the sub-sampling factor is mt because the re-sampling step is m times smaller than the average distance between two adjacent edge points).

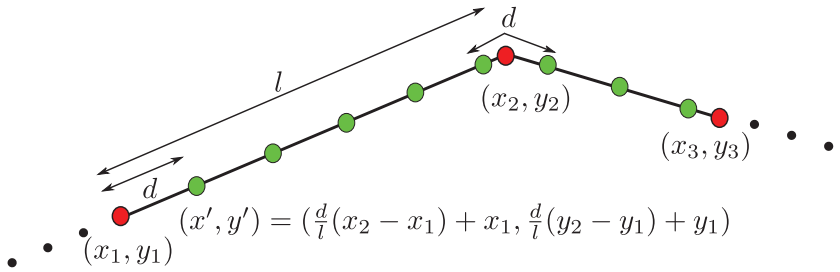


Figure 3.6: Line re-sampling. The red points (x_1, y_1), (x_2, y_2), $\cdots$ are the edge points extracted by Devernay's detector. They are irregularly sampled along the line. The re-sampling (in green) is along the length of the line with the uniform step d. The linear interpolation is used to compute the re-sampled point fast.

3.4.4 Computation of Straightness

The straightness of a line is measured as the root-mean-square (RMS) distance from its edge points to its global linear regression line. Given a set of corrected edge points $(x_{u_i}, y_{u_i})_{i=1,\cdots,N}$ which are supposed to be on the image of a straight line, one can compute the linear regression line by

$$\alpha x_{u_i} + \beta y_{u_i} - \gamma = 0 \tag{3.2}$$

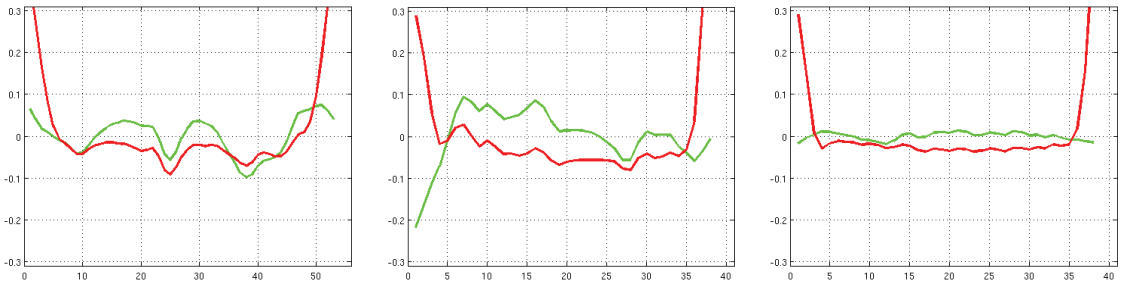
with $\tan 2\theta = -\frac{2(A_{xy} - A_x A_y)}{V_{xx} - V_{yy}}$, $\alpha = \sin\theta$, $\beta = \cos\theta$, $A_x = \frac{1}{N}\sum_{i=1}^{N} x_{u_i}$, $A_y = \frac{1}{N}\sum_{i=1}^{N} y_{u_i}$, $A_{xy} = \frac{1}{N}\sum_{i=1}^{N} x_{u_i} y_{u_i}$, $V_{xx} = \frac{1}{N}\sum_{i=1}^{N}(x_{u_i} - A_x)^2$, $V_{yy} = \frac{1}{N}\sum_{i=1}^{N}(y_{u_i} - A_y)^2$ and $\gamma = A_x \sin\theta + A_y \cos\theta$. Then the straightness measure is computed as

$$S = \sqrt{\frac{\sum_{i=1}^{N} (\alpha x_{u_i} + \beta y_{u_i} - \gamma)^2}{N}}. \tag{3.3}$$

3.5 Experiments

In this section, we just show some results to support our argument that the opaque fishing string is more appropriate to evaluate the correction precision. A good string should not have other imperfection aspects which introduce some error susceptible to be mixed with the lens distortion. We hope that once the distorted line is ideally corrected by a certain correction method, its straightness only reflects the correction performance, but is not affected by other factors.

In Fig. 3.7, the high frequency of the distorted sewing string, the distorted tennis racket string and the distorted opaque fishing string are compared to the straightness error of their corresponding corrected strings. The almost superimposing high frequency oscillation means that the high frequency of the distorted strings is not changed by the lens distortion correction. In such a case, the straightness error includes the high frequency of the distorted strings and does not really reflect the correction performance. So it is better to use the string which contains a high frequency oscillation as small as possible. Among the three types of strings, the opaque fishing string shows the smallest such oscillation. The larger oscillation of the sewing string is due to a variation of the thickness related to its twisted structure, while the tennis racket string is simply too rigid to be stretched, even if this is not apparent in Fig. 3.1b).



(a) The sewing string

(b) The tennis racket string

(c) The opaque fishing string

Figure 3.7: The small oscillation of the corrected lines is related to the quality of the strings. The green curve shows the RMS distance (in pixels) from the edge points of a corrected line to its regression line. The red curve shows the high frequency of the corresponding distorted line. The corrected line inherits the oscillation from the corresponding distorted line. (a) the sewing string. (b) the tennis racket string. (c) the opaque fishing line. The x-axis is the index of edge points. The range of y-axis is from -0.3 pixels to 0.3 pixels.

Chapter 4

Non-Parametric Lens Distortion Correction

Contents

Abstract *This chapter points out and attempts to remedy a serious discrepancy in results obtained by global calibration methods: The re-projection error can be rendered very small by these methods, but we show that the optical distortion correction is far less accurate. This discrepancy can only be explained by internal error compensations in the global methods that leave undetected the inadequacy of the distortion model. This fact led us to design a model-free distortion correction method where the distortion can be any image domain diffeomorphism. The obtained precision compares favorably to the distortion given by state of the art global calibration and reaches a RMSE of* 0.08 *pixels. This non-parametric method requires a flat and textured pattern. By using the dense matchings between the pattern and its photo, a distortion field can be obtained by triangulation and local affine interpolation. The obtained precision compares favorably to the distortion given by state of the art global calibration and reaches a RMSE of* 0.08 *pixels. The non-flatness of the pattern is a limitation of this method and can introduce a systematic error in the distortion correction. Nonetheless, we also show that this accuracy can still be improved.*

4.1 Introduction

This chapter presents a small step forward in a research programme whose aim is to devise a highly accurate camera calibration method. By highly accurate, we mean that the residual error between the camera and its numerical model obtained by calibration should be far smaller than the pixel size. At first sight, this problem seemed to have been solved adequately by recent global calibration methods. The celebrated Lavest et al. method [95] measures the non-flatness of a pattern and yields a remarkably small re-projection error of about 0.02 pixels, which outperforms the precision of other methods. The experiments described below will actually confirm this figure. For the goals of computer vision, this precision would be more than sufficient. Yet, this chapter describes a seriously discrepant accuracy measurement contradicting this hasty conclusion. According to the measurement tool of distortion correction precision developed in Chapter 3, the only objective and correct criterion is straightness of corrected lines.

Following this tool, the accuracy criterion used herewith directly measures the straightness of corrected lines. We shall see that this straightness criterion gives a RMSE in the order of 0.2 pixel, which contradicts the 0.02 pixel re-projection accuracy. This significant discrepancy means that, in the global optimization process, errors in the external and internal camera parameter are being compensated by opposite errors in the distortion model. Thus, an inaccurate distortion model can pass undetected [177, 108]. Such facts raise a solid objection to global calibration methods, which estimate simultaneously the lens distortion and the camera parameters. This chapter reconsiders the whole calibration chain and examines an alternative way to guarantee a high accuracy. A useful tool toward this goal will be proposed and carefully tested. It is a direct nonparametric distortion correction method. By nonparametric, we mean that the distortion model allows for any diffeomorphism.

We shall follow the usual assumption that a real camera deviates from the ideal pinhole model [89] by a lens distortion [25]. Thus, distortion correction is a fundamental step in multi-view geometry applications such as 3D reconstruction. The above mentioned error measurement discrepancy may explain why three categories of distortion correction methods still coexist:

- classic pattern-based methods, by minimizing the re-projection error;
- plumb-line methods, by minimizing the straightness error of corrected lines;
- enlarged epipolar methods, by minimizing the algebraic error in the estimate of enlarged fundamental matrix.

Traditionally, in classic pattern-based methods, the lens distortion is estimated together with the camera internal and external parameters [159, 172, 191, 95, 177]. So we call these methods global camera calibration methods. These methods usually are not blind and need a known planar or non-planar pattern which contains simple geometric shapes. The corners or the centroid of these shapes are used as accurate control points. The global process finds the camera parameters minimizing the distance between the observed position of these points in the real image, and their position in the image simulated by retro-projection of the pattern model using the camera model. By ignoring the distortion, this problem can be approximated by a linear minimization problem [1, 84, 172, 125, 66, 77, 179], which estimates the the camera projection matrix from 3D to 2D (a singular case occurs, however, when the pattern is planar, where only a 3×3 homography can be estimated). These methods are implicit because the entries of the 3×4 projection matrix do not have a direct physical meaning and are useless in camera modeling. Some decomposition methods have been proposed to extract the camera intrinsic parameters [125, 191]. The advantage of these linear methods is that a closed-form solution can be derived. But the non-linear lens distortion cannot be incorporated in the formulation and thus the obtained precision is usually limited. To solve this problem, several non-linear methods have been proposed which include lens distortion parameters in the camera model [26, 1, 179, 60, 159]. This kind of method can *a priori* model any kind of camera any and any lens distortion. But the result will be precise only if the model parameters capture the correct physical camera properties of cameras, and if the minimization algorithm finds a global minimum. The Levenberg-Marquardt algorithm [107, 123] (see Appendix A.3) is generally used in this situation to perform the non-linear minimization. Therefore a global convergence cannot be guaranteed unless a good initialization point is chosen. To avoid local minima in the non-linear minimization, a two-step strategy is often used. The closed-form solution is first found by a linear method. It is hereafter refined in a non-linear optimization adding the lens distortion parameters [172, 177]. Nevertheless, global camera calibration methods suffer a common drawback: errors in the external and internal camera parameter can be compensated by opposite errors in the distortion model. Thus the residual error can be small while the distortion model is not that precise [177, 108].

To avoid this error compensation between camera parameters, the general idea is to correct first only the lens distortion, without estimating the other camera parameters. The first paper about lens distortion correction is perhaps Brown's [25], which is based on the fact that the image of 3D line must be straight in the 2D image if the camera is a pinhole camera (no lens distortion) [53, 89]. This idea was also used in other works for various distortion models: a radial distortion model in [4, 151, 145], a FOV (Field Of View) model in [53], a rational function model in [38]. These methods are often called *plumb-line* methods, because they require the detection in the image of curves which are images of straight lines.

Recently more attention has been paid to estimate the distortion without specific patterns. The distortion is estimated from the correspondences between two or several images without

knowing any camera information. The main tool used here is enlarged epipolar constraints, which incorporate lens distortion into the epipolar geometry. Some iterative [161, 190] or non-iterative method, for example, quadratic-eigenvalue problem (QEP) in [126, 68], lifting method in [10], companion matrix method in [108], radial trifocal tensor in [168], quadrifocal tensor in [168], ratio function model in [39], Gröbner basis [104, 144, 32, 103, 97], are used to estimate the distortion and correct it. These methods are blind but parametric, and depend on the *a priori* choice of a distortion model with a fixed number of parameters. This *per se* is a drawback: such calibration methods require several trials and a manual model selection. Most methods assume a radial distortion modeled as a low-order even polynomial [172, 176] (other kinds of models are used for wide-angle or fish-eye lens [99, 53]).

The distortion correction method proposed here does not belong to any of the above mentioned three categories. Indeed, it is non-parametric, non-iterative, and model-free. Like most methods in the second and third category, the method decouples the distortion estimation from the calibration of camera internal parameters, thus avoiding any error compensation between them.

Some non-parametric methods are also proposed in literature which do not require an explicit distortion model. In [162], the distortion is reconstructed from images of spheres using the fact that spheres must appear circular in ideal stereographic projection.

Our plan is as follows. Section 4.2 gives the necessary definitions of the real camera and the pinhole model. It explains why a distortion correction up to a homography is sufficient for 3D applications, and defines the concept of *virtual pinhole camera*. The proposed nonparametric distortion correction is detailed in section 4.3, and is followed by experimental results in section 4.4. The last section 4.5 discusses how the high accuracy quest could be pursued.

4.2 The Virtual Pinhole Camera

By pinhole camera model we mean a distortion-free camera model

$$C := KR[I \mid -T]. \tag{4.1}$$

where T is a 3D point representing the optical center of camera, R is a 3D rotation representing camera orientation, K is a 3×3 upper triangular matrix containing the camera internal parameters. The classic camera model is

$$\hat{C} := K\hat{\mathcal{D}}R[I \mid -T] \tag{4.2}$$

where $\hat{\mathcal{D}}$ is a diffeomorphism of the image domain representing the non-linear lens distortion. In physical viewpoint, the distortion applies before the calibration matrix K. However, here we consider $K\hat{\mathcal{D}}$ together as a whole distortion and the proposed nonparametric method just corrects the distortion by recovering a 3×4 projective matrix without physical meaning. So we are free to put the distortion operator before K. This gives us another camera model:

$$\hat{C} := \mathcal{D}KR[I \mid -T] \tag{4.3}$$

In fact, $\mathcal{D}$ and $\hat{\mathcal{D}}$ are equivalent up to a normalization induced by K. To make the following presentation more clear, we continue to use the model in Eq. (4.3).

The nonparametric method will estimate the distortion up to an arbitrary invertible homography H: $\tilde{\mathcal{D}} = \mathcal{D}H$. The correction precision evaluation will be based on the straightness of corrected lines, which is preserved by any homography. Applying $\tilde{\mathcal{D}}^{-1}$ on $\hat{C}$, yields

$$\begin{aligned}\tilde{C} &= \tilde{\mathcal{D}}^{-1}\mathcal{D}KR[I\,|-T] \\ &= H^{-1}\mathcal{D}^{-1}\mathcal{D}KR[I\,|-T] = H^{-1}KR[I\mid -T]. \end{aligned} \tag{4.4}$$

Thus, inverting the distortion on all images produced by the camera yields a new camera model which becomes pinhole. H, K being invertible, the decomposition $H^{-1}K = \tilde{K}R'$ is unique by QR decomposition. So after distortion correction (up to a homography) we have $\tilde{C} = \tilde{K}R'R[I \mid -T] = \tilde{K}\tilde{R}[I \mid -T]$, which we call the *virtual pinhole camera obtained after distortion correction*. The orientation and internal parameters of this virtual model do not match the physics of the actual camera, but yield a virtual pinhole camera that can be used to the very same purposes. Indeed, consider several positions of the physical camera inducing as many camera models $C_i = \mathcal{D}KR_i[I \mid -T_i]$. Applying the correction $\tilde{\mathcal{D}}^{-1}$ to all images obtained from these camera positions yields virtual pinhole cameras $\tilde{C}_i = \tilde{K}\tilde{R}_i[I \mid -T_i]$, which maintains the same relative orientations. From these cameras the whole 3D scene can be reconstructed by standard methods, up to a 3D similarity.

4.3 Nonparametric Distortion Correction

4.3.1 The Experimental Set Up

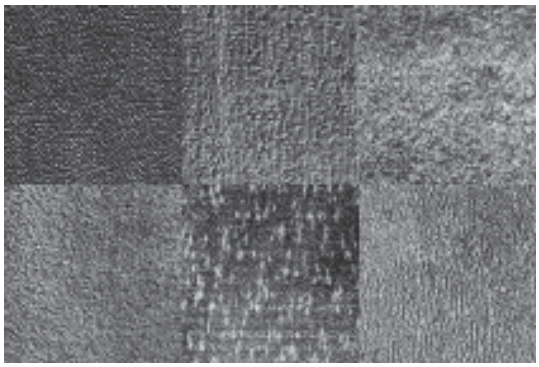
A nonparametric method requires the use of a highly textured planar pattern containing a dense feature point set, obtained by printing a textured image and pasting it on a very flat object (a mirror was used in the experiments). Two photographs of the pattern (Fig. 4.1) are taken by the camera in a fixed lens configuration (i.e. with fixed zoom and focus). Ideally, the whole captor must be covered by the whole pattern.

4.3.2 Feature Points

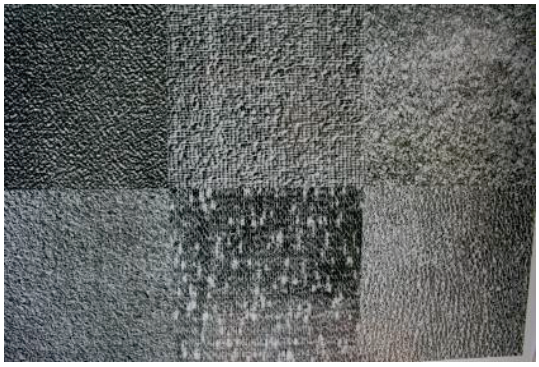
The distortion is estimated (up to a homography) as the diffeomorphism mapping the original digital pattern to its photograph. This requires a dense registration, which is obtained by the SIFT method [117]. Denote by I the original digital pattern, by P the printed pattern, and by v the photograph of P. The set of SIFT points matching from I to v is denoted by I_v and the corresponding points in v by v_I. Since the pattern is planar, there is a planar homography H such that $\mathcal{D}HI_v = v_I$. Knowing $\mathcal{D}H$ permits to synthesize a virtual pinhole camera by applying $(\mathcal{D}H)^{-1}$ on $\hat{C}$, as shown in Eq. (4.4).

Acceleration of the Matching Process The SIFT method extracts a rather dense set of feature points in textured images such as Fig. 4.1. This makes the matching process time consuming. To make the algorithm practical, some acceleration is required.

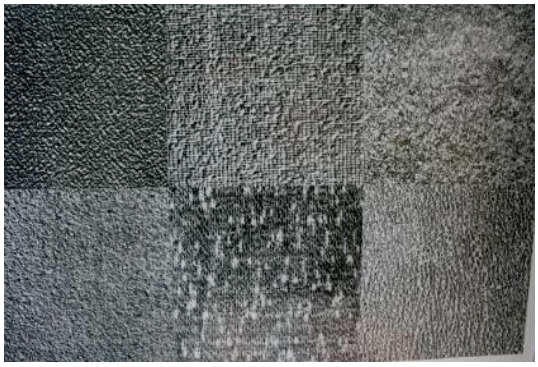
Assume that there are M and N SIFT feature points in the left and right image respectively. According to the SIFT matching protocol, each feature point in the left image is



(a)



(b)



(c)

Figure 4.1: (a) digital texture pattern: 1761×1174 pixels. (b) and (c) two similar photographs of the flat pattern.

compared to each feature point in the right image by the Euclidean distance between corresponding descriptors. If the ratio between the distance to the nearest feature point and the distance to the second nearest feature point is smaller than 0.6, the nearest feature point is considered as a valid correspondence. So the complexity of the matching process is $O(MN)$. In our experiment, typical $M \approx 10000$ and $N \approx 30000$, which makes the matching process too long. But the complexity can be reduced to $O(MP)$ if a feature point in the left image is guided to compare only P ($P << N$) feature points in the right image. The geometric constraints can be used to guide the matching process. If there is no distortion in images, the relation between two images is described, depending on the scene and the camera motion, either by a homography or the epipolar geometry. This geometry constraint can be reliably estimated using only a few most stable matching points by using only few robust SIFT key points, namely those whose Laplacian is highest. According to our tests on the SIFT key point precision, the matching error ε of the most stable feature points is less than 0.1 pixels. So in the absence of lens distortion, the estimated geometric constraint can guide precisely a feature point in the left image to find its correspondence in the right image, with a localization error of ε.

In the case where both images are related by a homography (flat pattern), a feature point p is projected to Hp where H is the homography estimated from few most stable SIFT matchings. Then the corresponding point of p can be found by applying the SIFT matching protocol on all the feature points q in a ε-neighborhood of Hp in the right image (see Fig. 4.2a).

In the general case where only an epipolar constraint can be estimated, the procedure is similar. A feature point p is projected to the epipolar line Fp where F is the fundamental matrix, estimated from the few most stable SIFT matchings. The corresponding point of some key point p belonging to the ε-neighborhood of a point q can be searched near the epipolar line Fp. So the search area becomes a rectangle-like domain (see Fig. 4.2b).

In the proposed method, photographs of a flat textured pattern are taken by a camera. The introduced distortion is pretty large (see Fig. 4.14a for example). Even though the homography is estimated by the most stable matchings, it can only predict the approximate corresponding point location, up to the distorsion error. So the search area must be enlarged. In the experiments, the value of ε was relaxed to 30 pixels to cope with the distortion. This is enough for most consumer cameras. For the efficiency of implementation, the ε-neighborhood was replaced by a square with size 2ε and the image domain was divided into many small squares (see Fig. 4.2c). The procedure is similar to the one without lens distortion. Once p is projected to Hp in the right image, the index of square (i, j) can be quickly recognized. Then the candidate corresponding points of p are all the feature points q in 9 squares of index (m, n) with $m = i - 1, i, i + 1$ and $n = j - 1, j, j + 1$. Then the SIFT matching protocol is applied on these candidates to find the corresponding point. The search area is larger than necessary. In fact, if Hp lies at the center of the square (i, j), it is sufficient to take all the feature points in the square (i, j) as candidates. But the broader search area ensures that a feature point can always find its corresponding point if it has one.

Assume the whole image domain is divided into D small squares and the feature points are uniformly distributed in the image domain. Then the complexity is reduced from $O(MN)$ to $O(9MN/D)$ with a gain factor $D/9$. Typically, for an image of size 1761×1174 pixels, we have about 600 squares with size 60 pixels. So the gained factor is about 60. A still better complexity reduction can be obtained if D is bigger (the squares smaller). But this requires

a more precise prior information about lens distortion. Lowe's SIFT matching protocol is based on the nearest neighbor distance ratio. With the acceleration, each feature point in the left image is only compared to a small subset of the feature points in the right image. This relaxes the SIFT matching protocol but can introduce more "outlier" matchings. An outlier elimination procedure will be explained in sections 4.3.4 and 4.3.5.

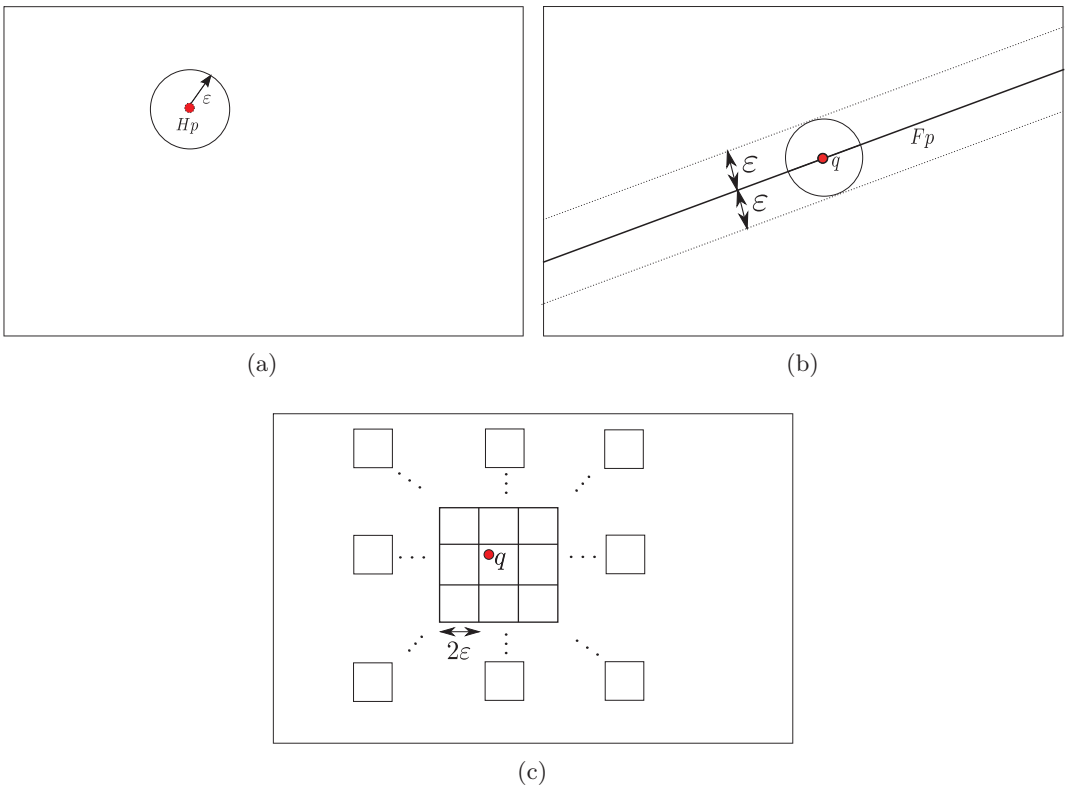


Figure 4.2: (a) homography case: the corresponding point of p can be found by applying the SIFT matching protocol to all the feature points in a ε-neighborhood of Hp. (b) epipolar geometry case: the corresponding point of p can be found by applying the SIFT matching protocol on all the feature points in a rectangle neighborhood defined by ε and the epipolar line Fp. (c) the image domain is divided into many square areas with size 2ε

4.3.3 Triangulation and Affine Interpolation

The correspondences (I_v, v_I) actually only define the distortion field $\mathcal{D}H$ on the SIFT points I_v. The distortion being very smooth and the SIFT points dense enough, an affine interpolation is sufficient to interpolate the distortion field. This interpolation is performed after the image domain has been partitioned by a Delaunay triangulation of the SIFT points in v and I respectively (Fig. 4.3). Remark that the triangulation is only performed on the SIFT points in one of the images. The same triangulation is just mapped to the other image by SIFT matchings (see Fig. 4.10 for example). A drawback of Delaunay triangulation is that

the triangles at the image border are elongated. This contradicts the assumption that the distortion can be locally approximated by an affine transform. Thus the estimated distortion field at the border can be imprecise. Fig. 4.1 shows the texture pattern, selected to yield a maximal density of reproducible SIFT points at fine scales.

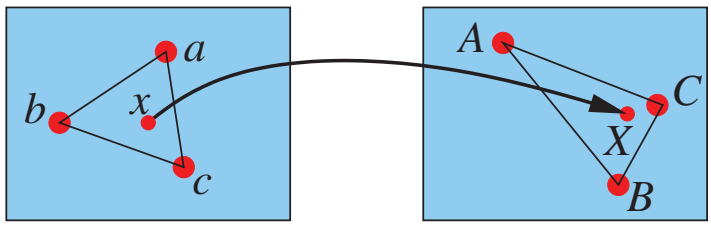


Figure 4.3: Local approximation of distortion by the affine transformation between corresponding Delaunay triangles. Point x is mapped to point X by the affine transformation that maps the triangle abc to the triangle ABC. (a, A), (b, B) and (c, C) are three pairs of correspondences.

4.3.4 Outliers Elimination: a Loop Validation

The few wrong SIFT matches (*outliers*) are nonetheless a serious problem (see Fig. 4.5 for example). In our case, precisely because of the lens distortion, matching points are not related by a homography, and directly applying RANSAC [67] would not work. The problem can be solved by a procedure which we shall call *loop validation* (Fig. 4.4). It consists of taking two similar photographs of the pattern u and v (obtained by moving slightly the camera between two successive snapshots) instead of one. With a straightforward notation we have $u_I = \mathcal{D}H_u I_u$ and $v_I = \mathcal{D}H_v I_v$ (since the same camera and configuration are used, $\mathcal{D}$ does not change). The points v_I can be projected back on I by the distortion field from u to I, obtaining $I_{uv} = (\mathcal{D}H_u)^{-1} v_I$. It follows that I_v and I_{uv} are related by a homography (without distortion) because

$$I_{uv} = (\mathcal{D}H_u)^{-1}\mathcal{D}H_v I_v = H_u^{-1} H_v I_v. \tag{4.5}$$

This homography can be estimated by the RANSAC algorithm and all the outliers not compatible with the homography are eliminated.

4.3.5 Vector Filter

The loop validation eliminates most outliers. But there still are special cases where this elimination is not complete. Given two very similar photos of the pattern (see Fig. 4.1 for example), u and v, it is possible that some matchings of I and u are similar to those of I and v. Assume the acceleration procedure makes I_v to match a the wrong point $v'_I = Tv_I = T\mathcal{D}H_v I_v$ where T is the translation moving the good correspondence v_I to v'_I. Then the loop validation sends the position v'_I to the image u and projects it back to the image I, obtaining I'_{uv}. Since images u and v are very similar, sometimes the local mapping from I to u around the position v'_I is also incorrect, having the form of $T\mathcal{D}H_u$. For the correspondences (I_v, v'_I) and (v'_I, I'_{uv}) both having the incorrect local mapping, we finally obtain $I'_{uv} = (T\mathcal{D}H_u)^{-1} v'_I =$

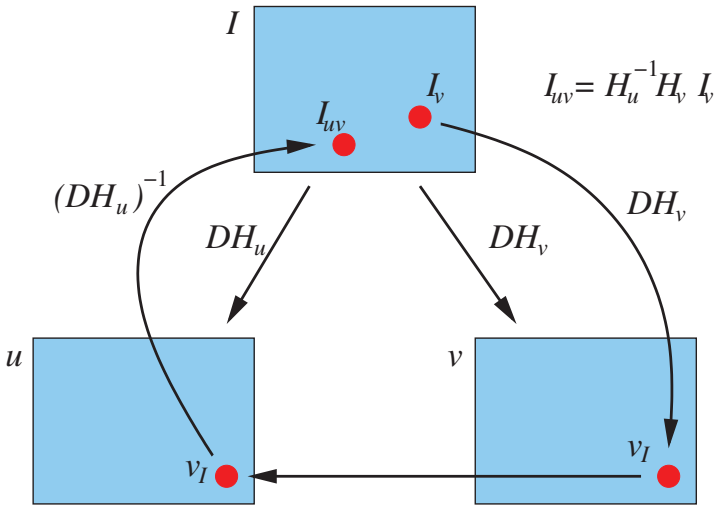


Figure 4.4: The loop validation used to remove outliers.

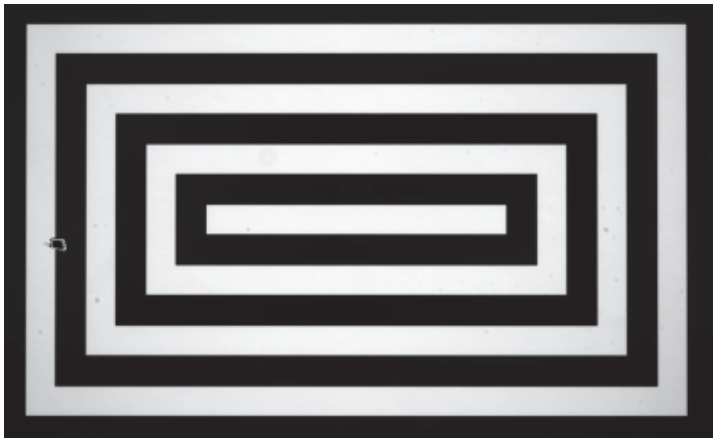


Figure 4.5: Even a single false match can cause a conspicuous artifact in the corrected image.

$(T\mathcal{D}H_u)^{-1}T\mathcal{D}H_vI_v = H_u^{-1}H_vI_v$. So I_v and I'_{uv} are also related by the homography $H_u^{-1}H_v$. This means that some "outliers" are invisible for loop validation if images u and v are similar.

The vector filtering follows the loop validation and removes the remaining "outliers", by using the fact that the lens distortion is smooth and that the SIFT matchings are dense. A vector field can be obtained by superposing images I and v and connecting the correspondences. For each vector, a few neighboring vectors are used to obtain a median vector. The vector is eliminated if the modulus of difference between the vector and the median vector is significantly larger (3 times in the experiments) than that of the average difference between the neighboring vectors and the median vector.

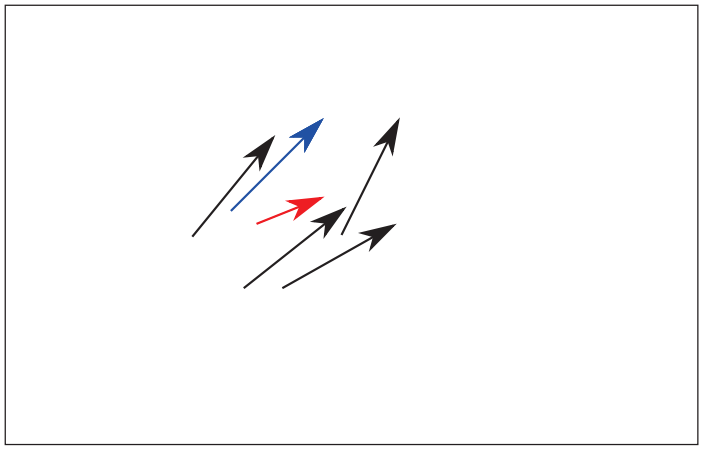
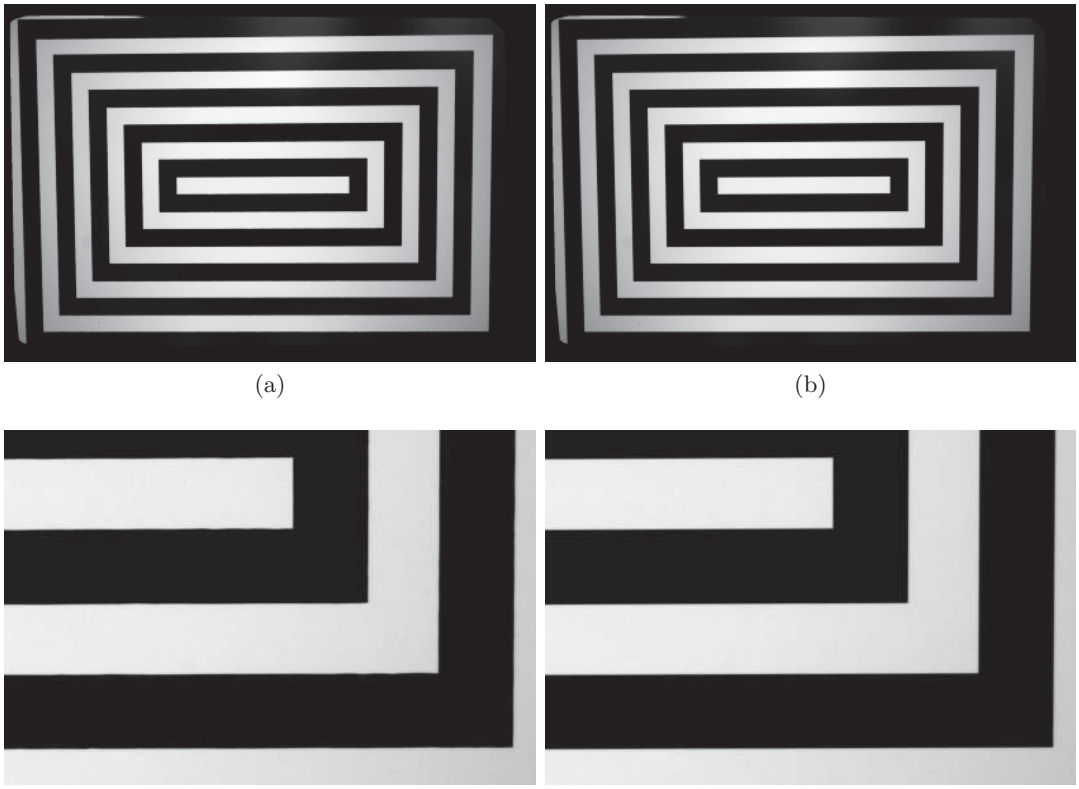


Figure 4.6: The vector filter. The red vector is shorter than its neighboring vector. The modulus of the difference between the red vector and the median vector (in blue) is much larger than the modulus of the average difference between the neighboring vectors and the median vector. So the matching pair corresponding to the red vector is eliminated.

4.3.6 Smoothing by Neighborhood Filter

The proposed distortion correction method is non-parametric and therefore entails no noise elimination. This can be clearly seen by reversing the distortion field on an image containing distorted lines (see Fig. 4.7a). This artifact can be corrected by smoothing the distortion field by a local filter. For each SIFT matching, 100 neighboring matchings around it are used to estimate the best local homography in the least-square sense. Then one point in each SIFT matching is adjusted according to its corresponding homography. This refinement is safe because SIFT matchings are dense and this does not modify the global property of the distortion field. This smoothing increases the precision of SIFT matchings and yields a final smooth distortion field. The image corrected by reversing this smoothed distortion field does not show any more zig-zag artifact (see Fig. 4.7b).



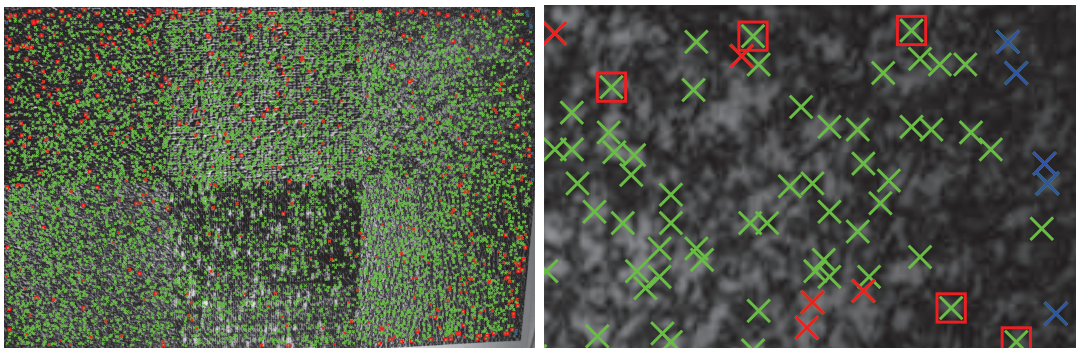
(a) (b)

(c) (d)

Figure 4.7: (a) the corrected image without smoothing. (b) the corrected image with smoothing. (c) one zoomed part of the image in (a). Notice the "zigzag effect" due to residual noise in SIFT matchings. (d) one zoomed part of the image in (b) after the application of the neighborhood filter.

4.3.7 Algorithm Summary

1. Take two slightly different photos of a textured planar pattern with constant camera settings;
2. apply SIFT between the original digital pattern and both photographs;
3. eliminate outliers by the loop validation step (see Fig. 4.8);
4. eliminate outliers by the vector filter (see Fig. 4.8);
5. increase the precision of SIFT matchings by a local neighbor filter (see Fig. 4.9);
6. interpolate the remaining matches to get a dense reverse distortion field (see Fig. 4.10);
7. by applying the reverse distortion field to all images produced by the real camera, the camera is converted into a virtual pinhole camera.

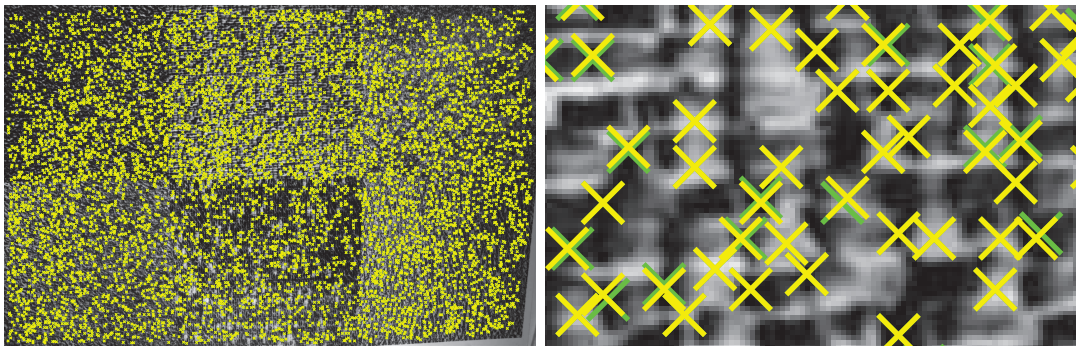


(a) (b)

Figure 4.8: Illustration of the loop validation and of the vector filter. The SIFT feature points are shown on image v in Fig. 4.4. (b) is a zoom of (a). The blue cross points (near the image border) are eliminated because they lie outside the Delaunay triangulation of the SIFT points of image u. The red cross points are eliminated by loop validation. The green points surrounded by red square are eliminated by the vector filter. The green cross points are the accepted points.

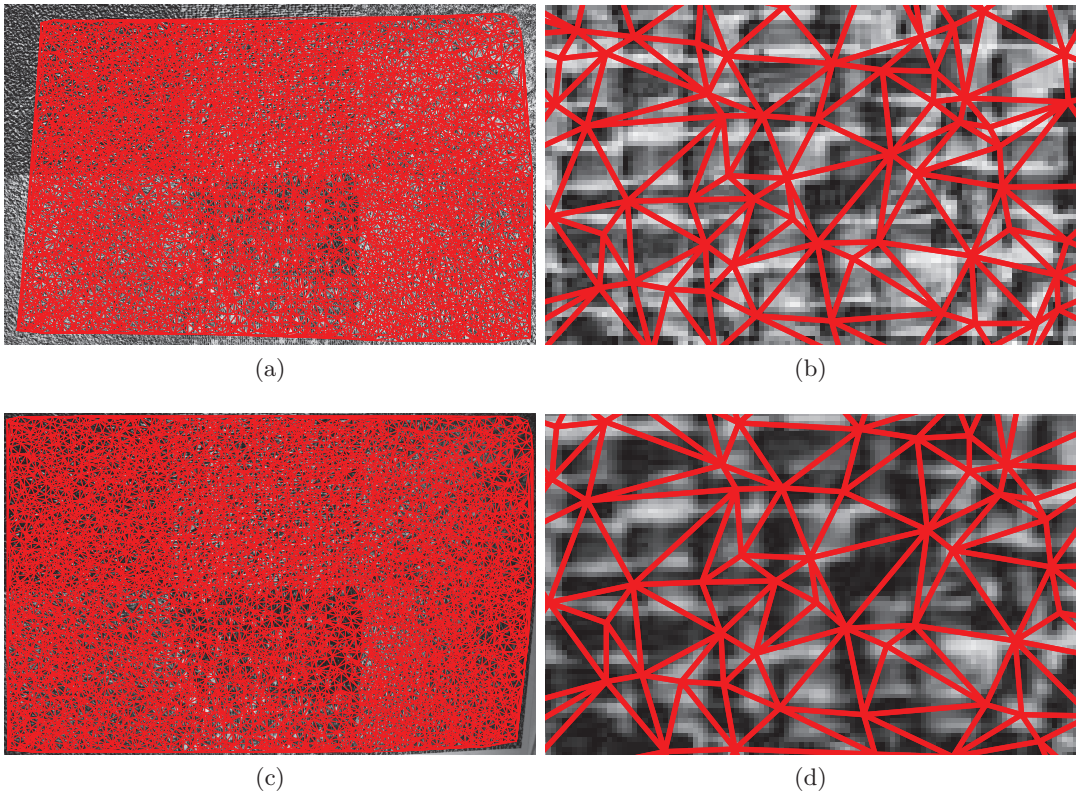
4.4 Experiments

The experiments were made with a Canon EOS 30D reflex camera and an EFS $18-55$mm lens. The minimal focal length (18mm) was chosen to produce a fairly large distortion. The distance between the camera and the object was about 30cm. To avoid any post-processing which could change the image properties, only raw images were used to perform the experiments. It is known that the distortion depends on the wavelength, which means that different colors undergo different distortions. This causes the phenomenon called "chromatic aberration". The green color dominates the Bayer cell (see Fig. 4.11). One possibility to avoid this would be to extract one green pixel from the four pixels of the Bayer cell to compose the distorted



(a) (b)

Figure 4.9: Illustration of the effect of local neighbor filter. The SIFT feature points are shown on image v in Fig. 4.4. (b) is a zoom of (a). The green cross points show the position before local neighbor filter. And the yellow cross points show the position after local neighbor filter.



(a) (b)

(c) (d)

Figure 4.10: Delaunay triangulation on the digital pattern, and on its photograph. (a) and (c) show the Delaunay triangulation on the digital pattern and the photograph. (b) is a zoom of (a). (d) is a zoom of (c). The Delaunay triangulation is performed on one of the two images. The triangulation is mapped to the other image by the SIFT matchings.

image. This is a 2-subsampling operation and it would give an aliased image. An alternative is to perform a simple averaging filter reducing the aliasing effect by simply summing up the four pixels of each 2×2 Bayer cell. The resulting image blur kernel after this subsampling is approximately the characteristic function of the pixel on the 2-subsampled image. The standard deviation of this blur in each coordinate direction can be computed as

$$\sigma_0^2 = \int_{-\frac{1}{2}}^{\frac{1}{2}} \int_{-\frac{1}{2}}^{\frac{1}{2}} (x^2 + y^2) dx dy = \frac{1}{6}. \tag{4.6}$$

It is shown in [132] that a well-sampled image requires a Gaussian blur with standard deviation 0.8. To rejoin this amount of blur a complementary Gaussian convolution with standard deviation $\sigma_g = \sqrt{0.8^2 - \sigma_0^2} \simeq 0.688$ is needed. The process to treat the RAW image is therefore:

1. Demosaick by summing up the four pixels of each 2×2 Bayer cell (R+G+G+B), obtaining a half-size image;
2. Convolve the image by a Gaussian blur with standard deviation 0.688.

To choose the pattern, the Brodatz textures [167] were tested by the SIFT method, simulating rotations and adding noise to test the number of robust matches. Six of them giving the most SIFT matchings were used to compose the pattern (see Fig. 4.1a for the digital pattern that was used). This pattern had actually more robust matches than a white noise pattern.

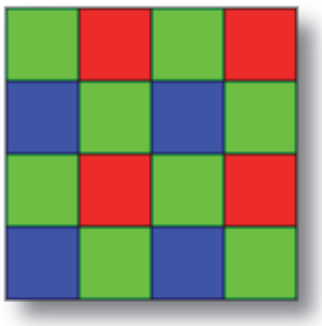


Figure 4.11: The Bayer arrangement of color filters on the pixel array of an image sensor: 50% green, 25% red and 25% blue

It is important to fix the camera parameters (focal length, focus, $\cdots$) in the experiments because the lens distortion changes with the camera configuration. A counterexample in Fig. 4.12 shows the effect of using an estimated distortion field for a give focus to correct an image taken under a different camera focus.

Fig. 4.13 shows a subsampling of the resulting distortion field, after the validation loop, and Fig. 4.14b shows the modulus of the interpolated distortion field on the discrete image domain. The distortion field is not circular symmetric, which is natural, the distortion being estimated up to an unknown homography.

To check the quality of the correction, we built a physical pattern with tightly stretched strings, that guarantees the straightness, see Fig. 4.14a. The distortion is visible near the

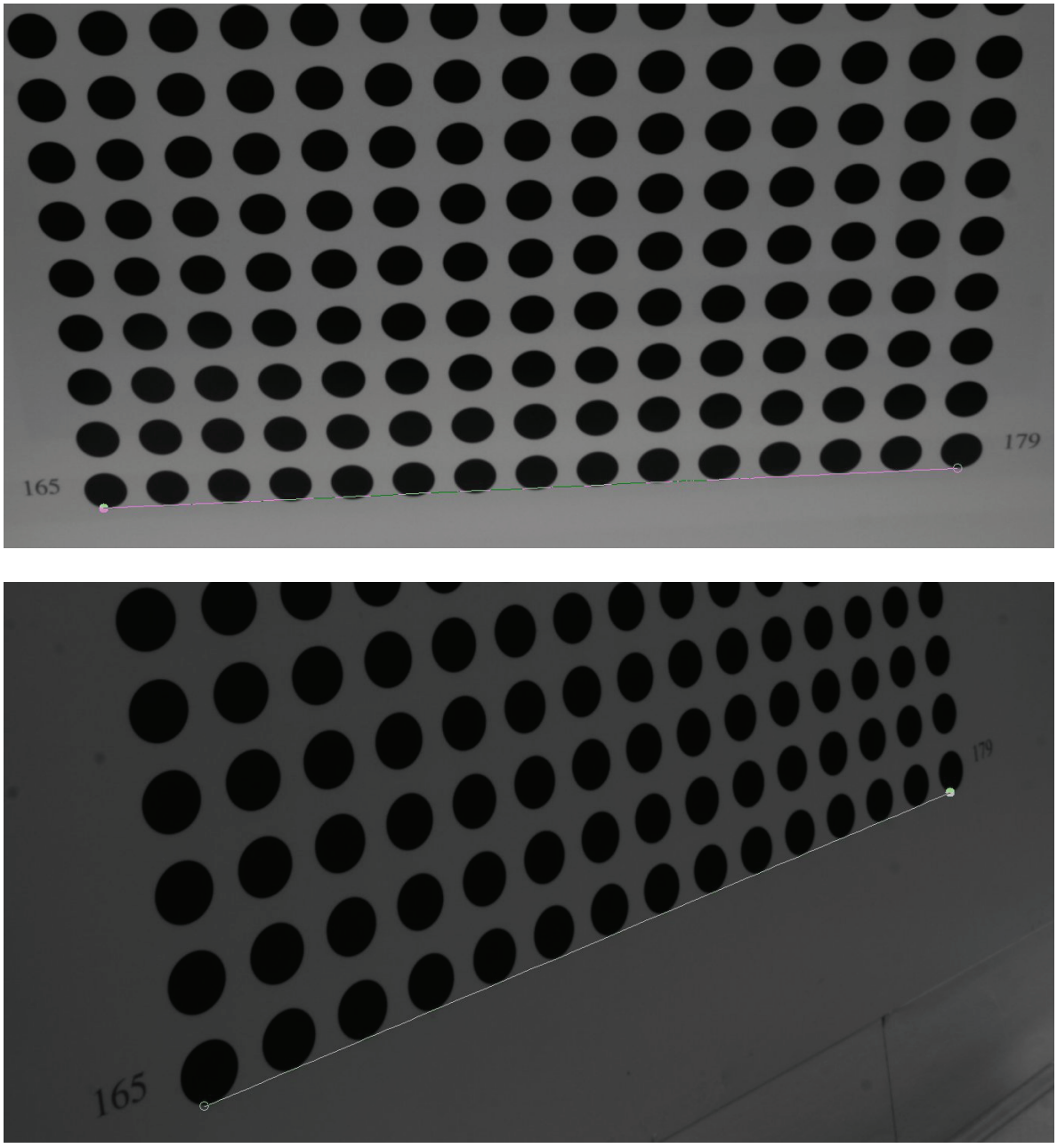


Figure 4.12: A visual examination of the distortion correction. A good correction is supposed to align the circles. The top image is not corrected because the used distortion field is estimated under a different camera focus. The bottom image shows a good example of distortion correction.

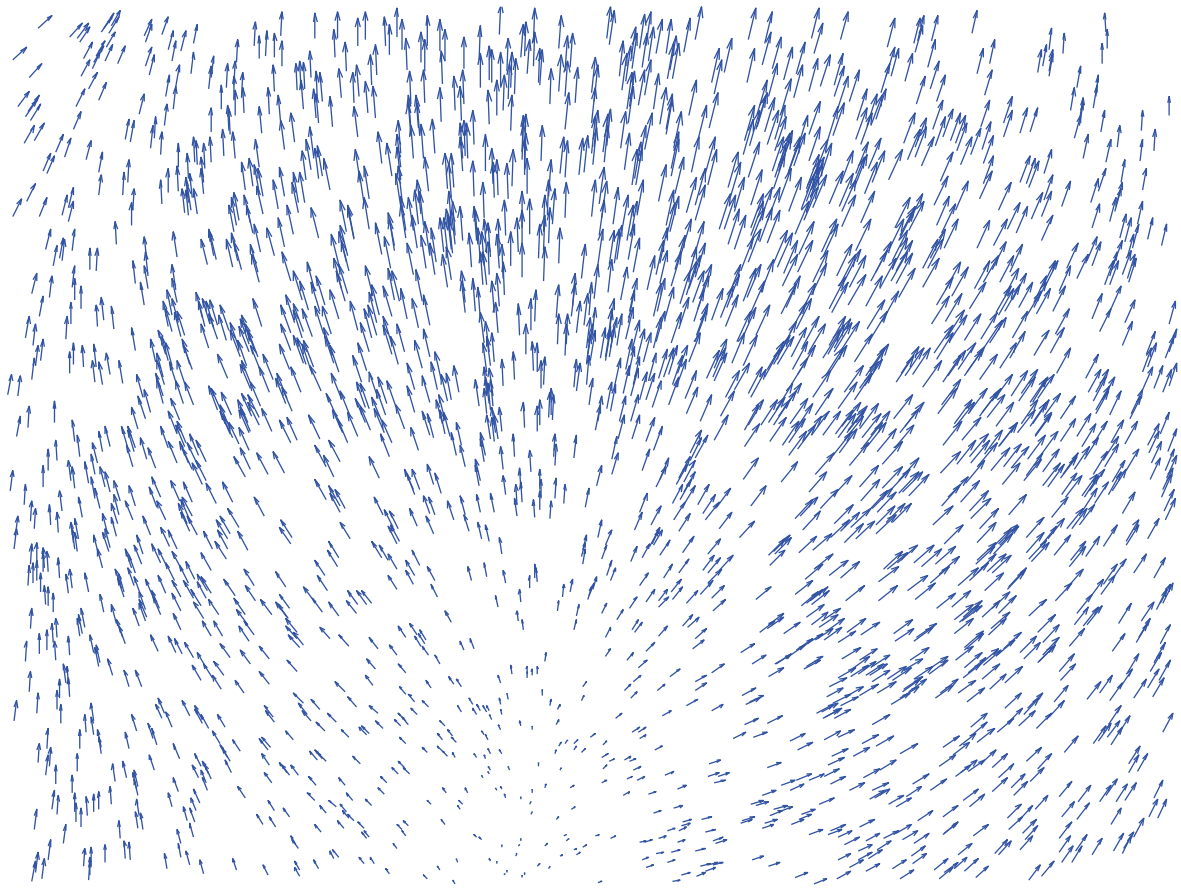
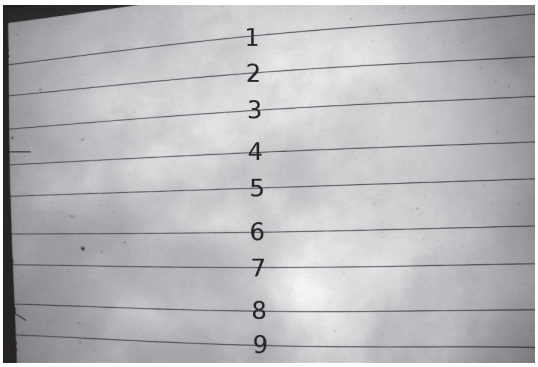


Figure 4.13: The (subsampled) distortion field directly defined on "inlier" correspondences after the loop validation, pointing from the points in the digital pattern to their correspondences in the distorted image.

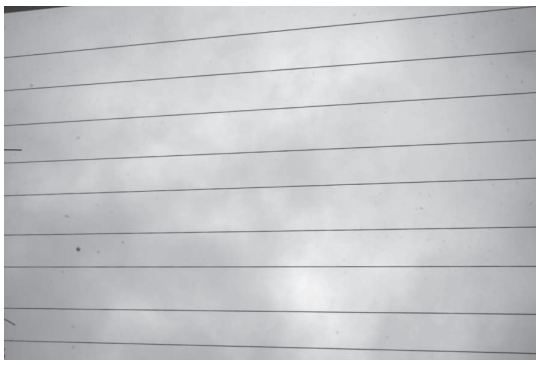


(a) distorted image of tightly stretched strings

(b) the distortion field estimated by the nonparametric method

(c) corrected image by the nonparametric method

(d) the distortion field estimated by the Lavest *et al.* method



(e) corrected image by Lavest *et al.* method

Figure 4.14: Distorted lines marked by numbers in (a) will be used to evaluate the precision after distortion correction. The norm of distortion field is coded into gray-level image here. Fig. 4.14b corresponds to Fig. 4.13.

RMSE (in pixels)		
	our method	Lavest method
line1	0.19/0.23	0.15/0.17
line2	0.08/0.11	0.08/0.10
line3	0.03/0.05	0.03/0.04
line4	0.06/0.05	0.10/0.09
line5	0.10/0.08	0.16/0.08
line6	0.11/0.10	0.14/0.13
line7	0.12/0.12	0.15/0.14
line8	0.09/0.09	0.12/0.12
line9	0.06/0.05	0.08/0.08

Table 4.1: Each distorted line marked by a number in Fig. 4.14a, is corrected either by the Lavest *et al.* method or by the proposed nonparametric method. The edge points are detected by Devernay's algorithm. The distortion error is computed as the root-mean-square distance (in pixels) from the edge points to their regression line. In each cell, there are two values because there are two sides for each line.

border of the image. Fig. 4.14c shows the image corrected by the proposed nonparametric method. Figs 4.14d and 4.14e show the distortion field and the corrected image provided by Lavest *et al.* algorithm. The lines numbered in Fig. 4.14a were used to evaluate the distortion error and to compare it with the error left by the Lavest *et al.* algorithm [95]. This algorithm is a global camera calibration method, which estimates camera external and internal parameters simultaneously based on several pattern photographs (Fig. 4.15), by minimizing the back-projection error. In addition, the Lavest *et al.* method does not require a complete flat pattern because it estimates also the 3D position of feature points on the pattern.

On each corrected line, subpixel precision edge points were obtained by Devernay's algorithm [52]. Then, their regression line was computed and the RMS (root-mean-square) distance from each edge point to the line was used as a purely geometric error measure. Table 4.1 shows the results. The proposed nonparametric method shows a comparable result with the Lavest *et al.* method. But this difference is not just quantitative. The Lavest *et al.* result is somewhat final. Indeed, it already includes a correction of the non-flatness of the pattern while the nonparametric method does not.

Figs 4.16 and 4.17 plot the straightness error along the lines corrected by the proposed method and by the Lavest *et al.* method respectively (that is, the distance between edge points to the regression line). One can observe small oscillations which are easily explained by noise and aliasing, but also a global tendency which is caused by the non-flatness of the pattern. A parallel deterministic tendency observed on the other lines confirms this explanation. Fig. 4.18 shows that a flatness error of 100 μm (the thickness of a normal A4 paper sheet) can produce the observed tendency. This non-flatness effect is stronger near the border of the image because the angle-of-view is larger. Simple physical measurements by the

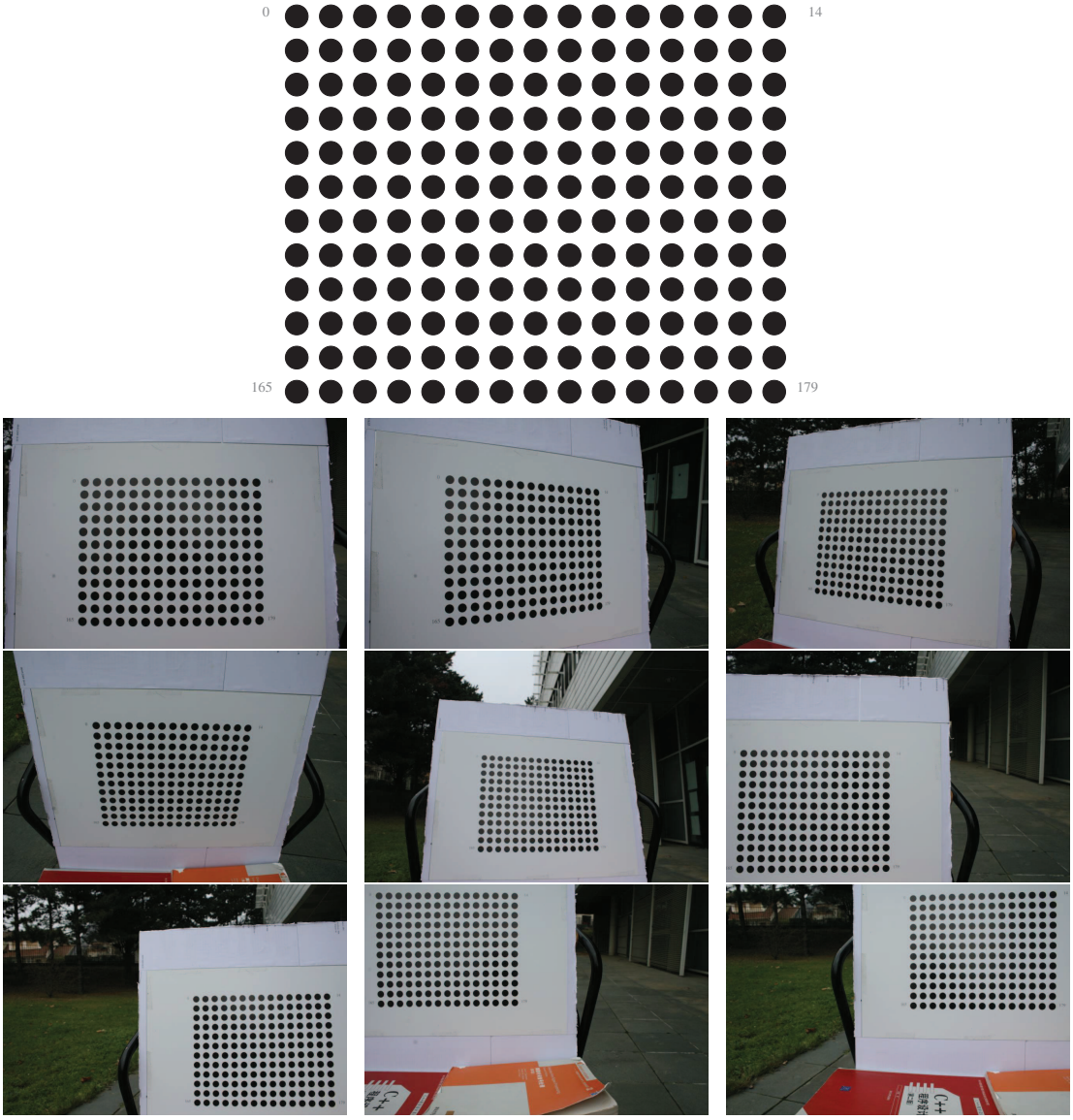
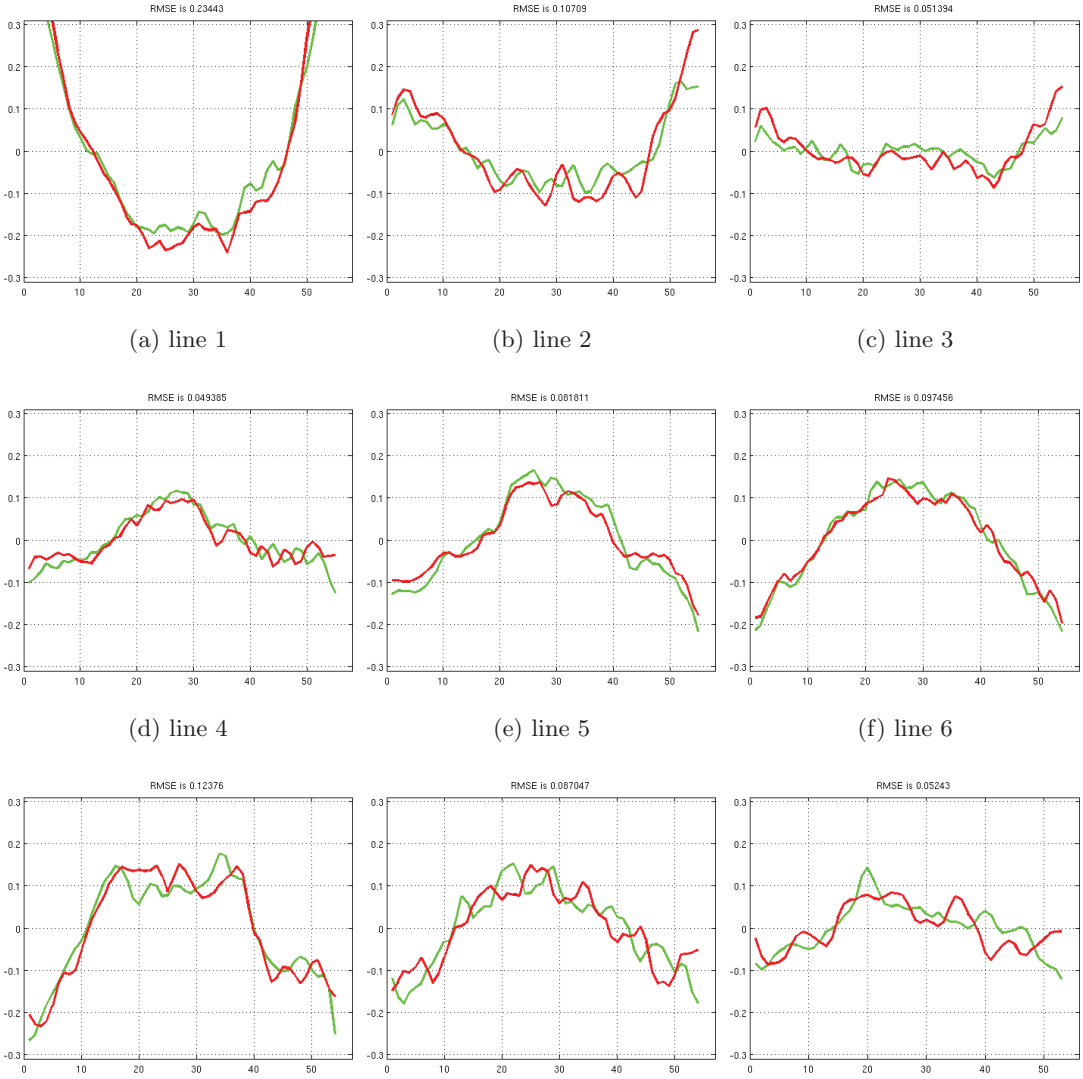


Figure 4.15: The top image is the pattern used in the Lavest *et al.* method. The other images are the photographs of the pattern taken from different viewpoints.

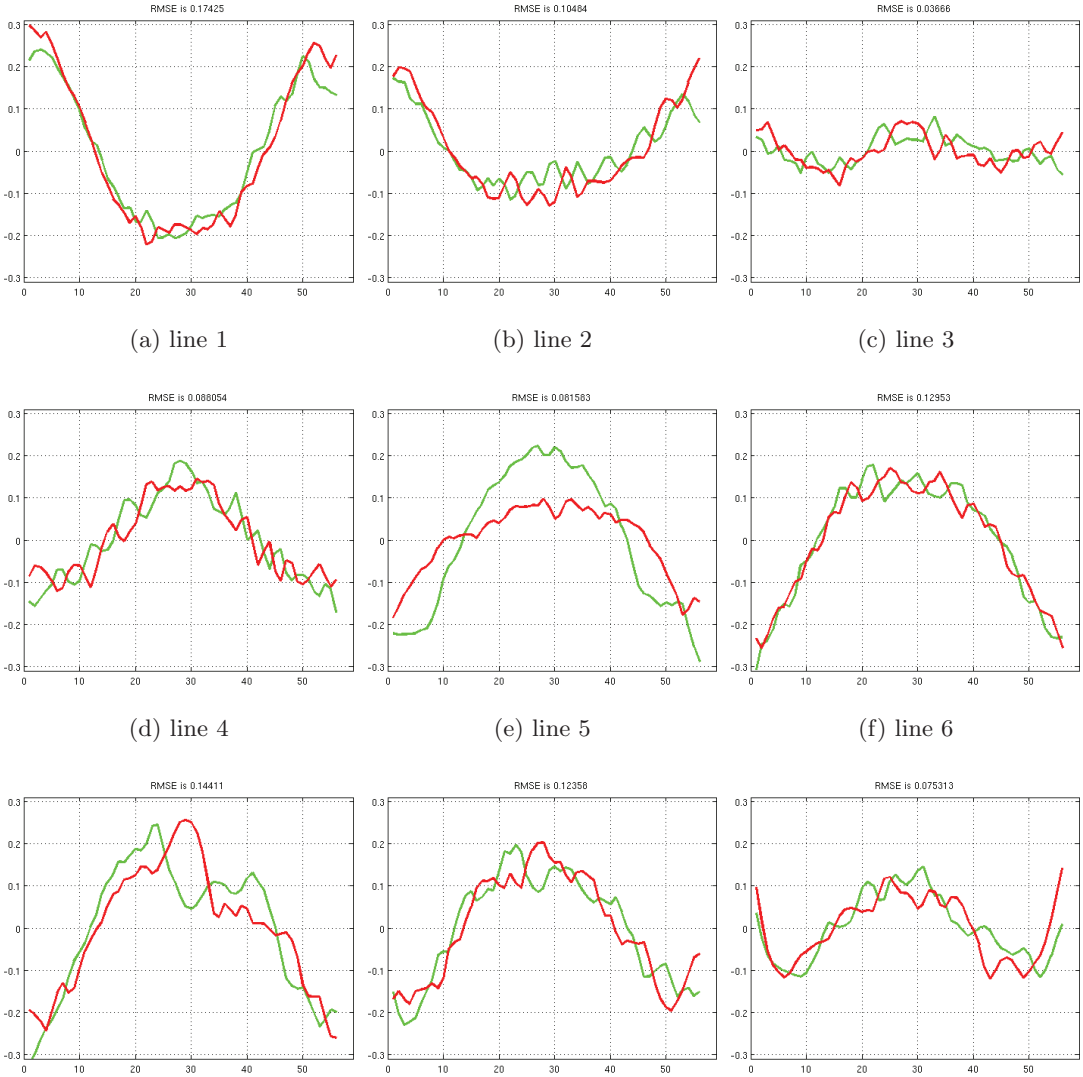


(a) line 1 (b) line 2 (c) line 3

(d) line 4 (e) line 5 (f) line 6

(g) line 7 (h) line 8 (i) line 9

Figure 4.16: The distance in pixels from the edge points to their regression line on the numbered lines in Fig. 4.14a, after correction by the proposed method. Note that each figure contains two curves because there are two lines for one string. The x-axis is the index of edge points. The range of y-axis is from -0.3 pixels to 0.3 pixels.



(a) line 1 (b) line 2 (c) line 3

(d) line 4 (e) line 5 (f) line 6

(g) line 7 (h) line 8 (i) line 9

Figure 4.17: The distance in pixels from the edge points to their regression line on the numbered lines in Fig. 4.14a, after correction by the Lavest *et al.* method. Note that each figure contains two curves because there are two lines for one string. The x-axis is the index of edge points. The range of y-axis is from -0.3 pixels to 0.3 pixels.

classic *ruler method* [1] confirmed that the pattern showed a non-flatness of this amount. The Lavest *et al.* method also leaves a global uncorrected tendency. The biggest straightness error is almost 10 times as big as the minimized re-projection error (about 0.02 pixels). This cannot be explained by the non-flatness of pattern because the Lavest *et al.* method also estimates the 3D position of feature points on the pattern. The only reasonable explanation is that errors in the external and internal camera parameter are being compensated by opposite errors in the distortion model. Thus, an inaccurate distortion model can pass undetected with a very small re-projection error.

The Lavest *et al.* method gives a quantitative evaluation of the non-flatness of the pattern, under two assumptions: one is that the circular feature pattern used in Lavest *et al.* has the non-flatness on the same order of the textured pattern; the other is that the 3D points estimated by Lavest *et al.* is more or less correct. Fig. 4.19a shows the surface of circular feature pattern interpolated by the 3D positions of the circle centers on the mirror pattern estimated by Lavest *et al.* method. By estimating a linear regression plane from these 3D points, the maximal distance $d_{\mathtt{max}}$ from the points to the plane can be computed. It seems that the point farthest from the regression plane is also close to the border of pattern, which is more responsible for the global tendency observed in the corrected lines. $2d_{\mathtt{max}} \approx 0.08$ mm is close to the thickness of a A4 paper sheet (0.1 mm) obtained by the *ruler method* (see footnote 1).

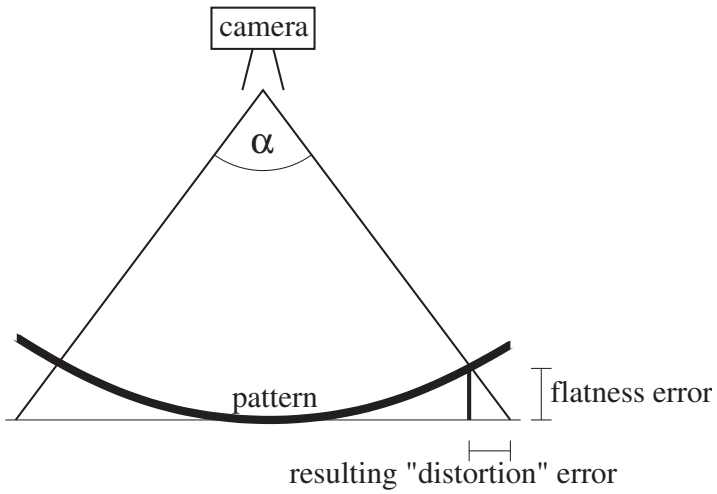


Figure 4.18: A flatness error in the pattern can be mistaken for a "distortion". In the experiments, with $\alpha \approx 65\,^\circ$ and a flatness error about 100 μm, the produced "distortion" error is about 64 μm ($\textit{flatness error} \times \tan\frac{\alpha}{2}$). Our pattern has the size 406×271 mm and produces a 1761×1174 image, then one pixel corresponds to 230 μm. Thus the observed error would be approximately 0.3 pixel.

[1] The ruler method consists in measuring the non-flatness of the pattern by a ruler and a A4 paper sheet. One A4 paper sheet is put on the pattern with a rigid ruler pushed on it. If the non-flatness of the pattern is less than the thickness of one A4 paper sheet, the paper cannot be dragged from the bottom of the ruler. In our test, the A4 paper sheet can be easily dragged out in some areas of the pattern. This means that the mirror is not flat. The thickness of 500 pieces of paper is about 5.65 cm. So one A4 paper sheet is about 100 μm, which is the magnitude of non-flatness of the pattern.

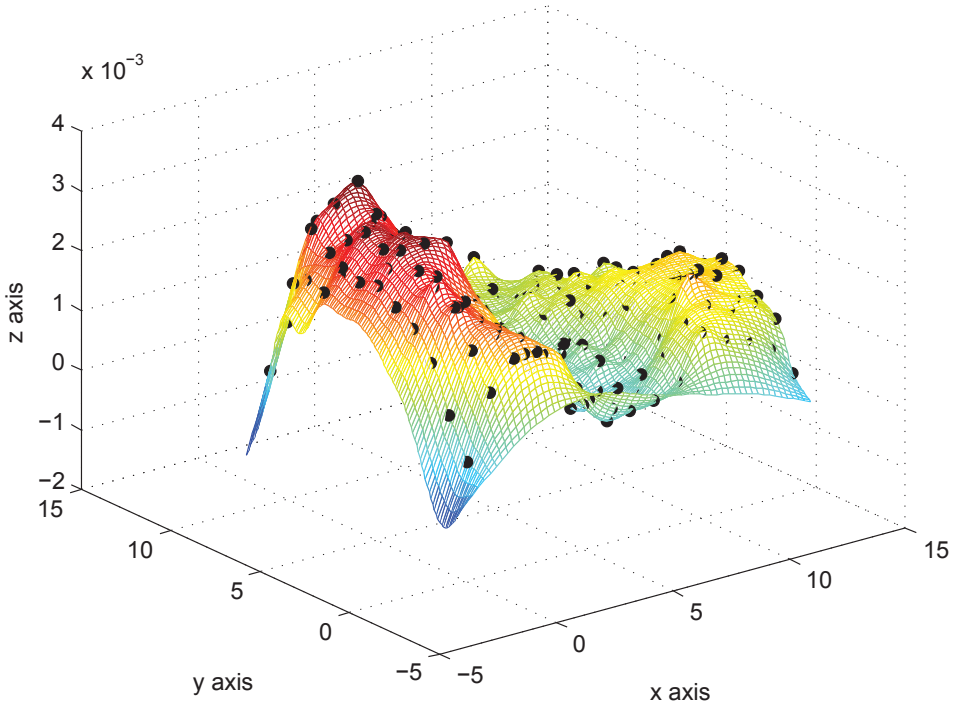
4.5 Discussion

The above experimental setting suggests two ways to eventually reach a still higher precision. The first way would simply be to use very flat patterns. But this raises the problem of making smart patterns. An alternative strategy was suggested in [4], using accurate straight objects, like the tightly stretched strings we already used here. Yet, it seems advisable to try to keep the strength and beauty of the Lavest *et al.* method, which is to estimate and correct the pattern's shape by the calibration process itself. An iterative method could be envisaged where, first, the distortion is corrected by the proposed nonparametric method and, second, the physical shape of the pattern is computed by the Lavest *et al.* method *with no distortion model*. Using this correction the distortion would be recomputed, and so on. This is, however, a complex process, which will require a heavier procedure and a mathematical analysis. In addition, the Lavest *et al.* method can only estimate the shape of some specific pattern (like Fig. 4.15), which does not necessarily have the same shape as the textured pattern we used in the nonparametric method. [2]

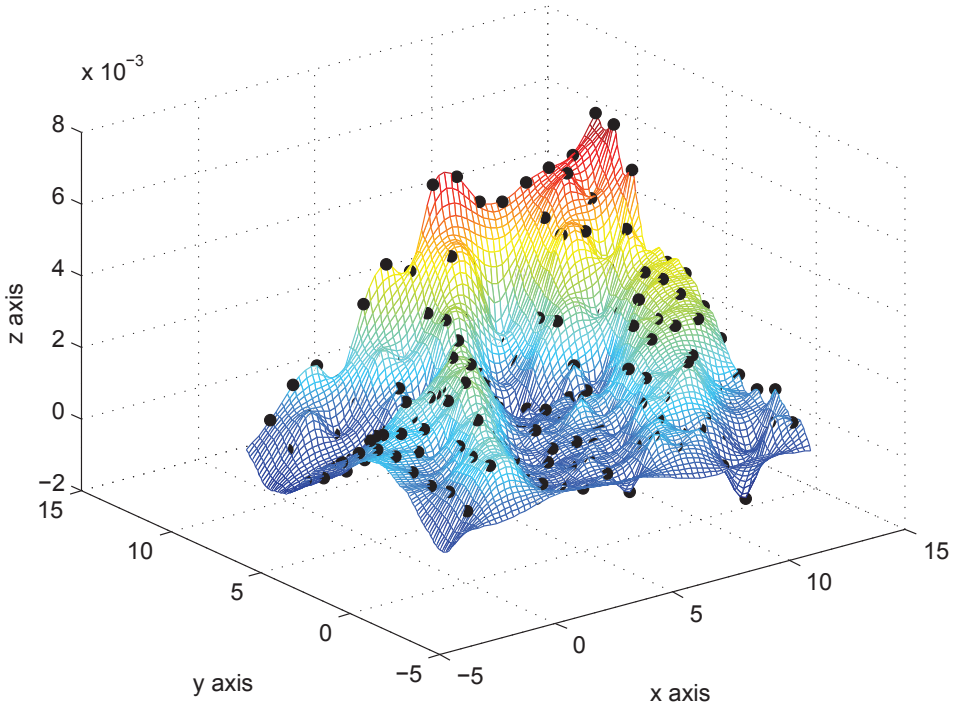
Using tightly stretched strings to correct distortion is another track we will follow later. Here we still concentrate on the non-flatness of the pattern. Bearing in mind the idea that the mirror is not flat, we used two thick industrial aluminium plates and pasted on them the disks pattern and the textured pattern respectively. The same procedure was used to estimate the shape of pattern by the Lavest *et al.* method (see Fig. 4.19b). Surprisingly the industrial aluminium pattern was not more flat than the mirror, even though it was far more solid and thicker than the mirror. The maximal distance from the point to the regression plane is $d_{\max} \approx 0.05$ mm. Then $2d_{\max}$ is again the thickness of a A4 paper sheet. So with the same experimental setting as before (camera focal length 18 mm; camera-object distance about 30 cm), the flatness error incurs the same magnitude of global tendency to the lines corrected by nonparametric method (see Fig. 4.20a for the distorted image, Fig. 4.20b for the corrected image, Fig. 4.21 for the straightness error along the corrected lines in Fig. 4.20b and Table 4.2 for the RME distance for each corrected line) with a textured pattern pasted on an aluminium plate. It seems difficult to obtain enough flatness for a sizeable material pattern of 30 centimeters.

For indoor 3D reconstruction, the angle-of-view of camera is smaller. This permits to reduce to some extent the effect of pattern non-flatness. By using the maximal focal length 55mm, a distance about 100cm was needed to capture the whole textured pattern as before. So the angle-of-view became about three times smaller. According to Fig. 4.18, the error incurred by the non-flatness should be also three times smaller. The improvement in correction accuracy can indeed be seen in Fig. 4.22 and Table 4.2 (see the distorted image in Fig. 4.20c and the corrected image in Fig. 4.20d).

[2] Both the circular feature pattern and the textured pattern are printed and pasted on the flat mirror. But nothing can ensure that the two mirrors have exactly the same shape and the imperfection introduced in the printing and gluing is the same for two patterns.



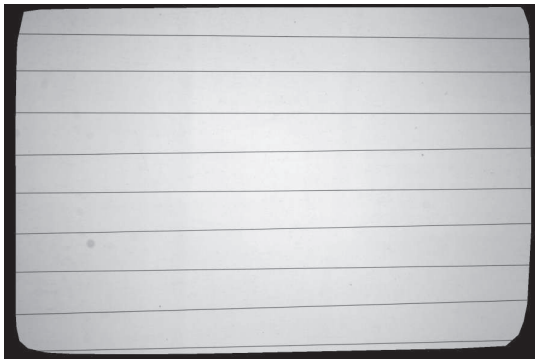
(a) The surface of the mirror pattern



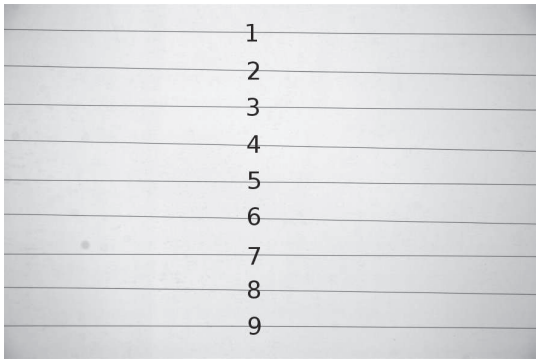
(b) The surface of the aluminium pattern

Figure 4.19: The surface of the circular feature pattern. (a) Mirror pattern. (b) Aluminium pattern. The black points are the 3D position of circle centers on the pattern estimated by Lavest *et al.* method. The surface of pattern is interpolated from these points. One unity is equivalent to $\frac{235}{14} \approx 16.8$ mm.

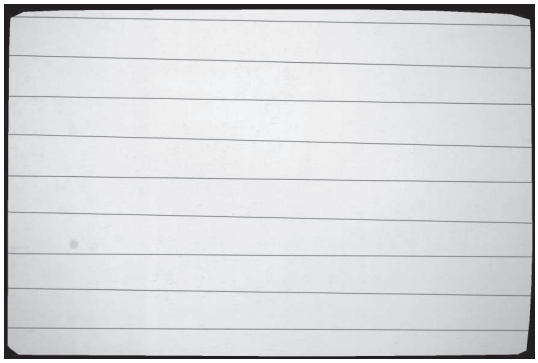
(a) The distorted image with 18mm focal length and about 30cm camera-pattern distance



(b) Left image corrected by nonparametric method

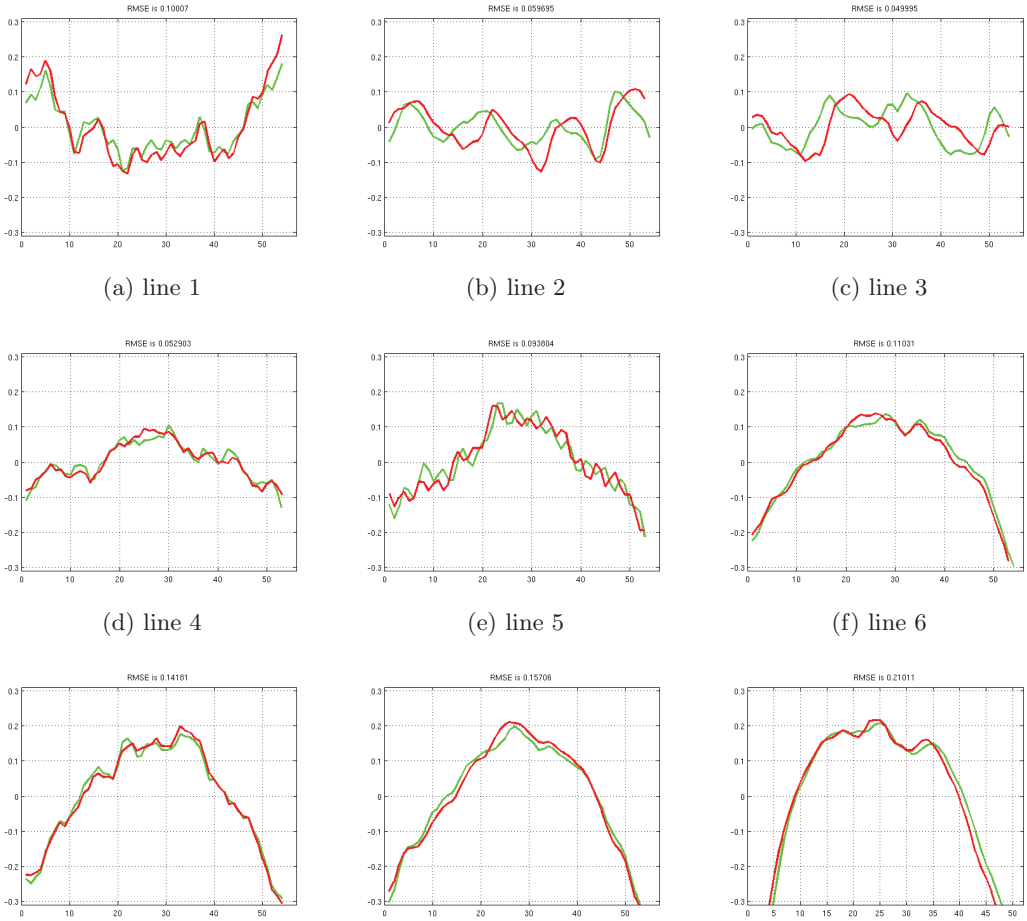


(c) The distorted image with 55mm focal length and about 100cm camera-object distance



(d) Left image corrected by nonparametric method

Figure 4.20: The correction example by the textured pattern pasted on the aluminium plate. (a) and (b) are the example under the experimental setting that camera has focal length 18mm and the camera-pattern distance about 30cm. (c) and (d) are the example under the experimental setting that camera has focal length 55mm and the camera-pattern distance about 100cm.

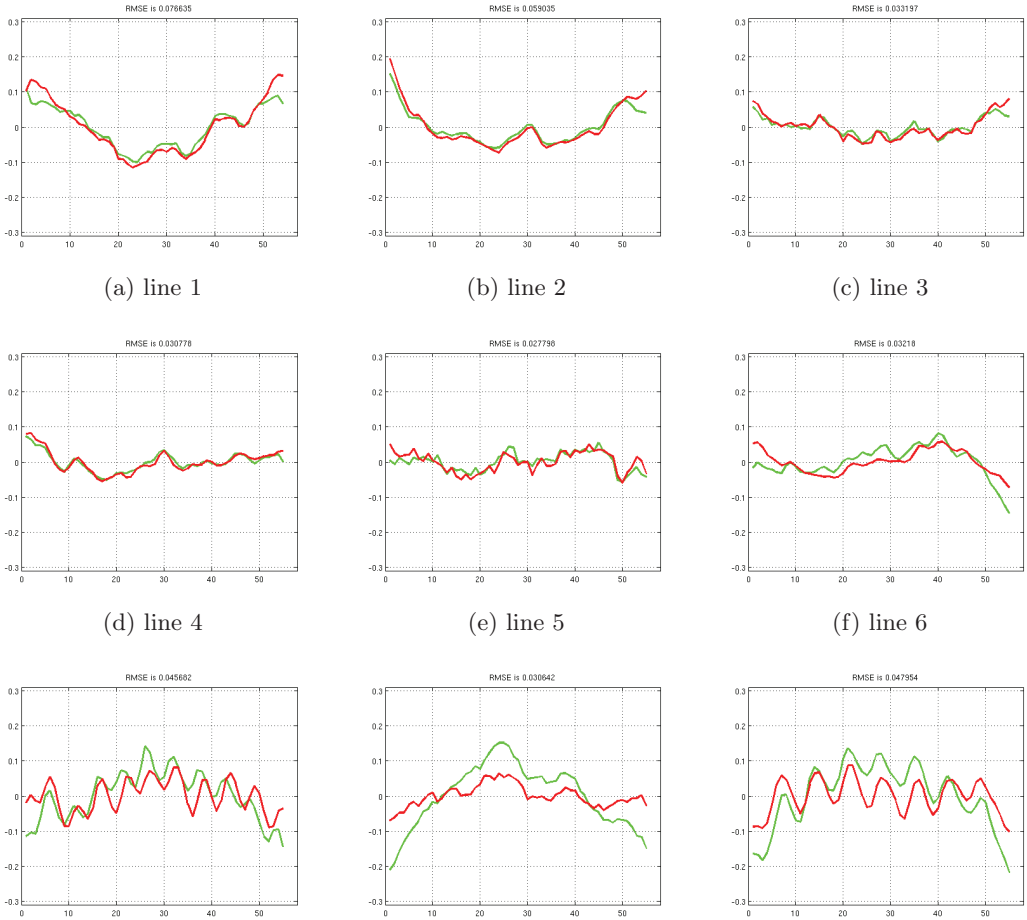


(a) line 1 (b) line 2 (c) line 3

(d) line 4 (e) line 5 (f) line 6

(g) line 7 (h) line 8 (i) line 9

Figure 4.21: The distance in pixels from the edge points to their regression line on the numbered lines in Fig. 4.20a, after correction by the proposed method. *The textured pattern is pasted on a aluminium plate.* Note that each figure contains two curves because there are two lines for one string. The x-axis is the index of edge points. The range of y-axis goes from -0.3 pixels to 0.3 pixels.



(a) line 1 (b) line 2 (c) line 3

(d) line 4 (e) line 5 (f) line 6

(g) line 7 (h) line 8 (i) line 9

Figure 4.22: The distance in pixels from the edge points to their regression line on the numbered lines in Fig. 4.20c, after correction by the proposed method. *The textured pattern is pasted on a aluminium plate.* Note that each figure contains two curves because there are two lines for one string. The x-axis is the index of edge points. The range of y-axis is from -0.3 pixels to 0.3 pixels.

RMSE (in pixels)		
	our method (near)	our method (far)
line1	0.07/0.10	0.06/0.08
line2	0.05/0.06	0.05/0.06
line3	0.05/0.05	0.03/0.03
line4	0.05/0.05	0.03/0.03
line5	0.09/0.09	0.03/0.03
line6	0.11/0.11	0.04/0.03
line7	0.14/0.14	0.07/0.05
line8	0.15/0.16	0.09/0.03
line9	0.20/0.21	0.09/0.05

Table 4.2: Near view and far view for nonparametric textured pattern based distortion correction. Left column: camera focal length 18mm and camera-pattern distance about 30cm, with distorted image in Fig. 4.20a and corrected image in Fig. 4.20b. Right column: camera focal length 55mm and camera-pattern distance about 100cm, with distorted image in Fig. 4.20c and corrected image in Fig. 4.20d. Each distorted line, marked by a number, is corrected by the proposed nonparametric method. The edge points are detected by Devernay's algorithm. The distortion error is computed as the root-mean-square distance (in pixels) from the edge points to their regression line.

Chapter 5

Self-Consistency and Universality of Camera Lens Distortion Models

Contents

Abstract *This chapter introduces the concepts of "self-consistency" and "universality" to evaluate the validity and precision of camera lens distortion models. Self-consistency is evaluated by the residual error when the distortion generated with a certain model is corrected by the best parameters for the same model (used in reverse way). Analogously, universality is measured by the residual error when a model is used to correct distortions generated by a family of other models. Five classic camera lens distortion models are reviewed and compared for their degree of self-consistency and universality. The study shows that radial symmetric models can be self-consistent, but cannot be used for non radial-symmetric distortion. Among the evaluated models, the polynomial and the rational models are the only ones to be universal up to precisions of 1/100 pixel. However, the polynomial model, being linear, is much simpler and faster to estimate. Unusually high polynomial orders are required to reach a* 1/100 *pixel precision. But our experiments show that such polynomials are easily computed, producing a precise lens distortion correction without over-fitting. Our conclusions are validated by three independent experimental setups: The models are compared first in synthetic experiments by their approximation power; second by fitting a real camera distortion estimated by a non parametric algorithm; and finally by the absolute correction measurement provided by photographs of tightly stretched strings, warranting a high straightness. Finally, our experiments show that in the polynomial model the residual errors stabilize for orders between 6 to 12, confirming that no over-fitting occurred. High order polynomials are unavoidable to obtain high precisions, and deliver accuracies hundred to thousand times higher than those obtained with classic models.*

5.1 Introduction

The pinhole camera model is widely used in computer vision applications because of its simplicity and its linearity in terms of projective geometry [89]. But real cameras deviate from the ideal pinhole model, mainly because of lens distortion [25]. Thus an accurate camera lens distortion correction is the first step towards high precision 3D metric reconstruction from photographs. With the steady progress in lens quality and computing power, high-precision 3D reconstructions become feasible, demanding in turn higher lens distortion precisions than those provided by classic methods. The object of this paper is to investigate the validity of distortion models at the light of precision requirements that over the pas decade has increased by two to three orders of magnitude. This increased accuracy requires a new methodology for evaluating distortion models. In a nutshell, our conclusion is that a polynomial model of higher degree than usual, ranging from 8 to 15, is necessary for reaching a pixel precision ranging from 1/100 to 1/1000. The polynomial model permits to approximate at this resolution any other model, and the inverse of any other model, including itself. When these properties are reached, the model is called *universal* and *self-consistent*. Among the other four models which will be compared (radial, division, FOV, and rational), only the rational model has the exigible self-consistency and universality, but to a far higher computational cost. (A complex incremental minimization algorithm is needed to solve the rational model, without ensuring the global minima.)

Since the first numerical lens distortion model by Brown [25], many methods [119, 113] have been proposed to correct lens distortion (see [37] for a review of the development of camera calibration methods in early years). The final aim is to obtain an ideal pinhole (or pinhole equivalent) camera by removing lens distortion, so that the classic multi-view geometry

techniques can be applied directly.

With the exception of a few non-parametric methods [61, 162, 80], an appropriate distortion model is indispensable to establish a correct camera model. The main distortion models are the radial model [25], the division model [68], the FOV model [53], the bicubic model [101], the rational model [39, 86]. This diversity is only marginally linked to the kind of camera. Thus, a synthetic quantitative and qualitative comparison is required. Do these models reflect camera lens distortion in its physical aspect? It could be argued that a correct model should originate from physical measurements on systems of lenses. Surprisingly enough, there is little physical background for the distortion models in the literature. It is true that in [177] lens distortion is decomposed into three effects: radial distortion, decentering distortion and thin prism distortion. But, still, it is only marginally based on a physical background. In fact, the final distortion includes effects caused by a complex lens system, by the camera geometry, and by the (not perfectly planar) shape of the captor. One is therefore led to figure out a flexible model with enough parameters to approximate any plausible distortion. In absence of a physical model, the model classification approach adopted here will be to look for models which actually cope with any other proposed distortion model, at a given precision. Such models will be called *universal*. The second question is the relationship between the distortion and the correction model, which should be inverse of each other. Indeed, the *correction model* and the *distortion model* must be different. A correction model is used to correct distorted images, while a distortion model is used to model the distortion of ideal images. In the literature, however, it seems that the roles of *distorted point* and *undistorted point* are interchangeable, which again confirms the lack of physical meaning for these models. For example, direct distortion models are used in global camera calibration [172, 191, 95, 177]. Yet, in most plumb-line methods [25, 53, 4, 151, 145, 38] or some pattern-free methods [161, 190, 68, 108, 168, 39, 144, 32, 103], the very same correction models are used without any fuss to approximate the inverse distortion.

Assume we simulate a camera lens distortion with a certain model and a certain set of parameters. Except for some trivial cases, the distortion will not be corrected by using the same model with other parameters, because the model itself is usually not *invertible*. We propose to measure the error incurring when inverting a distortion with the same model as for the distortion. This error when the best correcting parameters are applied will be a measurement of the model *self-consistency*. In other words, self-consistency relates to how well a model is able to correct distortion generated by a model of its own family. Of course the best models should be universal, therefore able to correct distortions generated by other models. We therefore propose to measure a model *universality* as the residual error when this model is used to correct distortion generated by a whole set of different models. A *universal model* is a model for which this error is very small no matter what other (reasonable) distortion model has been applied. A universal model must of course be also self-consistent. Our goal is to identify the least complex universal and self-consistent models.

The various distortion models will be carefully compared on realistic synthetic distortion data permitting to quantify the ideal attainable precision. Then, the same models will be compared on their capacity to fit a real camera lens distortion (estimated by a non-parametric algorithm [80]). Finally, the lens distortion correction accuracy by each model will be evaluated by using the plumb-line approach, with photographs of tightly stretched strings, warranting a high straightness, and giving an absolute measure of the correction quality. In short, there

will be four different numerical validations of our conclusions.

This chapter is organized as follows. Section 5.2 reviews five classic distortion models. Their self-consistency and universality are evaluated in Section 5.3 by synthetic experiments. Section 5.4 and 5.5 describe the experiments done with real camera lenses. Section 5.6 is a conclusion.

5.2 Distortion and Correction Models

We start by reviewing the most current models, namely the radial model [25], the division model [68], the FOV model [53], the polynomial model [101], and the rational function model [39, 86]. All of these models are expressed as distortion models, but are actually also used as correction models.

Denote by (x_u, y_u) an undistorted point, (x_d, y_d) the distorted point, (x_c, y_c) the distortion center, $(\bar{x}_u, \bar{y}_u)$ the radial undistorted point and $(\bar{x}_d, \bar{y}_d)$ the radial distorted point where $\bar{x}_u = x_u - x_c$, $\bar{y}_u = y_u - y_c$, $\bar{x}_d = x_d - x_c$ and $\bar{y}_d = y_d - y_c$. The distorted radius is $r_d = \sqrt{\bar{x}_d^2 + \bar{y}_d^2}$ and the undistorted radius $r_u = \sqrt{\bar{x}_u^2 + \bar{y}_u^2}$.

The radial model displaces a point along its radial direction originating at the distortion center. The distorted new radius r_d is a function of the original radius r_u,

$$r_d = r_u f(r_u) = r_u(k_0 + k_1 r_u + k_2 r_u^2 + \cdots). \tag{5.1}$$

The parameter k_0 representing a scaling does not introduce distortion. The scaled image is distorted by $k_1, k_2, \cdots$. If $k_1, k_2, \cdots$ are all positive, we have a *pincushion distortion*; if $k_1, k_2, \cdots$ are all negative, a *barrel distortion*. *Mustache distortion* occurs if the signs of $k_1, k_2, \cdots$ are not the same (see Fig. 5.1).

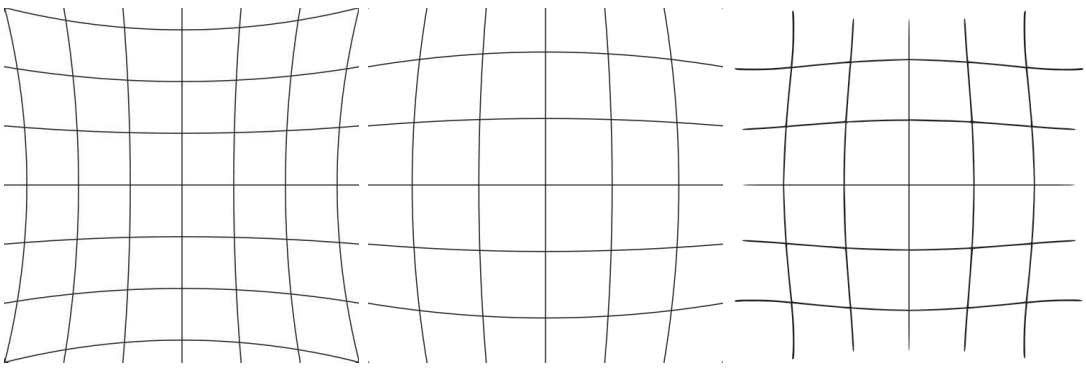


Figure 5.1: Left: pincushion distortion. Middle: barrel distortion. Right: mustache distortion.

The *division model* is nothing but the scalar inverse of the radial model,

$$r_d = r_u f(r_u) = \frac{r_u}{k_0 + k_1 r_u + k_2 r_u^2 + \cdots}. \tag{5.2}$$

In these models, the higher order coefficients are needed to model extreme distortion in fish-eye lenses or other wide angle lens systems. A more sparse representation is obtained by

parameterizing the distortion by the field of view (FOV),

$$r_d = r_u f(r_u) = r_u \frac{\tan(r_u \omega)}{2 r_u \tan(\frac{\omega}{2})}. \tag{5.3}$$

where the only parameter of the classic FOV model is the field of view order 1 coefficient. The terms of radial model can be added to make FOV model more complete.

In the *polynomial model* the distortion is modeled as a polynomial in $\bar{x}_u$ and $\bar{y}_u$. For example, the third order (bicubic) polynomial model is

$$\begin{aligned} \bar{x}_d &= a_1 \bar{x}_u^3 + a_2 \bar{x}_u^2 \bar{y}_u + a_3 \bar{x}_u \bar{y}_u^2 + a_4 \bar{y}_u^3 + a_5 \bar{x}_u^2 \\ &\quad + a_6 \bar{x}_u \bar{y}_u + a_7 \bar{y}_u^2 + a_8 \bar{x}_u + a_9 \bar{y}_u + a_{10} \\ \bar{y}_d &= b_1 \bar{x}_u^3 + b_2 \bar{x}_u^2 \bar{y}_u + b_3 \bar{x}_u \bar{y}_u^2 + b_4 \bar{y}_u^3 + b_5 \bar{x}_u^2 \\ &\quad + b_6 \bar{x}_u \bar{y}_u + b_7 \bar{y}_u^2 + b_8 \bar{x}_u + b_9 \bar{y}_u + b_{10} \end{aligned} \tag{5.4}$$

The rational function model is a quotient of two polynomials. A second order rational function model can be written as

$$\begin{aligned} \bar{x}_d &= \frac{a_1 \bar{x}_u^2 + a_2 \bar{x}_u \bar{y}_u + \cdots + a_5 \bar{y}_u + a_6}{c_1 \bar{x}_u^2 + c_2 \bar{x}_u \bar{y}_u + \cdots + c_5 \bar{y}_u + c_6} \\ \bar{y}_d &= \frac{b_1 \bar{x}_u^2 + b_2 \bar{x}_u \bar{y}_u + \cdots + b_5 \bar{y}_u + b_6}{c_1 \bar{x}_u^2 + c_2 \bar{x}_u \bar{y}_u + \cdots + c_5 \bar{y}_u + c_6} \end{aligned} \tag{5.5}$$

All of the above models, including the radial model, the division model and the FOV model which are radial symmetric, have the following decomposition in the x and y direction:

$$\begin{aligned} \bar{x}_d &= f_x(\bar{x}_u, \bar{y}_u) \\ \bar{y}_d &= f_y(\bar{x}_u, \bar{y}_u). \end{aligned} \tag{5.6}$$

The form of f_x and f_y depends on the specific model.

5.3 Self-Consistency and Universality

In the literature it is not always clear whether the above models are correction models or distortion models. We called *self-consistent* a model that can correct itself, and *universal* a model that can correct all others. Both qualities are theoretical properties of the model families. Thus, they can be genuinely evaluated by realistic synthetic experiments. Self-consistency and universality will be tested by generating a distortion with any of the above models, and then evaluating the error incurred when correcting the generated distortion with any of the above models.

5.3.1 Experiments with Known Distortion Center

We shall first assume that the distortion center (x_c, y_c) is known. To test the self-consistency of a certain model, its direct model in Eq. (5.6) was used to generate a distortion with realistic

coefficients (see Table 5.1). This distortion was corrected by identifying the best parameters in the same model,

$$\begin{aligned} \bar{x}_u &= g_x(\bar{x}_d, \bar{y}_d) \\ \bar{y}_u &= g_y(\bar{x}_d, \bar{y}_d) \end{aligned} \tag{5.7}$$

where the form of g_x, g_y depends on the model selected. In the synthetic test, $(\bar{x}_u, \bar{y}_u)$ and $(\bar{x}_d, \bar{y}_d)$ are both known. The unknowns are the parameters of g_x and g_y. So the question is how well we can approach the ideal correction $(\bar{x}_u, \bar{y}_u)$ by $g_x(\bar{x}_d, \bar{y}_d)$ and $g_y(\bar{x}_d, \bar{y}_d)$. We want to compute the coefficients of g_x and g_y by minimizing the difference between the ideal correction and the practical correction. The energy to be minimized can be written as

$$\begin{aligned} C = \int_0^{X_d} \int_0^{Y_d} \big(g_x(\bar{x}_d, \bar{y}_d) - \bar{x}_u\big)^2 \\ + \big(g_y(\bar{x}_d, \bar{y}_d) - \bar{y}_u\big)^2 \mathrm{d}x_d \mathrm{d}y_d. \end{aligned} \tag{5.8}$$

The distortion center being known, the unknown parameters are the parameters in g_x and g_y. In practice, the simulation is performed on M samples $(x_{u_i}, y_{u_i}), i = 1, \cdots, M$ regularly distributed on an image. The corresponding distorted samples $(x_{d_i}, y_{d_i}), i = 1, \cdots, M$ are obtained by Eq (5.6). The discrete energy to be minimized is

$$\begin{aligned} D = \sum_{i=1}^{M} D_i^2 = \sum_{i=1}^{M} \big(g_x(\bar{x}_{d_i}, \bar{y}_{d_i}) - \bar{x}_{u_i}\big)^2 + \\ + \big(g_y(\bar{x}_{d_i}, \bar{y}_{d_i}) - \bar{y}_{u_i}\big)^2 \end{aligned} \tag{5.9}$$

For the radial and the polynomial models, this problem can be formalized as a linear system by computing the derivatives of D with respect to unknown parameters respectively, and setting them to zero:

$$\mathbf{A}\mathbf{k} = \mathbf{b} \tag{5.10}$$

with $\mathbf{A}$ the coefficient matrix, $\mathbf{k}$ the unknown coefficient vector. The optimal solution minimizing the norm $\|\mathbf{A}\mathbf{k} - \mathbf{b}\|$ is $\mathbf{k} = \left(\mathbf{A}^T\mathbf{A}\right)^{-1}\mathbf{A}^T\mathbf{b}$. In practice, the coefficient matrix $\mathbf{A}$ is ill-conditioned and can make the solution unstable. The following normalization can be applied to make the linear system more stable. $\mathbf{A}$ is multiplied by normalization matrices $\mathbf{T}_1$ and $\mathbf{T}_2$ so that the entries of the normalized matrix $\hat{\mathbf{A}}$ do not vary a lot.

$$\hat{\mathbf{A}}\hat{\mathbf{k}} = \mathbf{T}_2\mathbf{A}\mathbf{T}_1\left(\mathbf{T}_1^{-1}\mathbf{k}\right) = \mathbf{T}_2\mathbf{b}, \tag{5.11}$$

chosen so that the entries of $\mathbf{T}_2\mathbf{A}\mathbf{T}_1$ get closer to each other. Then the solution is $\mathbf{k} = \mathbf{T}_1\left(\hat{\mathbf{A}}^T\hat{\mathbf{A}}\right)^{-1}\hat{\mathbf{A}}^T\mathbf{T}_2\mathbf{b}$.

For example, for the radial model of order 4 with coefficients k_0', k_1', k_2', k_3', k_4', the model in Eq. (5.7) has the form

$$\begin{aligned} g_x(\bar{x}_d, \bar{y}_d) &= \bar{x}_d(k_0' + k_1' r_d + k_2' r_d^2 + k_3' r_d^3 + k_4' r_d^4) \\ g_y(\bar{x}_d, \bar{y}_d) &= \bar{y}_d(k_0' + k_1' r_d + k_2' r_d^2 + k_3' r_d^3 + k_4 r_d^4). \end{aligned} \tag{5.12}$$

By some simple computations, the linear system in Eq. (5.10) can be explicitly written as

$$\begin{aligned}\mathbf{Ak} &= \begin{bmatrix} \sum_i r_{d_i}^2 & \sum_i r_{d_i}^3 & \sum_i r_{d_i}^4 & \sum_i r_{d_i}^5 & \sum_i r_{d_i}^6 \\ \sum_i r_{d_i}^3 & & \cdots & & \sum_i r_{d_i}^7 \\ \vdots & & \ddots & & \vdots \\ \sum_i r_{d_i}^6 & & \cdots & & \sum_i r_{d_i}^{10} \end{bmatrix} \begin{pmatrix} k'_0 \\ k'_1 \\ k'_2 \\ k'_3 \\ k'_4 \end{pmatrix} \\ = \mathbf{b} &= \begin{pmatrix} \sum_i r_{d_i}^2 \left(\bar{x}_{u_i}\bar{x}_{d_i} + \bar{y}_{u_i}\bar{y}_{d_i}\right) \\ \sum_i r_{d_i}^3 \left(\bar{x}_{u_i}\bar{x}_{d_i} + \bar{y}_{u_i}\bar{y}_{d_i}\right) \\ \sum_i r_{d_i}^4 \left(\bar{x}_{u_i}\bar{x}_{d_i} + \bar{y}_{u_i}\bar{y}_{d_i}\right) \\ \sum_i r_{d_i}^5 \left(\bar{x}_{u_i}\bar{x}_{d_i} + \bar{y}_{u_i}\bar{y}_{d_i}\right) \\ \sum_i r_{d_i}^6 \left(\bar{x}_{u_i}\bar{x}_{d_i} + \bar{y}_{u_i}\bar{y}_{d_i}\right) \end{pmatrix}. \end{aligned} \tag{5.13}$$

The entries of $\mathbf{A}$ differ by a big ratio $\frac{\sum_i r_{d_i}^{10}}{\sum_i r_{d_i}^2}$, which cause an numerical instability of the linear system. The normalization matrices $\mathbf{T}_1$ and $\mathbf{T}_2$ used to lessen the instability in Eq. (5.11) can be computed explicitly as

$$\begin{aligned}\mathbf{T}_1 &= \begin{bmatrix} \frac{1}{\sum_i r_{d_i}^2} & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{\sum_i r_{d_i}^3} & \cdots & & 0 \\ \vdots & & \ddots & & \vdots \\ 0 & \cdots & & & \frac{1}{\sum_i r_{d_i}^6} \end{bmatrix} \\ \mathbf{T}_2 &= \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & \frac{\sum_i r_{d_i}^2}{\sum_i r_{d_i}^3} & \cdots & & 0 \\ \vdots & & \ddots & & \vdots \\ 0 & \cdots & & & \frac{\sum_i r_{d_i}^2}{\sum_i r_{d_i}^6} \end{bmatrix}. \end{aligned} \tag{5.14}$$

The same procedures can be used to test the universality of models. Only the polynomial model and the radial model (with fixed distortion center) can be solved by a linear method. For all the other models, a non-linear method must be used, even if (x_c, y_c) is known. The minimization is performed by first doing an incremental Levenberg-Marquardt (LM) algorithm (see Appendix A.3) which estimates the parameters in increasing order. The algorithm starts estimating the parameters of a low order model; its results are used to initialize the optimization of the the model with an order incremented by 1, and the process continues until the aimed order. The Jacobian matrix

$$\mathbf{J} = \begin{bmatrix} \frac{\partial D_1}{\partial k'_0} & \frac{\partial D_1}{\partial k'_1} & \cdots & \frac{\partial D_1}{\partial k'_{N-1}} & \frac{\partial D_1}{\partial k'_N} \\ \frac{\partial D_2}{\partial k'_0} & \frac{\partial D_2}{\partial k'_1} & \cdots & \frac{\partial D_2}{\partial k'_{N-1}} & \frac{\partial D_2}{\partial k'_N} \\ \vdots & & \ddots & & \vdots \\ \frac{\partial D_M}{\partial k'_0} & \frac{\partial D_M}{\partial k'_1} & \cdots & \frac{\partial D_M}{\partial k'_{N-1}} & \frac{\partial D_M}{\partial k'_N} \end{bmatrix}$$

$\mathbf{J}$ made of the partial derivatives of each signed energy component D_i $(i = 1, \cdots, M)$ with respect to each unknown parameter k'_j $(j = 1, \cdots, N)$ is computed explicitly to make the

incremental LM algorithm efficient. The self-consistency and universality properties of all models are recapitulated in Table 5.2 and the parameters for generating the distortion are in Table 5.1. $M = 5104$ points were regularly distributed in an image domain with size 1761×1174. They were first distorted by applying one kind of distortion (indicated in the entry row of Table 5.2) and then corrected by another model (indicated in the entry column of Table 5.2). The distortion center was fixed at the center $(880.5, 587)$ of the image and was assumed to be known. Table 5.2 shows the average error $\bar{D}$ and the maximal error D_∞:

$$\begin{aligned} \bar{D} &= \sqrt{D(k'_0, k'_1, k'_2, \cdots)/M} && (5.15) \\ D_\infty &= \max_i |D_i(k'_0, k'_1, k'_2, \cdots)| && (5.16) \end{aligned}$$

after estimating the parameters $k'_0, k'_1, k'_2, \cdots$ which minimize the energy in Eq (5.9).

The Rational Function Model This model is somewhat an exception. In [68] it is solved linearly by using a "lifted process" technique. This model is in fact designed to recover the 3D point $\mathbf{d}(i, j)$ on the *normalized image plane* represented in camera based coordinate system from the distorted point (i, j). The point $\mathbf{d}(i, j)$ is on the line passing through the undistorted point and the optical center:

$$\begin{aligned} \mathbf{d}(i,j) &= \mathbf{A}\mathcal{X}(i,j) \\ &= \begin{pmatrix} \mathbf{A}_{11}i^2 + \mathbf{A}_{12}ij + \cdots + \mathbf{A}_{15}j + \mathbf{A}_{16} \\ \mathbf{A}_{21}i^2 + \mathbf{A}_{22}ij + \cdots + \mathbf{A}_{25}j + \mathbf{A}_{26} \\ \mathbf{A}_{31}i^2 + \mathbf{A}_{32}ij + \cdots + \mathbf{A}_{35}j + \mathbf{A}_{36} \end{pmatrix} \end{aligned} \quad (5.17)$$

where $\mathcal{X}(i, j) = (i^2, ij, j^2, i, j, 1)^T$ is the "lifted" coordinate of the distorted image point $(i, j) = (x_d, y_d)$; $\mathbf{A}$ the extended camera calibration matrix:

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} & \mathbf{A}_{13} & \mathbf{A}_{14} & \mathbf{A}_{15} & \mathbf{A}_{16} \\ \mathbf{A}_{21} & \mathbf{A}_{22} & \mathbf{A}_{23} & \mathbf{A}_{24} & \mathbf{A}_{25} & \mathbf{A}_{26} \\ \mathbf{A}_{31} & \mathbf{A}_{32} & \mathbf{A}_{33} & \mathbf{A}_{34} & \mathbf{A}_{35} & \mathbf{A}_{36} \end{bmatrix}. \quad (5.18)$$

If there is no distortion, $\mathbf{A}$ is degenerated to be a 3×3 matrix, which is the inverse of the camera calibration matrix $\mathbf{K}^{-1}$. The half-line $\lambda\mathbf{d}(i, j)$ $(\lambda \neq 0)$ is the 3D back-projected ray, passing through the camera optical center and the undistorted 2D image point. The inhomogeneous coordinate of $\mathbf{d}(i, j)$ is $(p, q)^T$ defined by

$$\begin{aligned} p &= \frac{\mathbf{A}_{11}i^2 + \mathbf{A}_{12}ij + \cdots + \mathbf{A}_{15}j + \mathbf{A}_{16}}{A_{31}i^2 + \mathbf{A}_{32}ij + \cdots + \mathbf{A}_{35}j + \mathbf{A}_{36}} \\ q &= \frac{\mathbf{A}_{21}i^2 + \mathbf{A}_{22}ij + \cdots + \mathbf{A}_{25}j + \mathbf{A}_{26}}{A_{31}i^2 + \mathbf{A}_{32}ij + \cdots + \mathbf{A}_{35}j + \mathbf{A}_{36}} \end{aligned} \quad (5.19)$$

which is the coordinate of the undistorted image point on the *normalized image plane* (not on the physical CCD image plane). There is an unknown homography between the corrected point by the rational function model and the undistorted point in the image plane. In practice, the undistorted point in the image plane is what we are looking for, and is not available. To

find the matrix $\mathbf{A}$, a planar pattern containing known feature points $\mathbf{x}_i$ can be used. The extended calibration matrix $\mathbf{A}$ is a bridge linking $\mathbf{x}_i$ and the lift correspondence $\mathcal{X}_i$:

$$\begin{aligned} \mathbf{H}\mathbf{x}_i &= \lambda_i \mathbf{A}\mathcal{X}_i \\ \iff \mathbf{x}_i &= \lambda_i \mathbf{H}^{-1}\mathbf{A}\mathcal{X}_i = \lambda_i \mathbf{A}'\mathcal{X}_i \\ \iff [\mathbf{x}_i]_\times \mathbf{A}'\mathcal{X}_i &= \mathbf{0} \end{aligned} \tag{5.20}$$

Recall that $\mathbf{A}\mathcal{X}_i$ is the homogeneous coordinate of the projection of a 3D point on the normalized image plane. So here the unknown homography $\mathbf{H}$ sends the points from the pattern to the normalized image plane. $\mathbf{A}' = \mathbf{H}^{-1}\mathbf{A}$ will recover the point on the original pattern. One pair of correspondence gives two equations for $\mathbf{A}'$, so 9 pairs of correspondences are sufficient to estimate $\mathbf{A}'$. Note that if $\mathbf{A}$ rectifies the camera into a pinhole camera, then $\mathbf{A}'$ does it too, in spite of the hidden $\mathbf{H}$. This linear algorithm can be directly applied in our synthetic test, where the known undistorted points replace the points on the pattern.

But like many other linear algorithms in multi-view geometry, this "lifted technique" minimizes the algebraic error, which is not directly related to the geometric error. Sometimes a small algebraic error can give a big geometric error. So in the simulation, the parameters of the rational function model are still estimated by the incremental LM algorithm by using the result of the linear "lifted technique" as an initialization.

model	parameters
radial 2° 1084 → 1050	$k_0 = 1.0$, $k_1 = 0.25\text{e}{-4}$, $k_2 = -0.5\text{e}{-7}$
radial 4° 991.6 → 1050	$k_0 = 1.0$, $k_1 = 0.25\text{e}{-4}$, $k_2 = -0.5\text{e}{-7}$, $k_3 = 1.0\text{e}{-10}$, $k_4 = -1.5\text{e}{-14}$
division 2° 1083 → 1050	$d_0 = 1.0$, $d_1 = -0.25\text{e}{-4}$, $d_2 = 0.5\text{e}{-7}$
division 4° 988.7 → 1050	$d_0 = 1.0$, $d_1 = -0.25\text{e}{-4}$, $d_2 = 0.5\text{e}{-7}$, $d_3 = -1.0\text{e}{-10}$, $d_4 = 1.5\text{e}{-14}$
FOV 3° 501.4 → 1050	$k_0 = 1.0$, $\omega = 1.0 \times 10^{-3}$, $k_2 = -2.0 \times 10^{-7}$, $k_3 = 4.0 \times 10^{-10}$
polynomial 3° 1050 → 1064	$a_1 = b_1 = -1.0\text{e}{-8}, \cdots, a_5 = b_5 = 2.0\text{e}{-5}, \cdots,$ $a_8 = 0.9$, $a_9 = 0.1$, $a_{10} = 0.0$, $b_8 = 0.1$, $b_9 = 0.9$, $b_{10} = 0.0$
polynomial 4° 1050 → 1075	$a_1 = b_1 = 5.0\text{e}{-12}$, $a_6 = b_6 = -1.0\text{e}{-8}$, $a_{10} = b_{10} = 2.0\text{e}{-5}, \cdots,$ $a_{13} = 0.9$, $a_{14} = 0.1$, $a_{15} = 0.0$, $b_{13} = 0.1$, $b_{14} = 0.9$, $b_{15} = 0.0$
rational 2° 1031 → 1104	$a_1 = 1.0 \times 10^{-5}$, $a_2 = 2.0 \times 10^{-5}$, $a_3 = 3.0 \times 10^{-5}$, $a_4 = 0.9$, $a_5 = 0.1$, $a_6 = 0.0$ $b_1 = 3.0 \times 10^{-5}$, $b_2 = 2.0 \times 10^{-5}$, $b_3 = 1.0 \times 10^{-5}$, $b_4 = 0.1$, $b_5 = 0.9$, $b_6 = 0.0$ $c_1 = 1.0 \times 10^{-8}$, $c_2 = 1.0 \times 10^{-8}$, $c_3 = 1.0 \times 10^{-8}$, $c_4 = 0.0001$, $c_5 = 0.0001$, $c_6 = 1.0$

Table 5.1: Models used to generate distortion, with their realistic parameters. The values on the left and on the right of → are the undistorted radius and the distorted radius respectively. For the polynomial model, the coefficients are the same for x and y component, except for the order 1 coefficients. Note that the distortion can be barrel, pincushion or mustache.

5.3.2 Experiments with Unknown Distortion Center

In practice, the distortion center (x_c, y_c) is unknown. It should also be considered as a parameter in the minimization formulation. The minimization problem becomes non-linear

	R 2°	R 4°	D 2°	D 4°	F 3°	P 3°	P 4°	Ra 2°
R 2°	9e−2/8e−1	1e−1/4e−1	6e−2/5e−1	2e−1/5e−1	3e−2/1e−1	6e+1/2e+2	6e+1/2e+2	7e+1/2e+2
R 4°	2e−3/3e−2	2e−3/2e−2	8e−4/9e−3	2e−3/8e−3	8e−4/6e−3	6e+1/2e+2	6e+1/2e+2	7e+1/2e+2
D 2°	6e−2/6e−1	2e−1/6e−1	3e−2/3e−1	2e−1/1e+0	4e−2/2e−1	6e+1/2e+2	6e+1/2e+2	7e+1/2e+2
D 4°	1e−3/2e−2	1e−3/9e−3	4e−4/4e−3	1e−3/7e−3	7e−4/6e−3	6e+1/2e+2	6e+1/2e+2	7e+1/2e+2
F 3°	8e−2/3e−1	7e−2/8e−1	9e−2/4e−1	6e−2/7e−1	2e−2/2e−1	6e+1/2e+2	6e+1/2e+2	7e+1/2e+2
P 3°	6e−1/3e+0	5e−1/2e+0	5e−1/3e+0	6e−1/3e+0	2e−1/8e−1	2e−1/2e+0	7e−1/6e+0	5e−1/3e+0
P 4°	6e−1/3e+0	5e−1/2e+0	5e−1/3e+0	6e−1/3e+0	2e−1/8e−1	5e−2/6e−1	1e−1/2e+0	7e−2/6e−1
P 8°	6e−2/4e−1	2e−2/1e−1	6e−2/4e−1	2e−2/1e−1	7e−3/6e−2	7e−5/1e−3	7e−4/1e−2	4e−5/6e−4
P 15°	1e−2/6e−2	8e−3/5e−2	1e−2/6e−2	8e−3/5e−2	3e−4/1e−3	1e−7/7e−7	2e−7/3e−6	4e−7/4e−6
Ra2°	5e+0/2e+1	7e+0/3e+1	5e+0/2e+1	7e+0/3e+1	4e+1/1e+3	5e−1/3e+0	4e−1/1e+0	1e−1/8e−1
Ra6°	5e−2/2e−1	3e−2/2e−1	1e−1/7e−1	9e−2/7e−1	1e−1/6e−1	2e−6/3e−5	1e−4/2e−3	8e−8/9e−7
Ra10°	4e−2/2e−1	2e−2/1e−1	1e−1/8e−1	9e−2/7e−1	3e−2/3e−1	8e−8/1e−6	2e−7/3e−6	4e−9/5e−8

Table 5.2: Self-consistency and universality with known distortion center. The average error ($\bar{D}$)/maximal error (D_∞) (in pixels) is shown. The left column entries show the model and the order used for correction. The top entry row gives the model and the order used to generate the distortion. The five compared model classes are R-Radial, D-Division, F-FOV, P-Polynomial, and Ra-Rational. The parameters in Table 5.1 were used to generate the distortion. The green color is used to highlight the average error $\bar{D} \leqslant 10^{-2}$, the blue color for $10^{-2} < \bar{D} \leqslant 10^{-1}$ and the red color for $\bar{D} > 10^{-1}$.

for most models if (x_c, y_c) is unknown. This is true for the radial model, the division model, the FOV model and the rational model. In contrast the polynomial and the rational function models are invariant to a translation of the distortion center. The point (x_c, y_c) can be fixed arbitrarily, and in the polynomial case the minimization problem is linear while the rational model is still non-linear. This is a decisive advantage with respect to the other models. The self-consistency and universality results are recapitulated in Table 5.3 with the parameters for generating distortion in Table 5.1. For the distortion generation, the distortion center was fixed at the center $(880.5, 587)$ of the image, while for the correction, the initial distortion center was realistically taken $(50, 50)$ pixels away from the true position. For the radial model and the division model, the Levenberg-Marquardt algorithm could still find the true distortion center, and the minimized error was the same as when the distortion center was known. Nevertheless, for the FOV model, a bad initialization of the distortion center degraded the correction performance. For the polynomial model, the solution can be found linearly by fixing an arbitrary distortion center. For the rational function model, even though it is invariant to a translation of the distortion center, incremental LM algorithm cannot ensure the correct minimization.

5.3.3 Comparison

The tables show that the models are self-consistent for an average precision of the order of 10^{-2} pixel if the order of correction is high enough when the distortion center is known. The radial model and the division model are consistent with each other, whether the distortion center is known or not. The FOV model is a little less consistent with the radial model and the division model when the distortion center is known. With an unknown distortion center,

	R 2°	R 4°	D 2°	D 4°	F 3°	P 3°	P 4°	Ra 2°
R 2°	9e−2/8e−1	1e−1/4e−1	6e−2/5e−1	2e−1/5e−1	3e−2/1e−1	6e+1/2e+2	6e+1/2e+2	7e+1/2e+2
R 4°	2e−3/3e−2	2e−3/2e−2	8e−4/9e−3	2e−3/8e−3	8e−4/6e−3	6e+1/2e+2	6e+1/2e+2	7e+1/2e+2
D 2°	6e−2/6e−1	2e−1/6e−1	3e−2/3e−1	2e−1/1e+0	4e−2/2e−1	6e+1/2e+2	6e+1/2e+2	7e+1/2e+2
D 4°	1e−3/2e−2	1e−3/9e−3	4e−4/4e−3	1e−3/7e−3	2e+1/4e+1	6e+1/2e+2	6e+1/2e+2	7e+1/2e+2
F 3°	7e−1/2e+0	3e+0/2e+1	2e+0/9e+0	3e+0/2e+1	4e+1/6e+1	6e+1/2e+2	6e+1/2e+2	7e+1/2e+2
P 3°	6e−1/3e+0	5e−1/2e+0	5e−1/3e+0	6e−1/3e+0	2e−1/8e−1	2e−1/2e+0	7e−1/6e+0	5e−1/3e+0
P 4°	6e−1/3e+0	5e−1/2e+0	5e−1/3e+0	6e−1/3e+0	2e−1/8e−1	5e−2/6e−1	1e−1/2e+0	7e−2/6e−1
P 8°	6e−2/4e−1	2e−2/1e−1	6e−2/4e−1	2e−2/1e−1	7e−3/6e−2	7e−5/1e−3	7e−4/1e−2	4e−5/6e−4
P 15°	1e−2/6e−2	8e−3/5e−2	1e−2/6e−2	8e−3/5e−2	3e−4/1e−3	1e−7/7e−7	2e−7/3e−6	4e−7/4e−6
Ra2°	7e+1/9e+1	7e+1/1e+2	7e+1/9e+1	7e+1/1e+2	7e+1/8e+1	7e+1/7e+1	7e+1/7e+1	7e+1/7e+1
Ra6°	7e+1/7e+1	7e+1/7e+1	7e+1/7e+1	7e+1/7e+1	7e+1/7e+1	7e+1/7e+1	7e+1/7e+1	7e+1/7e+1
Ra10°	7e+1/7e+1	7e+1/7e+1	7e+1/7e+1	7e+1/7e+1	7e+1/7e+1	7e+1/7e+1	7e+1/7e+1	7e+1/7e+1

Table 5.3: Self-consistency and universality with unknown distortion center. The initial distortion center was set $(50, 50)$ pixels away from its true position. The average error $(\bar{D})$/maximal error (D_∞) (in pixels) is shown. The left column entries give the model and the order used for correction. The top row entries give the model and the order used to generate the distortion. The five compared model classes are R-Radial, D-Division, F-FOV, P-Polynomial, and Ra-Rational. The parameters in Table 5.1 were used to generate the distortion. The green color is used to highlight the average error $\bar{D} \leqslant 10^{-2}$, the blue color for $10^{-2} < \bar{D} \leqslant 10^{-1}$ and the red color for $\bar{D} > 10^{-1}$.

the FOV correction performance decays. The polynomial model instead seems to be able to correct any type of distortion, but a higher order is often necessary to correct the radial, division or FOV distortions. This higher order is not a problem, because of the computational efficiency of the linear method. The rational function model should have the same performance as the polynomial model. But due to the complexity in the non-linear minimization, it is often stuck by the local minima. *In conclusion, the polynomial model is the only one to be jointly self-consistent, universal and linear among the compared models.*

5.3.4 Realistic Distortion

A real distortion can be far more complex than what the above simple models can generate. A more realistic distortion contains a radial symmetric term, a term for decentering distortion and a term for thin prism distortion [177],

$$\begin{aligned}
\bar{x}_d &= \bar{x}_u \left(k_0 + k_1 r_u + k_2 r_u^2 + \cdots\right) \\
&+ \left[p_1 \left(r_u^2 + 2\bar{x}_u^2\right) + 2p_2 \bar{x}_u \bar{y}_u\right] \left(1 + p_3 r_u^2\right) + s_1 r_u^2 \\
\bar{y}_d &= \bar{y}_u \left(k_0 + k_1 r_u + k_2 r_u^2 + \cdots\right) \\
&+ \left[p_2 \left(r_u^2 + 2\bar{y}_u^2\right) + 2p_1 \bar{x}_u \bar{y}_u\right] \left(1 + p_3 r_u^2\right) + s_2 r_u^2
\end{aligned}$$

with p_1, p_2, p_3 parameters for decentering distortion and s_1, s_2 parameters for thin prism distortion. They are both tangential distortions. In Table 5.4, the self-consistency and universality of the models were again tested with known distortion center after adding a tangential distortion with $p_1 = 4.0\text{e}{-6}, p_2 = -2.0\text{e}{-6}, p_3 = 0, s_1 = 3.0\text{e}{-6}, s_2 = 1.0\text{e}{-6}$. By adding this

non-radial component in the distortion, the radial model, the division model and the FOV model do not reach anymore the 10^{-2} pixel precision since they are all radial symmetric. These three models are almost consistent to each other (see Table 5.2), which explains why the erros in the first five rows are on the same order of magnitude. As for the rational function model, it again has the minimization problem. The polynomial model is the only model getting a higher precision when increasing the model order.

	R 2°	R 4°	D 2°	D 4°	F 3°	P 3°	P 4°	Ra 2°
R 2°	7e+0/3e+1	5e+0/2e+1	7e+0/3e+1	5e+0/2e+1	7e−1/2e+0	6e+1/2e+2	6e+1/2e+2	7e+1/2e+2
R 4°	7e+0/3e+1	5e+0/2e+1	7e+0/3e+1	5e+0/2e+1	7e−1/2e+0	6e+1/2e+2	6e+1/2e+2	7e+1/2e+2
D 2°	7e+0/3e+1	5e+0/2e+1	7e+0/3e+1	5e+0/1e+1	7e−1/2e+0	6e+1/2e+2	6e+1/2e+2	7e+1/2e+2
D 4°	7e+0/3e+1	5e+0/2e+1	7e+0/3e+1	5e+0/2e+1	7e−1/2e+0	6e+1/2e+2	6e+1/2e+2	7e+1/2e+2
F 3°	7e+0/3e+1	5e+0/2e+1	7e+0/3e+1	5e+0/2e+1	7e−1/2e+0	6e+1/2e+2	6e+1/2e+2	7e+1/2e+2
P 3°	6e−1/4e+0	5e−1/4e+0	5e−1/4e+0	6e−1/4e+0	2e−1/9e−1	3e−1/3e+0	8e−1/7e+0	3e−1/2e+0
P 4°	6e−1/3e+0	5e−1/2e+0	5e−1/3e+0	6e−1/3e+0	2e−1/8e−1	6e−2/7e−1	2e−1/2e+0	5e−2/4e−1
P 8°	6e−2/4e−1	2e−2/1e−1	6e−2/4e−1	2e−2/1e−1	7e−3/6e−2	1e−4/2e−3	1e−3/2e−2	1e−5/2e−4
P 15°	1e−2/6e−2	8e−3/5e−2	1e−2/6e−2	8e−3/5e−2	3e−4/2e−3	3e−7/2e−6	3e−7/6e−6	3e−7/4e−6
Ra2°	5e+0/2e+1	7e+0/3e+1	5e+0/2e+1	7e+0/3e+1	3e+0/1e+1	6e−1/3e+0	4e−1/1e+0	1e−1/8e−1
Ra6°	1e−1/9e−1	1e−1/8e−1	1e−1/9e−1	9e−2/8e−1	3e−2/3e−1	2e−6/3e−5	2e−4/2e−3	3e−7/5e−6
Ra10°	1e−1/9e−1	1e−1/8e−1	1e−1/8e−1	9e−2/7e−1	3e−2/2e−1	8e−8/1e−6	3e−7/5e−6	3e−9/6e−8

Table 5.4: Self-consistency and universality with known distortion center. Compared with Tables 5.2 and 5.3, besides the distortion generated by the parameters in Table 5.1, an additional tangential distortion is added. Each entry shows the average error ($\bar{D}$)/maximal error (D_∞) (in pixels). The green color is used to highlight the average error $\bar{D} \leqslant 10^{-2}$, the blue color for $10^{-2} < \bar{D} \leqslant 10^{-1}$ and the red color for $\bar{D} > 10^{-1}$. The only blue-green to green lines are obtained for the polynomial model with degree 8 to 15.

5.4 Real Distortion Fitting Experiments

After its validation on synthetic examples, we present here real tests to verify that the proposed high order polynomial model works for real distortion correction. This test was inspired from the non-parametric lens distortion estimation method in Chapter 4 [80] but could be performed on any distortion model obtained by blind correction. This method requires a highly textured planar pattern, which is obtained by printing a textured image and pasting it on a very flat object (a mirror was used in the experiments). Two photos of the pattern were taken by a Canon EOS 30D SLR camera with EFS $18-55$mm lens. The minimal focal length (18mm) was chosen (with fixed focus) to produce a fairly large distortion (see Fig. 5.2 for the digital textured image and two photos of the pattern). The distortion was estimated (up to a homography) as the diffeomorphism mapping the original digital pattern to a photograph of it. The algorithm is summarized in the following (see Chapter 4 for details).

1. Take two slightly different photographs of a textured planar pattern with a camera whose settings are frozen;

2. apply the SIFT method [117] between the original digital pattern and both photographs to find matchings;

3. eliminate outliers by a loop validation step;

4. refine the precision of the SIFT matchings by moving each point in one image by applying the local homography estimated from its neighboring matchings;

5. triangulate and interpolate the remaining matchings to get a dense reverse distortion field;

6. by applying the reverse distortion field to all images produced by the real camera, the camera is converted into a virtual pinhole camera.

The matchings delivered by step 4 (about 8000 matchings in our experiments) in the above algorithm are "outliers"-free and precise thanks to the loop validation and local homography. So we can directly try all models to fit these "outliers"-free matchings (the distortion center is also estimated except the polynomial model and rational model whose distortion center is fixed at the center of image). The residual fitting error shows to what extent the models are faithful to a real camera lens distortion. In addition, there is an arbitrary homography between the digital pattern and its photograph. So the compatibility of the models with a homography is also implicitly tested. We used 50% matchings to estimate the parameters for different models and the other 50% to evaluate the fitting error. The results are recapitulated in Table 5.5, compared to the non-parametric method [80]. They show that all of the radial symmetric models fail (including the radial, division and FOV models) because the distortion field is not radial symmetric due to the implicit unknown homography. In this experiment, the rational model gives a performance similar to the polynomial model, but rather *by chance* (in contrast, in the synthetic test, the incremental LM algorithm does not find a global minimum). The fitting error of the polynomial model becomes stable when its order attains 7, which means that it does not suffer from numerical instability or noise fitting. *The precision attained with the polynomial model is about 500 times higher than with classic models!* The non-parametric method gives a slightly larger fitting error than the polynomial model because the triangulation (step 5 in the above summarized algorithm) at the border of image can be imprecise (see the difference at the border for non-parametric method and the polynomial model in Fig. 5.4 and 5.3).

5.5 Plumb-Line Validation

It should be noted that the non-parametric method does not give a ground truth. It is just a non-parametric estimation of the lens distortion, and it is subject to errors. These errors are evident in that the non-parametric model has a mean fitting error of 0.18 pixels. Thus, we need a more objective evaluation to check the quality of the correction polynomial model. To this purpose, a physical frame with tightly stretched cylindrical strings was built. The physical tension of the strings guarantees a very high straightness. Once the parameters of the models of different orders are estimated by "outliers"-free matchings fitting (section 5.4), a distortion field can be constructed and applied for the distortion correction of images of strings taken by the same camera with the same fixed lens configuration (see Fig. 5.4). The average/max distance from the edge points (detected by Devernay's method [52]) of the corrected lines to the corresponding regression line was computed. Table 5.6 recapitulates average/max distance

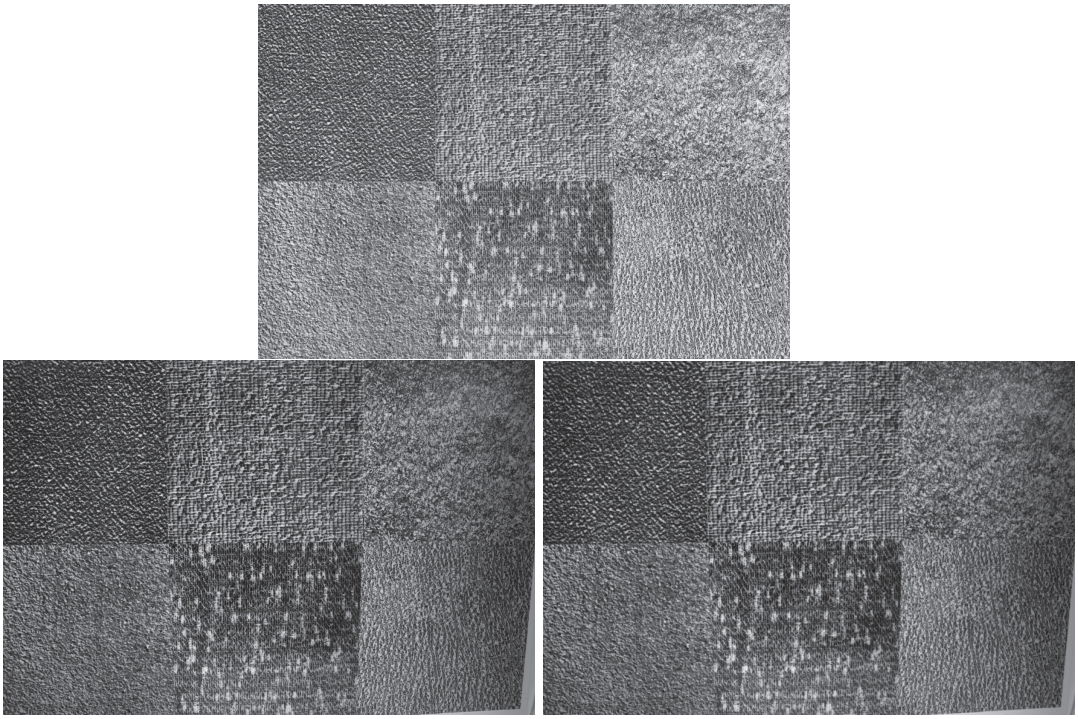


Figure 5.2: The textured pattern and two photos. Top row: the digital textured pattern. Bottom row: two similar photos of the pattern.

for all lines in the image. The polynomial model still gives a stable performance when the order attains 6, *which means that a polynomial model of order* 6 *was already capable of capturing the whole distortion.* The residual fitting error comes from the noise of matching points. The polynomial model has far too few parameters to fit this noise, which guarantees the correction quality and stability. The rational model gives a comparable performance at the price of much higher computational cost (more than 200 times slower). All of the other parametric models do not give a satisfactory result for the reason explained above (their average/max error is in fact larger than shown because the used line segment detector [174] sometimes detects only one part of the non-corrected lines). The non-parametric method gives a performance very close to the polynomial model, which confirms again the effectiveness of the polynomial model (see Fig. 5.4 for the visual inspection).

Remark: In Table 5.6, the fit distortion models are used to correct the distorted lines. Since the fit distortion models are estimated by fitting the "outliers"-free matchings between the digital pattern and one of its photos, the increase of order of models decrease the average fitting error as shown in Table 5.5. But the fitting error is not directly related to the straightness error in Table 5.6. Only when the fit models can well approximate the diffeomorphism between the digital pattern and its photo (homography followed by lens distortion), the straight lines can be well corrected and the straightness error decreases with the model order. This is the case of polynomial model and rational model. But for the radial model, division model and FOV model, since they are not capable to approximate the the diffeomorphism between the digital pattern and its photo (Table 5.5), the distortion lines are not well corrected. So the

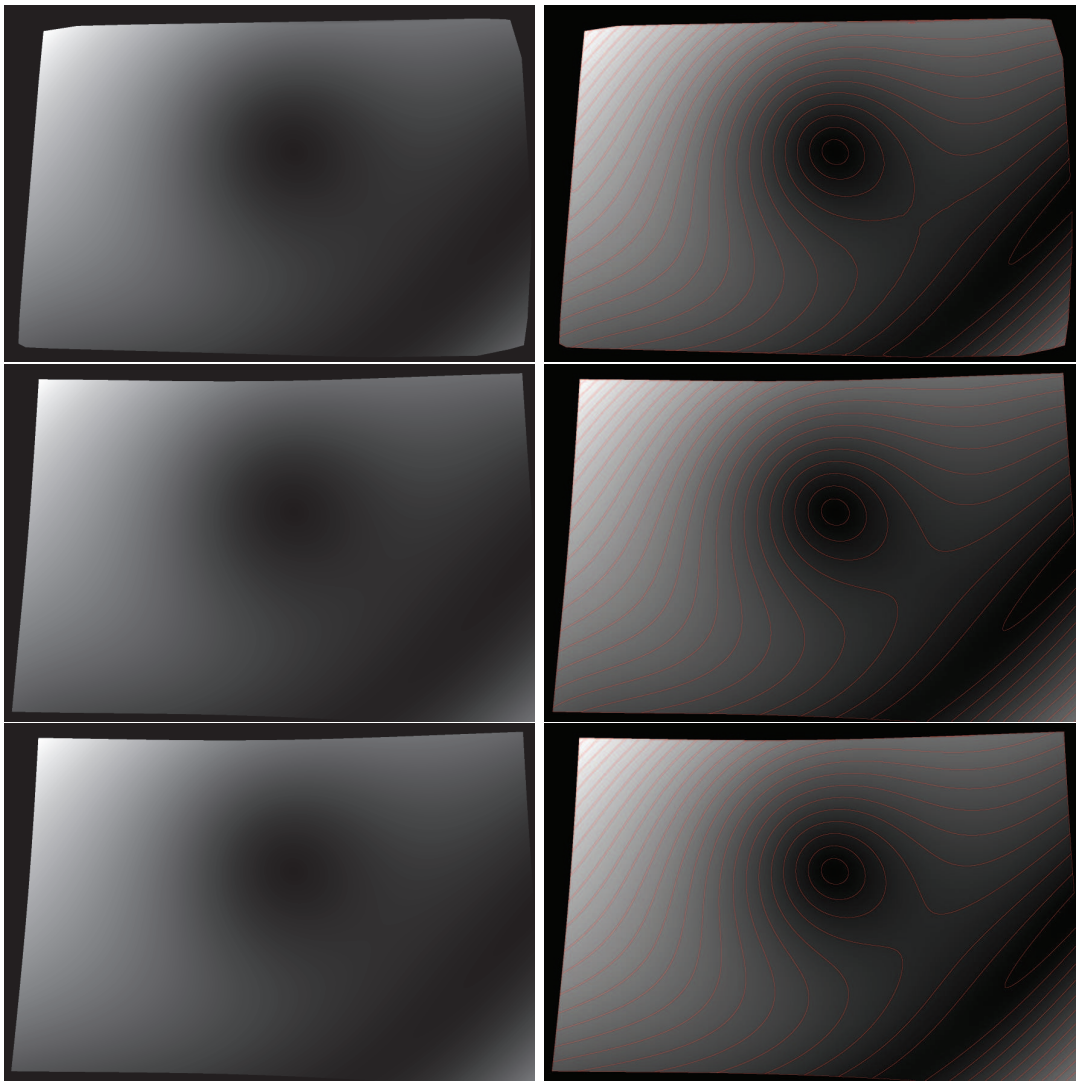


Figure 5.3: The distortion field estimated by different methods. Top row: the distortion field obtained by a non-parametric method [80] and its level lines (in red) with quantization step 20. Middle row: the distortion field constructed by the estimated parameters of polynomial model of order-12 and the level lines (in red) with quantization step of 20. Bottom row: the distortion field constructed by the estimated parameters of rational function model of order-12 and the level lines (in red) with quantization step of 20.

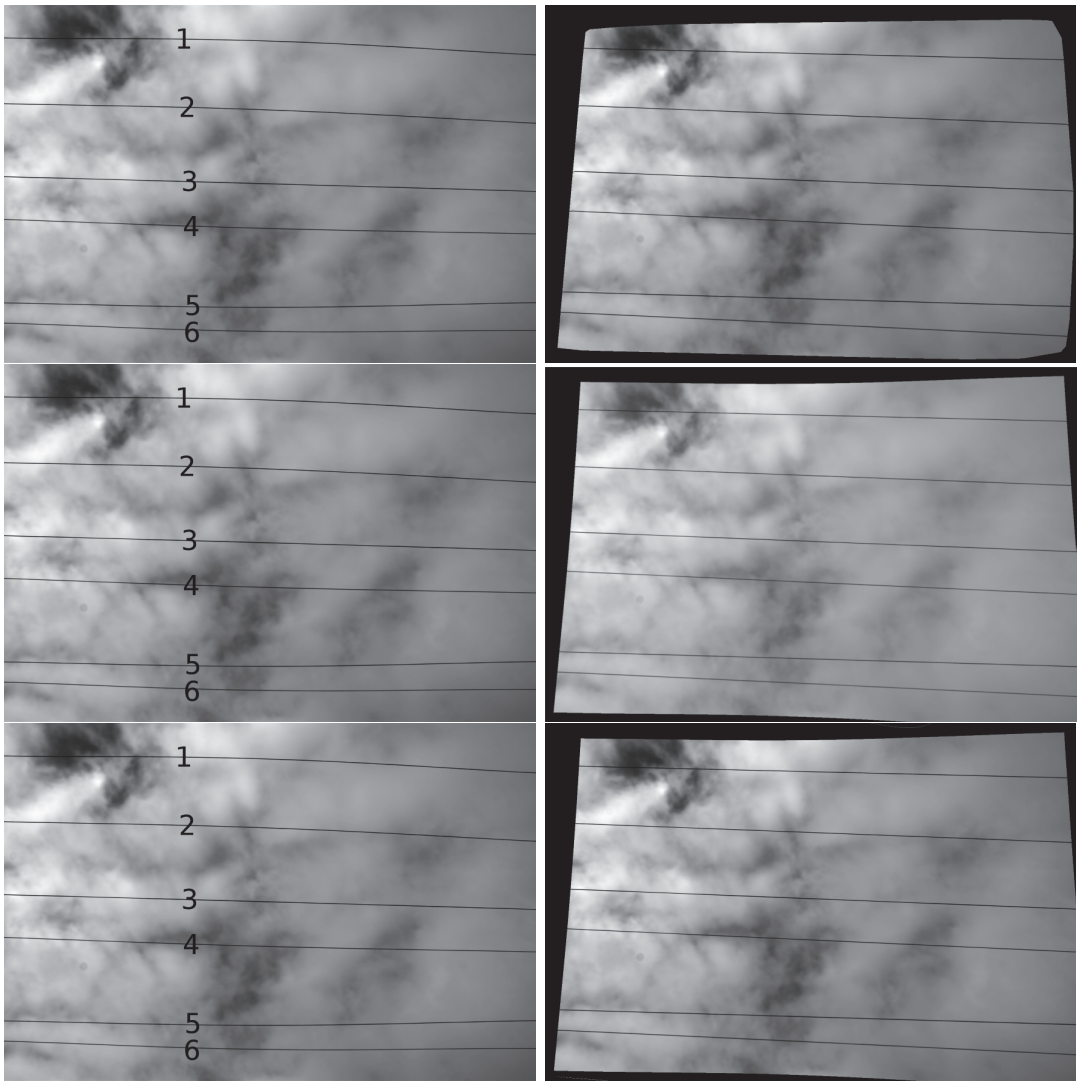


Figure 5.4: Top row: distorted image of tightly stretched lines and the corrected image by the non-parametric method. Middle row: distorted images of tightly stretched lines and the corrected image by the estimated polynomial model. Bottom row: distorted images of tightly stretched lines and the corrected image by the estimated rational function model.

order	Parametric model type					Non-parametric model [80]
	Radial	Division	FOV	Polynomial	Rational	
3	24.66/89.84	26.98/103.30	26.72/101.74	1.48/10.65	0.19/1.86	0.18/6.21
4	24.55/89.08	26.95/103.22	25.47/94.53	1.26/8.93	0.05/0.49	
5	24.30/85.74	26.94/104.32	25.48/95.17	0.21/2.09	0.05/0.33	
6	24.28/85.58	25.43/94.53	25.47/94.87	0.08/0.83	0.05/0.27	
7	24.29/85.45	24.37/86.58	25.17/93.22	0.04/0.35	0.04/0.24	
8	24.27/85.93	24.27/85.87	25.18/93.66	0.04/0.27	0.04/0.87	
9	24.28/86.07	24.28/86.14	25.15/93.63	0.04/0.25	0.04/0.39	
10	24.27/86.24	24.28/86.21	24.28/86.03	0.04/0.22	0.04/0.39	
11	24.26/86.64	24.26/86.62	24.27/86.52	0.04/0.22	0.04/0.39	
12	24.27/86.48	24.27/86.53	24.26/86.57	0.04/0.26	0.04/0.39	

Table 5.5: The fitting error (in pixels) of the compared models to the matchings between a digital textured image and its photograph. The matchings obtained at step 4 in the summarized algorithm are "outliers"-free and precise. Column 1 is the order of the model. 50% matchings are used to estimate the parameters and the shown average/maximal fitting error is computed on the other 50% matchings by applying the estimated parameters. The fitting error of non-parametric method in the last column is computed in the same manner: 50% matchings are fist used to estimate the deformation field, and the other 50% matchings are used to compute the fitting error the estimated deformation field. Remark that since we minimize the average error (not the maximal error), in some instances, the maximal error increases with the increase of order.

increase of model order does not necessarily lead to the better correction of the distorted lines. In addition, the line segment detector used to detect the lines can miss some non-corrected lines.

5.6 Conclusion

We introduced the self-consistency and universality criteria for camera lens distortion models. Using these tools, the five most classic distortion classes of models were evaluated and compared. The polynomial and rational function model were shown to be both self-consistent and universal, to the cost of a high degree. This high degree raises no computational issue for the polynomial model. Indeed, after a correct conditioning it can always be solved linearly. In contrast, the rational model needs to be solved by an incremental Levenberg-Marquardt algorithm initialized by a linear method (even though it is not ensured that the complex non-linear minimization always find a global minimum). Furthermore, the polynomial model is translation invariant, which makes it insensitive to a translation of the distortion center. This model is not adapted to global camera calibration methods where the internal and external parameters and the distortion model are estimated simultaneously. The distortion correction must be dealt with as an independent and previous step to camera calibration. It might be objected that the high number of parameters in the polynomial interpolation (156 for an 11-order polynomial) could cause over-fitting bias in the results. Yet, the number of control points (about 4000) is far higher, about 30 times the number of polynomial coefficients. Our

order	Parametric model type					Non-parametric model [80]
	Radial	Division	FOV	Polynomial	Rational	
3	6.86/36.96	3.77/21.13	4.06/20.70	0.74/4.36	0.11/0.77	0.09/0.44
4	6.91/33.70	3.85/21.75	5.80/33.49	0.58/2.47	0.09/0.51	
5	7.61/32.86	3.78/19.09	5.83/30.72	0.15/0.51	0.09/0.50	
6	7.65/33.58	5.92/29.95	5.89/29.90	0.09/0.45	0.09/0.49	
7	5.53/24.94	6.39/31.21	6.30/33.42	0.09/0.51	0.09/0.49	
8	7.60/32.91	7.59/33.11	5.68/25.26	0.09/0.52	0.09/0.49	
9	6.69/26.71	6.78/26.95	5.83/26.64	0.09/0.52	0.09/0.51	
10	5.42/21.54	6.16/26.11	7.55/33.62	0.09/0.55	0.09/0.52	
11	2.33/8.85	2.38/8.65	3.11/11.30	0.09/0.52	0.09/0.52	
12	1.72/5.15	1.53/4.36	1.95/6.67	0.09/0.51	0.09/0.52	

Table 5.6: The average/max distance (in pixels) from edge points of corrected lines to the corresponding regression line. The parameters of the models are estimated by 50% matchings coming from step 4 in the summarized algorithm. The distorted image in Fig. 5.4 is then corrected by using all models. The corrected lines are extracted by using the algorithm in [174], which is supposed to extract straight lines in images. Note that for the radial model, division model, FOV model, the correction is not satisfying. Sometimes only one part of or even no line can be extracted. So the error is bigger than shown. But all lines are reliably extracted from the image corrected by the polynomial model, rational model or the non-parametric method.

experiments show that the residual errors stabilize for orders between 6 to 12, confirming that no over-fitting occurred. Our experiments also show that high order polynomials are really needed if we wish to obtain high precisions. And they indeed deliver accuracies hundred to thousand times higher than those obtained with classic models.

Chapter 6

High Precision Camera Calibration with a Harp

Contents

Abstract *Plumb line lens distortion correction methods permit to avoid numerical compensation between the camera internal and external parameters in global calibration method. Once the distortion has been corrected by a plumb line method, the camera is ensured to transform, up to the distortion precision, 3D straight lines into 2D straight lines, and therefore becomes a pinhole camera. This chapter introduces a plumb line method for correcting and evaluating camera lens distortion with high precision. The evaluation criterion is defined as the average standard deviation from straightness of a set of approximately equally spaced straight strings photographed uniformly in all directions by the camera, so that their image crosses the whole camera field. The method uses an easily built "calibration harp," namely a frame on which good quality strings have been tightly stretched to ensure a very high physical straightness. Photos of the harp of different orientations are taken to estimate the best coefficients of polynomial model (the most universal and self-consistent model) to correct the distortion. Both the harp of sewing strings and the harp of opaque fishing strings are tested in the experiments. With the harp of sewing strings, the correction precision is better than the non-parametric pattern based method and no global artificial bias is observed. With the harp of opaque fishing strings, the residual oscillation due to braid pattern of sewing strings is largely reduced and the achieved average correction precision is about* 0.02 *pixels This precision is much better than the result given the global camera calibration, which is not stable and varies with the parameters used in the distortion model.*

6.1 Introduction

This chapter presents a method to correct camera lens distortion with high precision. Subpixel precision is not necessary for the human vision which is not affected by distortions of less than 2 pixels. However, there is no limit to the desired precision when the camera is used for 3D reconstruction or photogrammetry tasks. Traditionally, lens distortion and the other camera parameters are estimated simultaneously as camera internal and external parameters [159, 172, 191, 95, 177]. In these global calibration methods all parameters are estimated by minimizing the error between the camera and its numerical model on feature points identified in several views, all in a single non-linear optimization. The result will be precise if (and only if) the model captures the correct physical property of cameras and if the minimization algorithm finds a global minimum. Unfortunately global camera calibration suffers a common drawback: errors in the external and internal camera parameter can be compensated by opposite errors in the distortion model. Thus the residual error can be apparently small, while the distortion model is not precisely estimated [177, 108]. For example the Lavest *et al.* method [95] measures the non-flatness of a pattern and yields a remarkably small re-projection error of about 0.02 pixels, while the straightness of corrected lines has a 0.2 pixel RMSE. This drawback becomes even more serious for high resolution cameras because the same camera orientation/position error causes a larger error measured in pixels, compared to low resolution cameras[1]. As we shall see, the explanation of this discrepancy between the distortion error and re-projection error is that a compensation of errors occur. The distortion model with few parameters is inaccurate but the external camera parameters adapt to compensate the error. This error

[1]Assume a camera CCD covers a 60° field of view and has 0.1° orientation error. For a low resolution camera 512×512 pixels CCD, the 0.1° orientation error corresponds to 0.85 pixels in image, while for a high resolution camera with a 1280×1280 pixels CCD, the error becomes 2.13 pixels.

compensation in global calibration can be avoided by proceeding to distortion correction before camera calibration. Recent distortion correction methods use the correspondences between two or several images, without knowledge of any camera information. The main tool they use is slackened epipolar constraints, which incorporate lens distortion into the epipolar geometry. Several iterative [161, 190] or non-iterative methods [126, 68, 10, 108, 39] are used to estimate the distortion and to correct it. These methods are used with a low order parametric distortion model and therefore cannot achieve high precision.

Non-parametric methods which establish a direct diffeomorphism between a flat pattern and a frontal photograph of it [80, 162] should be ideal for high precision distortion correction. Indeed, they do not depend on the *a priori* choice of a distortion model with a fixed number of parameters. Yet, to achieve a high precision, they depend on the design of a sizeable very flat non deformable plate with highly accurate patterns printed on it[2]. This replaces a technological challenge by another, which is not simpler.

Plumb-line methods [25] should therefore be an alternative, but is it easier to create very straight lines? For plumb-line methods, an appropriate distortion model still is necessary to precisely remove the distortion. Almost all of the existing models can be directly incorporated into a plumb-line method. But some of them are too complicated [25], while some are not general enough to capture the distortion [53]. For most distortion models, the distortion center is a sensitive parameter when a realistic distortion is treated. The bare polynomials proposed in [175] are therefore a good choice, being a translation invariant to the distortion center and linear approximation of any vector field. This model free approximation can approximate complex radial and non-radial distortions as well provided its degree is high enough. According to the criteria of *self-consistency* and *universality* [3] developed in Chapter 5 [175] to compare many camera distortion models, the polynomial models are the most flexible and accurate.

The proposed method is introduced in section 6.2, followed by synthetic and real experiments in section 6.3, along with a comparison to other methods. Section 6.4 discuss the usage of strings of better quality, which leads to better correction precision. The error compensation in global camera calibration is demonstrated in section 6.5. Section 6.6 is a conclusion.

6.2 The Harp Calibration Method

In one sentence, the proposed method combines the advantage of plumb-line methods with the universality of the model free polynomial approximation. The plumb-line method consists in correcting the distorted points which are supposed to be on a straight line, by minimizing the average distance from the corrected points to their corresponding regression lines. In the sequel, denote (x_u, y_u) undistorted point, (x_d, y_d) distorted point, (x_c, y_c) distortion center, $(\bar{x}_u, \bar{y}_u)$ radial undistorted point and $(\bar{x}_d, \bar{y}_d)$ radial distorted point with $\bar{x}_u = x_u - x_c$, $\bar{y}_u = y_u - y_c$, $\bar{x}_d = x_d - x_c$ and $\bar{y}_d = y_d - y_c$. The distorted radius $r_d = \sqrt{\bar{x}_d^2 + \bar{y}_d^2}$ and the undistorted radius $r_u = \sqrt{\bar{x}_u^2 + \bar{y}_u^2}$.

[2] 10 micron flatness is needed to achieve the precision 0.01 pixels.

[3] Self-consistency is evaluated by the residual error when distortion generated with a certain model is corrected (using the model in reverse way) by the best parameters for the same model. Analogously, universality is measured by the residual error when a model is used to correct distortions generated by a family of other models. A model is self-consistent and universal if it can approximate any other model and the inverse of any other model, including itself, with precision on the order of 0.01 pixels.

6.2.1 The Polynomial Model

Unlike many distortion models, the polynomial model is not radial symmetric and the distortion in the x and y direction is modeled with different parameters and even different orders. Denoting by p and q the order of distortion for the x and y component respectively, the polynomial model has the form

$$\begin{aligned}
\bar{x}_d &= b_0\bar{x}_u^p + b_1\bar{x}_u^{p-1}\bar{y}_u + b_2\bar{x}_u^{p-2}\bar{y}_u^2 + \cdots + b_p\bar{y}_u^p \\
&\quad + b_{p+1}\bar{x}_u^{p-1} + b_{p+2}\bar{x}_u^{p-2}\bar{y}_u + \cdots + b_{2p}\bar{y}_u^{p-1} \\
&\quad + \cdots + b_{\frac{(p+1)(p+2)}{2}-3}\bar{x}_u + b_{\frac{(p+1)(p+2)}{2}-2}\bar{y}_u \\
&\quad + b_{\frac{(p+1)(p+2)}{2}-1} \\
\bar{y}_d &= c_0\bar{x}_u^q + c_1\bar{x}_u^{q-1}\bar{y}_u + c_2\bar{x}_u^{q-2}\bar{y}_u^2 + \cdots + c_q\bar{y}_u^q \\
&\quad + c_{q+1}\bar{x}_u^{q-1} + c_{q+2}\bar{x}_u^{q-2}\bar{y}_u + \cdots + c_{2q}\bar{y}_u^{q-1} \\
&\quad + \cdots + c_{\frac{(q+1)(q+2)}{2}-3}\bar{x}_u + c_{\frac{(q+1)(q+2)}{2}-2}\bar{y}_u \\
&\quad + c_{\frac{(q+1)(q+2)}{2}-1}.
\end{aligned} \tag{6.1}$$

The number of parameters for the x and y components is respectively $\frac{(p+1)(p+2)}{2}$ and $\frac{(q+1)(q+2)}{2}$. The model is called bicubic if $p = q = 3$. Note that the model being translation invariant, the choice of the distortion center (x_c, y_c) has no influence. By the analysis in Chapter 5, the polynomial model is self-consistent and more universal than other traditional models. So the polynomial model can also be used as a correction model by interchanging the role of distorted point coordinates and undistorted point coordinates in Eq. (6.1).

6.2.2 The Plumb-Line Method

To correct the distortion from a single image, only the distorted points are available in general. In such a case, some prior or implicit information is necessary to correct the distortion, for example the extended epipolar geometry between corresponding distorted points [10, 108], or the prior shape of some image features [162]. The plumb-line method is based on the fact that the 2D image of a 3D line remains straight if the camera is a pinhole camera (no lens distortion). The average distance from all edge points of corrected lines to their respective regression line is the most geometric error measurement.

6.3 Experimental Method

In this section, we detail how to integrate the polynomial model into the plumb-line method and try different strategies to minimize the distortion. The synthetic tests will show that the realistic distortion can be efficiently removed by using an appropriate minimization algorithm. These test have a ground truth which will actually permit to single out the right minimization strategy. In the real tests, the proposed method will be compared to other methods and will show its higher correction precision.

6.3.1 Synthetic Tests

Given a set of corrected points $(x_{u_i}, y_{u_i})_{i=1,\cdots,N}$ which are supposed to be on a line, we compute the linear regression line:

$$\alpha x_{u_i} + \beta y_{u_i} - \gamma = 0 \tag{6.2}$$

with $\tan 2\theta = -\frac{2(A_{xy} - A_x A_y)}{V_{xx} - V_{yy}}$, $\alpha = \sin\theta$, $\beta = \cos\theta$, $A_x = \frac{1}{N}\sum_{i=1}^{N} x_{u_i}$, $A_y = \frac{1}{N}\sum_{i=1}^{N} y_{u_i}$, $A_{xy} = \frac{1}{N}\sum_{i=1}^{N} x_{u_i} y_{u_i}$, $V_{xx} = \frac{1}{N}\sum_{i=1}^{N}(x_{u_i} - A_x)^2$, $V_{yy} = \frac{1}{N}\sum_{i=1}^{N}(y_{u_i} - A_y)^2$ and $\gamma = A_x \sin\theta + A_y \cos\theta$. The sum of squared distances from the points to this regression line is $\sum_{i=1}^{N} (\alpha x_{u_i} + \beta y_{u_i} - \gamma)^2$. By considering G groups of lines, the total sum of squared distance is

$$S = \sum_{g=1}^{G}\sum_{l=1}^{L_g}\sum_{i=1}^{N_{gl}} S_{gli}^2 = \sum_{g=1}^{G}\sum_{l=1}^{L_g}\sum_{i=1}^{N_{gl}} (\alpha_g x_{u_{gli}} + \beta_g y_{u_{gli}} - \gamma_{gl})^2 \tag{6.3}$$

with:
- L_g the number of lines in group g;
- N_{gl} the number of points of line l in group g;
- $N = N_{11} + \cdots + N_{1L_1} + \cdots + N_{G1} + \cdots + N_{GL_G}$ the total number of points;
- $(x_{d_{gli}}, y_{d_{gli}})$ the i-th distorted point on line l in group g;
- $(x_{u_{gli}}, y_{u_{gli}})$ the corresponding corrected point.

The root mean squared distance is

$$d = \sqrt{\frac{\sum_{g=1}^{G}\sum_{l=1}^{L_g}\sum_{i=1}^{N_{gl}} S_{gli}^2}{N}}. \tag{6.4}$$

The polynomial model will be chosen to correct the distorted lines. Recall that (x_c, y_c) can be fixed arbitrarily, thanks to the model translation invariance. For the sake of succinctness, the following discussion assumes a bicubic model with $p = q = 3$, but the method is completely general and will be applied in the experiments with higher degrees. Combining Eq. (6.1) and Eq. (6.3), the energy S becomes

$$\begin{aligned} S \;=\; & \sum_{g=1}^{G}\sum_{l=1}^{L_g}\sum_{i=1}^{N_{gl}} \Big(\alpha_g\big(b_0 \bar{x}_{d_{gli}}^3 + \cdots + b_9 + x_c\big) \\ & + \beta_g\big(c_0 \bar{x}_{d_{gli}}^3 + \cdots + c_9 + y_c\big) - \gamma_{gl}\Big)^2 \end{aligned} \tag{6.5}$$

To minimize the energy S in the parameters $b_0, b_1, \cdots, c_0, c_1, \cdots$ is a non-linear problem. To get an idea of the attainable precision we rendered first this problem linear by assuming that α_g, β_g are known. Differentiating S with respect to each parameter yields the linear system

$$\mathbf{A}\mathbf{x} = \mathbf{0} \tag{6.6}$$

with

$$\begin{aligned} \mathbf{x} \;=\; & (\gamma_{11}, \cdots, \gamma_{1L_1}, \cdots, \gamma_{G1}, \cdots, \gamma_{GL_G}, \\ & b_0, \cdots, b_9, c_0, \cdots, c_9)^T. \end{aligned}$$

$\mathbf{A}$ is composed of 3 sub-matrices

$$\mathbf{A} = \begin{bmatrix} \mathbf{A_\gamma} \\ \mathbf{A_b} \\ \mathbf{A_c} \end{bmatrix}. \tag{6.7}$$

When there is only one group of lines, the rows of $\mathbf{A_b}$ are proportional to the corresponding rows $\mathbf{A_c}$. So for testing, we always use several groups of lines to avoid this situation. Yet the coefficient matrix is still singular since the last row of $\mathbf{A_b}$ and the last row of $\mathbf{A_c}$ are a linear combination of the rows of $\mathbf{A}_\gamma$. This can be solved by fixing b_9 and c_9 to be 0. In that case, the coefficient matrix is non-singular and has only one solution. But this solution is trivial because S becomes 0 by setting all the coefficients $b_1, \cdots, b_9, c_1, \cdots, c_9$ to 0 and $\gamma_{gl} = x_c\alpha_g + y_c\beta_g$. To avoid this trivial solution minimal constraints must be added; we fix $b_7 = 1$ and $c_8 = 1$. This amounts to introducing a scale in the solution. The values fixed for b_9 and c_9 induce a translation to the solution. The minimized S can be changed by the introduced scale. But this change is consistent if $x_c, y_c, b_9, c_9, b_7, c_8$ are fixed.

In the tests, we used 8 groups of lines with orientations 10°, 20°, 30°, 40°, 50°, 60°, 70°, 80° to estimate the correction parameters by minimizing S. Once the parameters are estimated, another independent group of lines with orientation 55° is used for the verification. The ideal lines are distributed in an image with size 1761×1174. The sampling step of each line is 30 pixels and the number of samples on each line is never less than 15. The distance between two adjacent lines is 30 pixels. The ideal lines are distorted by a radial distortion plus a tangential distortion[4]. The correction result is recapitulated in Table 6.1. A precision of the order of 10^{-2} pixels can be achieved by increasing the order of polynomial model. The lines in this table can be grouped in pairs of successive even and odd order having almost the same precision. We have no explanation for this phenomenon. Fig. 6.1 shows ideal lines, distorted lines and corrected lines of the test group with orientation 55°. Remark that the corrected lines are close to the ideal lines but they do not completely superimpose due to the introduced translation and scale in the correction process. In fact, we could apply any rotation and translation on the corrected lines to obtain other corrected lines, with the same residual error. However, applying a homography on the corrected lines could lead to a different error, depending on the local scale introduced by the homography.

In practice the orientation of the lines is unknown. The minimization of the energy in Eq. (6.3) is therefore a non-linear problem. As we have seen, α_g, β_g and γ_{gl} of the corrected lines can be parametrized by $b_0, b_1, \cdots, c_0, c_1, \cdots$. So the only unknown parameters in Eq. (6.3) are the parameters of the polynomial model (the distortion center is fixed at the center of image). Coefficients of order-0 terms and order-1 terms of the polynomial model are set such that the correction is close to identity at the center of image. Namely, for the

[4] The distortion is added according to the equation:

$$\begin{aligned}
\bar{x}_d &= \bar{x}_u \left(k_0 + k_1 r_u + k_2 r_u^2 + \cdots\right) \\
&+ \left[p_1 \left(r_u^2 + 2\bar{x}_u^2\right) + 2p_2 \bar{x}_u \bar{y}_u\right] \left(1 + p_3 r_u^2\right) + s_1 r_u^2 \\
\bar{y}_d &= \bar{y}_u \left(k_0 + k_1 r_u + k_2 r_u^2 + \cdots\right) \\
&+ \left[p_2 \left(r_u^2 + 2\bar{y}_u^2\right) + 2p_1 \bar{x}_u \bar{y}_u\right] \left(1 + p_3 r_u^2\right) + s_2 r_u^2
\end{aligned}$$

with $k_0, k_1, \cdots$ the radial distortion coefficients, p_1, p_2, p_3 the decentering distortion coefficients, s_1, s_2 thin prism distortion coefficients. In our synthetic test, $k_0 = 1.0$, $k_1 = 1.0\text{e}{-4}$, $k_2 = -2.0\text{e}{-7}$, $k_3 = 4.0\text{e}{-10}$, $k_3 = -6.0\text{e}{-14}$, $p_1 = 4.0\text{e}{-6}, p_2 = -2.0\text{e}{-6}, p_3 = 0, s_1 = 3.0\text{e}{-6}, s_2 = 1.0\text{e}{-6}$.

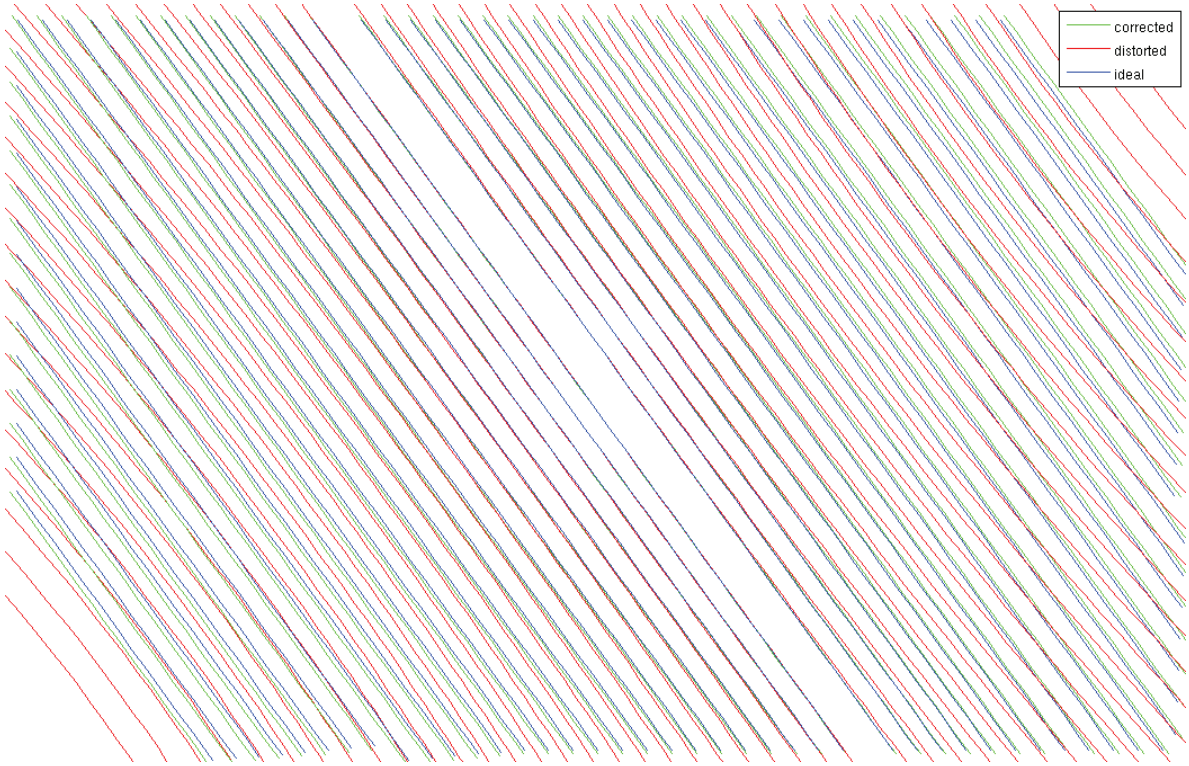


Figure 6.1: The corrected, distorted and ideal lines for the test group of lines with orientation 55° by using the linear method. The green lines are lines corrected by using the estimated parameters; the red lines are the distorted lines, and the blue lines are the ideal lines.

order	d (in pixels) in Eq (6.4)	
$p = q$	linear estimation	indep. measure
3	0.6935	0.6239
4	0.6096	0.5312
5	0.2439	0.2093
6	0.2419	0.2064
7	0.1050	0.0879
8	0.1031	0.0870
9	0.0521	0.0512
10	0.0515	0.0509
11	0.0477	0.0449
12	0.0474	0.0444

Table 6.1: Line correction with known orientation. The ideal lines are distorted by radial distortion plus tangential distortion (see footnote 4). The energy in Eq. (6.5) is minimized by linear method. The root mean square distance in Eq (6.4) is used as distortion measurement. Column 1 is the order of the polynomial model. Column 2 is the measurement for the lines with orientation from 10° to 80°. Column 3 is the measurement for the independent group of lines with orientation 55°.

x-component of the polynomial model, the coefficient of the term $(x_d - x_c)$ is set to 1, the coefficient of the term of $(y_d - y_c)$ is set to 0, and the coefficient of order-0 term is set to 0; for the y-component of the polynomial model, the coefficient of the term $(x_d - x_c)$ is set to 0, and the coefficient of the term of $(y_d - y_c)$ is set to 1, and the coefficient of order-0 term is set to 0. Four different strategies will be tested on the synthetic data to ensure that this non-convex minimization process reaches or gets very close to the solution. In case the strategy uses a non-linear minimization, the Levenberg-Marquardt (LM) algorithm (see Appendix A.3) is applied with Jacobian matrix explicitly computed.

The first strategy is to simply apply the LM algorithm (see Table 6.2). But it is often stuck at local minima. It works well only if the initialization is already close to the global minimum.

The second strategy is an iterative linear minimization. The linear method requires the orientations of the lines. We first compute the linear regression lines from the distorted points. The orientations of the linear regression lines allow us to apply the linear method to estimate the parameters of the polynomial model. Once the parameters of the polynomial model are estimated, we correct the distorted points and compute again the orientations of the linear regression lines from the corrected points. The new orientations allow us to apply again the linear method to estimate the parameters of polynomial model. This procedures can be iterated until the average error does not decrease significantly (for example, 0.01 pixel in the experiments).

The third strategy is to fix the orientation obtained by LM and do the iterative linear minimization to improve the result (see Table 6.2).

The fourth strategy is the incremental LM algorithm followed by the iterative linear minimization. The incremental LM algorithm estimates the parameters of a high order polynomial

model starting from the results of a lower order model (see Table 6.4). For example, to estimate the parameters of a order-11 polynomial, we begin to estimate the parameters of an order-3 polynomial. Then the estimated order-3 parameters are used as the initialization for order-4 polynomial, and so on. This procedure is iterated until order-11. Finally a step of iterative linear minimization is added to improve the precision.

Minimizing the energy in Eq. (6.3) when the orientation of lines is unknown is a difficult non-linear optimization problem, particularly when the order of polynomial model is high. Comparing the results of the different mentioned optimization strategies with the best attainable precisions in Table 6.1, we conclude that the incremental LM algorithm with an iterative linear method gives the best accuracy.

order $p = q$	d (in pixels) in Eq (6.4)			
	estimation		indep. measure	
	LM	iter. linear	LM	iter. linear
3	0.7013	0.6489	0.5988	0.5596
4	0.6419	0.6109	0.5491	0.5087
5	0.2937	0.2852	0.2522	0.2507
6	0.2698	0.2624	0.2280	0.2222
7	0.2609	0.1509	0.1956	0.1452
8	0.3472	0.1611	0.2584	0.1574
9	0.2522	0.1503	0.2300	0.1561
10	0.9814	0.1863	0.5841	0.1763
11	0.5797	0.1593	0.4178	0.1454

Table 6.2: Line correction with unknown orientation by Levenberg-Marquardt (LM) algorithm and iterative linear method (strategy 1 and 3). The ideal lines are distorted by radial distortion plus tangential distortion (see footnote 4). The energy in Eq. (6.5) is minimized by LM plus a step of iterative linear minimization. The root mean squared distance in Eq (6.4) is computed as measurement. Column 1 is the order of the polynomial model. Column 2 is the measurement of LM (strategy 1) for the lines with orientation from 10° to 80°. Column 3 is the measurement of LM plus a linear minimization (strategy 3) for the lines with orientation from 10° to 80°. Column 4 and 5 corresponds to column 2 and 3 respectively, by using the the independent group of lines with orientation 55° which is not used for optimization. Notice that the error first decreases with the order and then increases again, which can only be attributed to the fact that the minimization process fails and stays away from a global minimum.

6.3.2 Experiments on Real Data

The real experiments were made with a Canon EOS 30D reflex camera and an EFS $18-55$mm lens. The minimal focal length (18mm) was chosen to produce a fairly large distortion. The RAW images were demosaicked by summing up the four pixels of each 2×2 Bayer cell, obtaining a half-size image. The "calibration harp" was built by stretching tightly *sewing*

order	d (in pixels) in Eq. (6.4)	
$p = q$	iter. linear	indep. measure
3	0.6315	0.5083
4	0.6136	0.4849
5	0.2601	0.2374
6	0.2594	0.2371
7	0.1469	0.1368
8	0.1455	0.1360
9	0.1105	0.1096
10	0.1106	0.1098
11	0.1156	0.1116

Table 6.3: Line correction with unknown orientation by iterative linear method (strategy 2). The ideal lines are distorted by radial distortion plus tangential distortion (see footnote 4). The energy in Eq. (6.5) is minimized by iterative linear minimization (strategy 2). The root mean squared distance in Eq. (6.4) is computed as measurement. Column 1 is the order of the polynomial model. Column 2 is the measurement of iterative linear minimization for the lines with orientation from 10° to 80°. Column 3 corresponds to column 2, by using the independent group of lines with orientation 55° which is not used for optimization.

strings on a wooden frame. The tension of the strings can actually be verified by pinching them and listening to the sound. Fig. 6.2 shows the distorted images of the harp with different orientations against the sky as background. The distortion is visible near the border of the image. All the photos, including the photos of the pattern for Lavest *et al.* method, were taken at the same time by the camera with fixed configuration (focal length, focus, etc.).

To correct the distortion, the edge points of the distorted lines need to be extracted in high precision. This is performed by Devernay's edge detector [52] and LSD algorithm [174], followed by a 1D Gaussian convolution and a sub-sampling, as presented in Chapter 3.

Once the edge points associated to the distorted line segments have been extracted at sub-pixel precision, they can be directly integrated into the energy term in Eq. (6.5). According to the performance of the strategies examined in the synthetic tests, only strategy 4 was validated. It implements the incremental LM followed by iterative linear minimization. All the distorted lines in Fig. 6.2 are used in the minimization with a polynomial model of order 11. An independent distorted image, which is not used in the minimization, is used for verification. The correction result is recapitulated in Fig. 6.3.

Several different methods, like the non-parametric method in Chapter 4 [80], the Lavest *et al.* calibration method [95] and an iterative linear plumb-line method [4], were also tried (see the correction precision in Table 6.5).

The non-parametric method in Chapter 4 [80] estimates the distortion as the diffeomorphism (up to a homography) mapping the original digital pattern to a photograph of it by triangulating and interpolating dense correspondences (see the result in Fig. 6.4).

The Lavest *et al.* calibration method [95] is similar to global camera calibration methods, but it takes into account the pattern is non-flatness and therefore estimates also the 3D

order $p = q$	d (in pixels) in Eq (6.4)			
	estimation		indep. measure	
	LM	iter. linear	LM	iter. linear
3	0.7042	0.6514	0.6018	0.5618
4	0.5995	0.5794	0.5128	0.4735
5	0.2571	0.2510	0.2219	0.2167
6	0.2463	0.2419	0.2093	0.2053
7	0.2126	0.1091	0.1740	0.0925
8	0.2067	0.1062	0.1661	0.0909
9	0.1953	0.0599	0.1569	0.0588
10	0.1823	0.0576	0.1425	0.0571
11	0.1805	0.0546	0.1419	0.0524

Table 6.4: Line correction with unknown orientation by incremental Levenberg-Marquardt (LM) algorithm and iterative linear method (strategy 4). The ideal lines are distorted by radial distortion plus tangential distortion (see footnote 4). The energy in Eq. (6.5) is minimized by incremental LM plus a step of iterative linear minimization. The root mean squared distance in Eq (6.4) is computed as measurement. Column 1 is the order of the polynomial model. Column 2 is the measurement of incremental LM for the lines with orientation from 10° to 80°. Column 3 is the measurement of incremental LM plus a linear minimization (strategy 4) for the lines with orientation from 10° to 80°. Column 4 and 5 correspond to column 2 and 3 respectively, by using the the independent group of lines with orientation 55° which is not used for optimization. The conclusion is that only strategy 4 reaches a high accuracy, growing with the order of the polynomials.

position of the feature points on the pattern (see result in Fig. 6.5). Thus it is a complete *bundle adjustment* method [171], applied to a well photographed pattern.

Alvarez's iterative linear plumb-line method [4, 3] uses a pure radial distortion model and minimizes the variance of distance from the corrected points to their regression line by iterative linear method (see result in Fig. 6.7).

Like in the real experiments in Chapter 5, we can also use the polynomial model to fit the "outliers"-free matchings between the digital pattern and its photo, instead of the triangulation and affine interpolation used in Chapter 4 [80] (see result in Fig. 6.6).

The correction precision (also called "straightness") is computed as the RMS distance from the edge points on the corrected line to their regression line. It is known that an arbitrary homography does not change the "straightness"of a perfectly straight line. But in the experiments, the corrected lines are never perfectly straight. So it is possible that the "straightness" is enlarged or reduced by the unknown homography. To reduce the influence introduced by the homography, The coefficients of order-0 and order-1 terms of the polynomial model are set as section 6.3.1. And this is also the case for the model used in Alvarez's method and Lavest *et al.* method. For the non-parametric method based on the textured pattern, the photo is taken such that the whole camera captor is covered by the whole pattern. This gives a weak homography which does not introduce a big scale to enlarge or reduce the "straightness".

The correction precision is recapitulated in Table 6.5 for all considered methods. It seems that the two plumb-line based methods (Alvarez *et al.* method and the proposed one) give a higher precision than the other methods. This is not surprising because on the one hand both methods explicitly minimize the straightness error of the corrected lines; on the other hand, neither methods suffer from the error compensation in global calibration methods or the non-flatness of the pattern in non-parametric method. The disadvantage of the Alvarez *et al.* method is that it uses a simple radial distortion model with distortion center fixed at the center of the image to get an iterative linear solution. But this model is not enough general to explain the real distortion. This explains why the Alvarez *et al.* method corrects some lines less precisely than others.

For the Lavest *et al.* method, the minimized re-projection error is about 0.02 pixels while the corrected lines do not have that precision (see Fig. 6.5). This can only be explained by an error compensation between camera internal and external parameters.

For the non-parametric method, a global tendency in the straightness error of the corrected lines can be observed (Fig. 6.4). This was in fact due to the unavoidable drawback of this method: there is never a guarantee that the pattern is completely flat. The non-flatness of the pattern introduces a bias in the estimated distortion field, which causes the observable global distortion in the plotted curves in Fig. 6.4. Remark that the very similar global tendency can be observed when the distortion is approximated by a 11-order polynomial instead of triangulation and affine interpolation (see Fig. 6.6).

To eliminate this source of error, the solution is either to construct a very flat pattern, or to recover the 3D shape of a non-flat pattern. But neither is very feasible in practice. In contrast, to appropriately use a plumb-line method, we need a pattern containing very straight lines, and this is far easier in practice. As shown in Fig. 6.3, the distortion correction is so accurate that no global tendency is visible in the corrected curves. The root mean square (RMS) distance of each line is also significantly smaller than for the non-parametric method (Table 6.5). It is particularly striking in Fig. 6.3 that the superimposed curves of the left

and right side of each string are fairly uncorrelated, meaning that no deterministic distortion is left. The erratic oscillation of very small amplitude can be attributed to any cause, from the lack of uniformity of the harp background causing a shift in the edge detection, to the inhomogeneous blur in the image itself or to the quality of the strings. But it cannot be due to a residual mismatch of the polynomial model itself, because otherwise the curves on both sides of each string would show parallel distortions. This confirms *a posteriori* the reliability of the polynomial model.

The estimated distortion fields of the above methods are also different. For the non-parametric method, the distortion field consists of the vectors pointing from a certain point in the undistorted image to its correspondence in the distorted image. So an undistorted image can be directly obtained given a distorted image. Remark that the non-parametric method estimates the distortion field up to an unknown homography. So the distortion field is not radial symmetric at all (see the distortion field in Fig. 6.4). For all the other methods, the distortion parameters are estimated from the distorted image to the undistorted image. But this correction model does not send the points in the distorted image on integer points of the undistorted image domain. So the resulted undistorted image can contain holes. This problem can be solved by either reversing the model, or by computing the corresponding distorted point in the distorted image for the integer-position point in the corrected image by a non-linear minimization, followed by an image interpolation. The radial model with known distortion center (used in Alvarez *et al.* method) is invertible. But this inversion does not give an explicit formula of the inverse model. In contrast, the inverse polynomial model (distortion model) can be easily computed explicitly. Given a polynomial model (correction model), an arbitrary set of distorted points and corrected points can be generated. From these points, the coefficients of the inverse model in Eq. (6.1) can be estimated by a linear method:

$$
\begin{bmatrix}
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\
\bar{x}_{u_i}^p & \bar{x}_{u_i}^{p-1}\bar{y}_{u_i} & \bar{x}_{u_i}^{p-2}\bar{y}_{u_i}^2 & \cdots & 1 & 0 & 0 & 0 & \cdots & 0 \\
0 & 0 & 0 & \cdots & 0 & \bar{x}_{u_i}^q & \bar{x}_{u_i}^{q-1}\bar{y}_{u_i} & \bar{x}_{u_i}^{q-2}\bar{y}_{u_i}^2 & \cdots & 1 \\
\bar{x}_{u_{i+1}}^p & \bar{x}_{u_{i+1}}^{p-1}\bar{y}_{u_{i+1}} & \bar{x}_{u_{i+1}}^{p-2}\bar{y}_{u_{i+1}}^2 & \cdots & 1 & 0 & 0 & 0 & \cdots & 0 \\
0 & 0 & 0 & \cdots & 0 & \bar{x}_{u_{i+1}}^q & \bar{x}_{u_{i+1}}^{q-1}\bar{y}_{u_{i+1}} & \bar{x}_{u_{i+1}}^{q-2}\bar{y}_{u_{i+1}}^2 & \cdots & 1 \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots
\end{bmatrix}
$$

$$
\begin{pmatrix}
b_0 \\
\vdots \\
b_{\frac{(p+1)(p+2)}{2}-1} \\
c_0 \\
\vdots \\
c_{\frac{(q+1)(q+2)}{2}-1}
\end{pmatrix}
=
\begin{pmatrix}
\vdots \\
\bar{x}_{d_i} \\
\bar{y}_{d_i} \\
\bar{x}_{d_{i+1}} \\
\bar{y}_{d_{i+1}} \\
\vdots
\end{pmatrix}
$$

Each distorted/undistorted pair gives two equations. So at least $\frac{(p+1)(p+2)}{4} + \frac{(q+1)(q+2)}{4}$ pairs of correspondences are required to estimate the parameters
$b_0, \cdots, b_{\frac{(p+1)(p+2)}{2}-1}, c_0, \cdots, c_{\frac{(q+1)(q+2)}{2}-1}$.

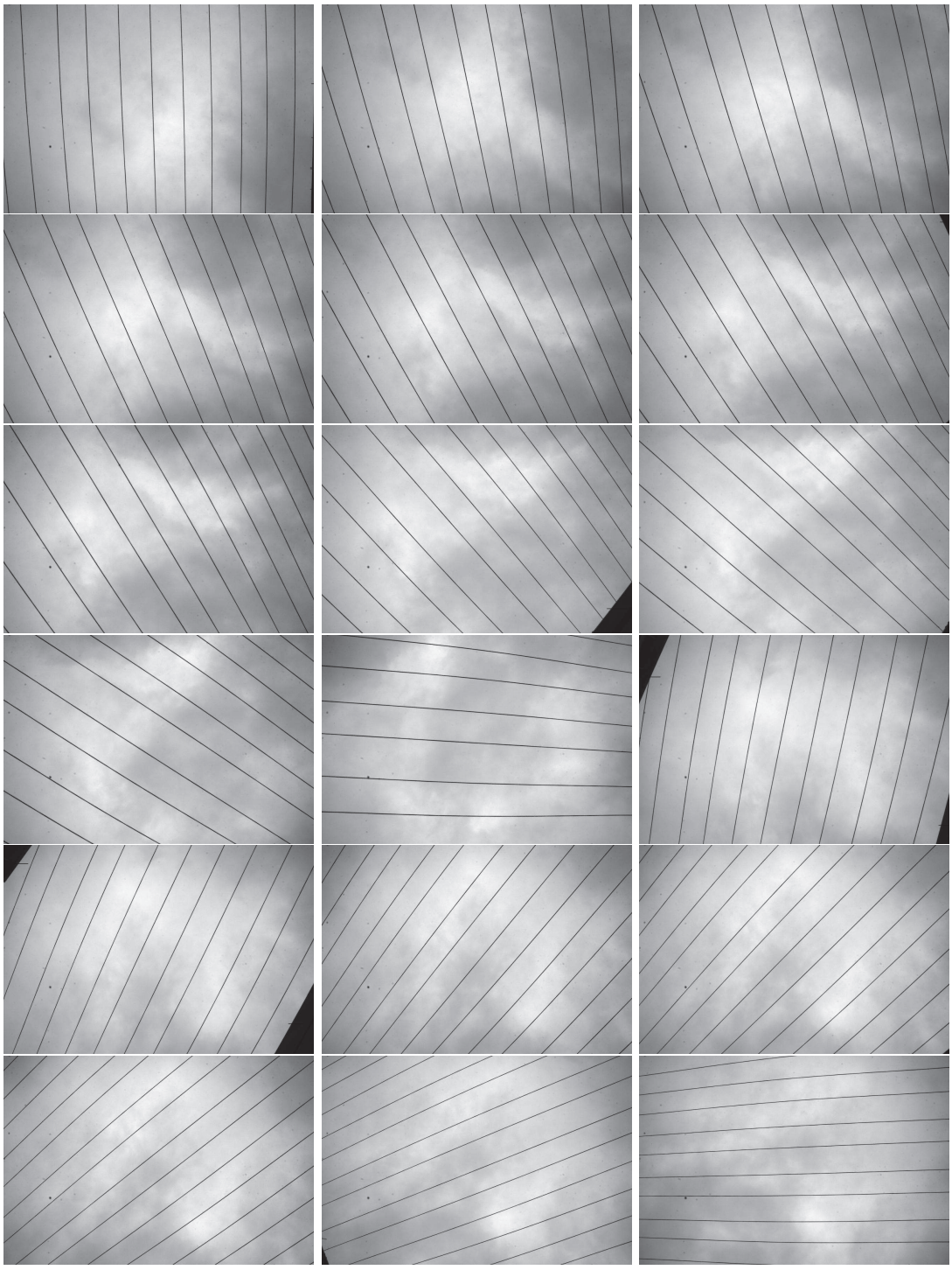


Figure 6.2: Distorted sewing strings taken by the camera by hand with different orientations. The background is the sky.

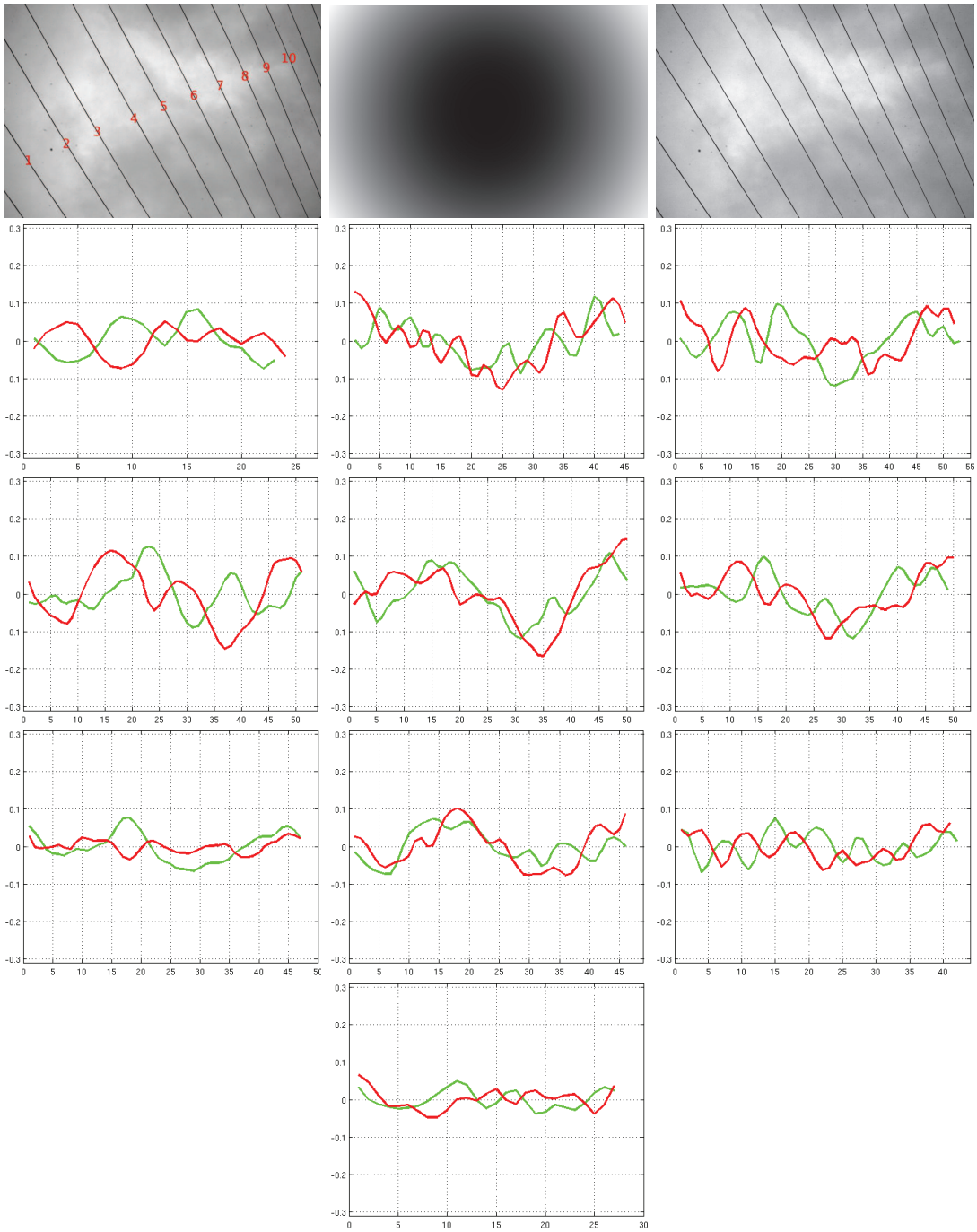


Figure 6.3: Correction performance of the proposed plumb-line based method with the calibration harp using sewing strings. On the first row, from left to right: the independent distorted image, the distortion field and the corrected image. From the second row to the last row, from left to right: the distance in pixels from the edge points to their regression line on line 1 to 10 in the distorted image on top-left, after correction. Note that each figure contains two curves because there are two sides for each string. Both curves form a braid pattern because these sewing strings are actually braided. This braid pattern is not present for the fishing and tennis racket strings. The x-axis is the index of edge points. The range of y-axis is from -0.3 pixels to 0.3 pixels.

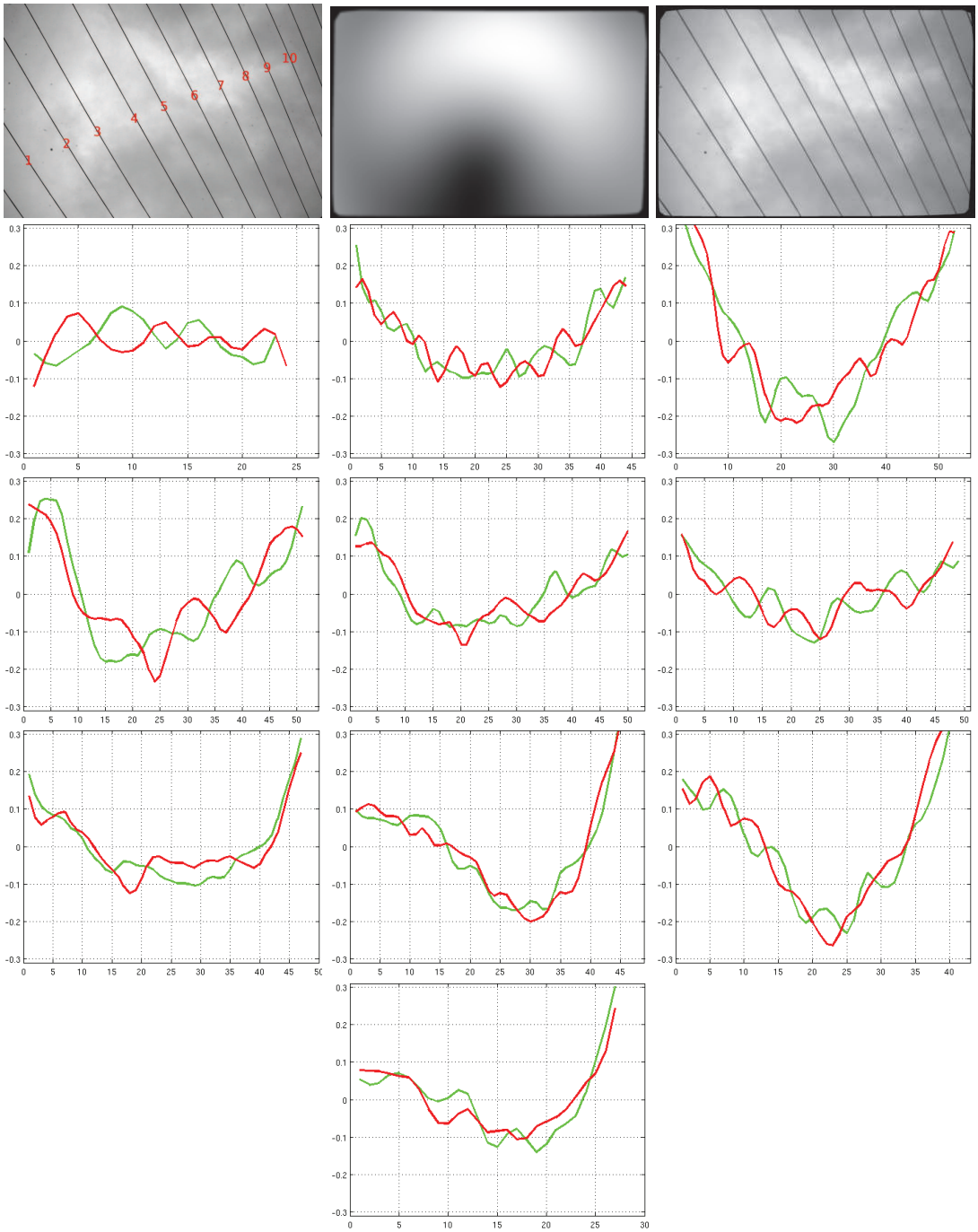


Figure 6.4: Correction performance of the non-parametric pattern-based method [80]. On the first row from left to right: the independent distorted image, the distortion field and the corrected image. From the second row to the last row, from left to right: the distance in pixels from the edge points to their regression line on lines 1 to 10 in the distorted image on top-left, after correction. Note that each figure contains two curves because there are two sides for one string. The x-axis is the index of edge points. The range of y-axis is from -0.3 pixels to 0.3 pixels.

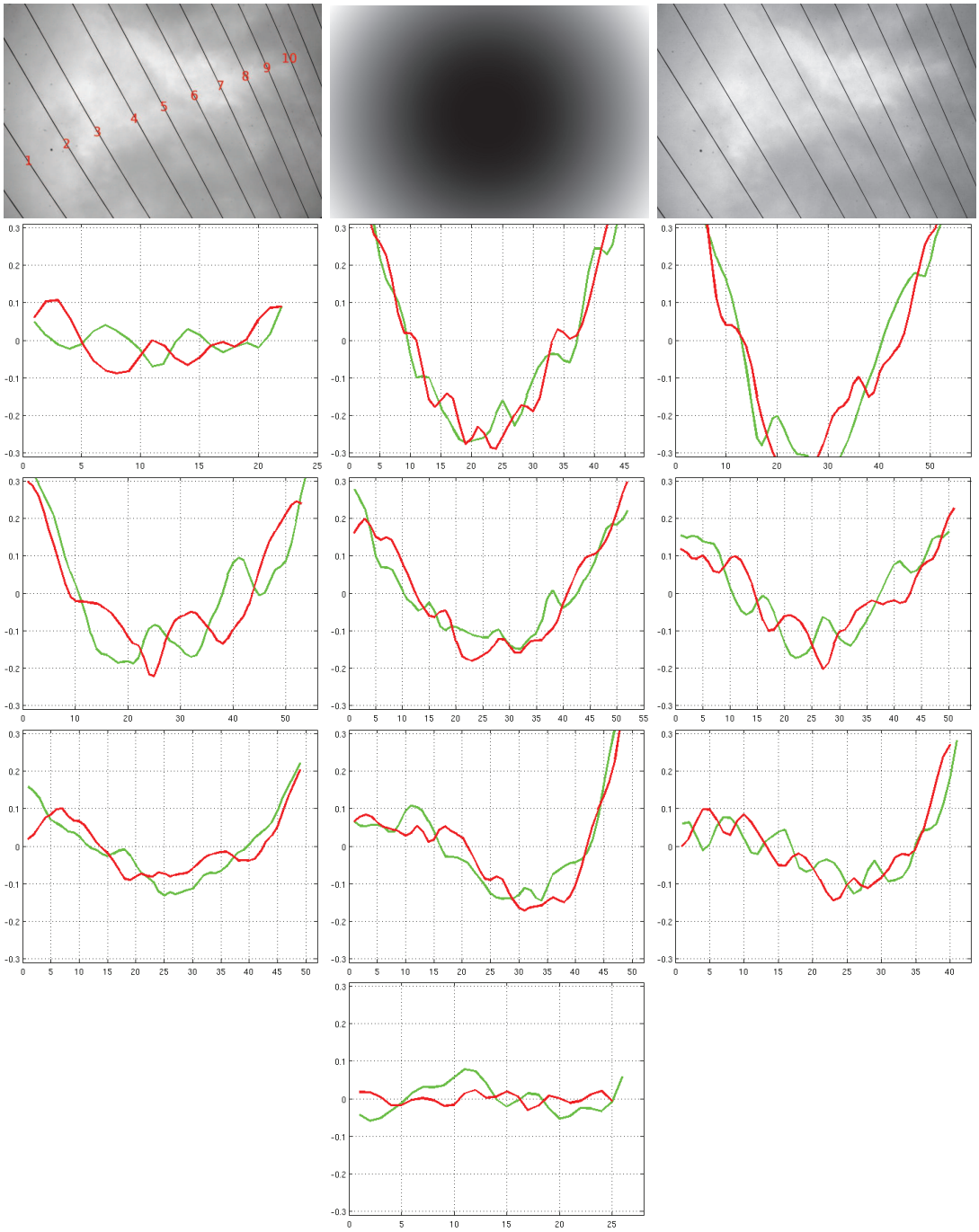


Figure 6.5: Correction performance of the Lavest *et al.* method [95]. On the first row, from left to right: the independent distorted image, the distortion field and the corrected image. From the second row to the last row, from left to right: the distance in pixels from the edge points to their regression line on line 1 to 10 in the distorted image on top-left, after correction. Note that each figure contains two curves because there are two sides for one line. The x-axis is the index of edge points. The range of y-axis is from -0.3 pixels to 0.3 pixels.

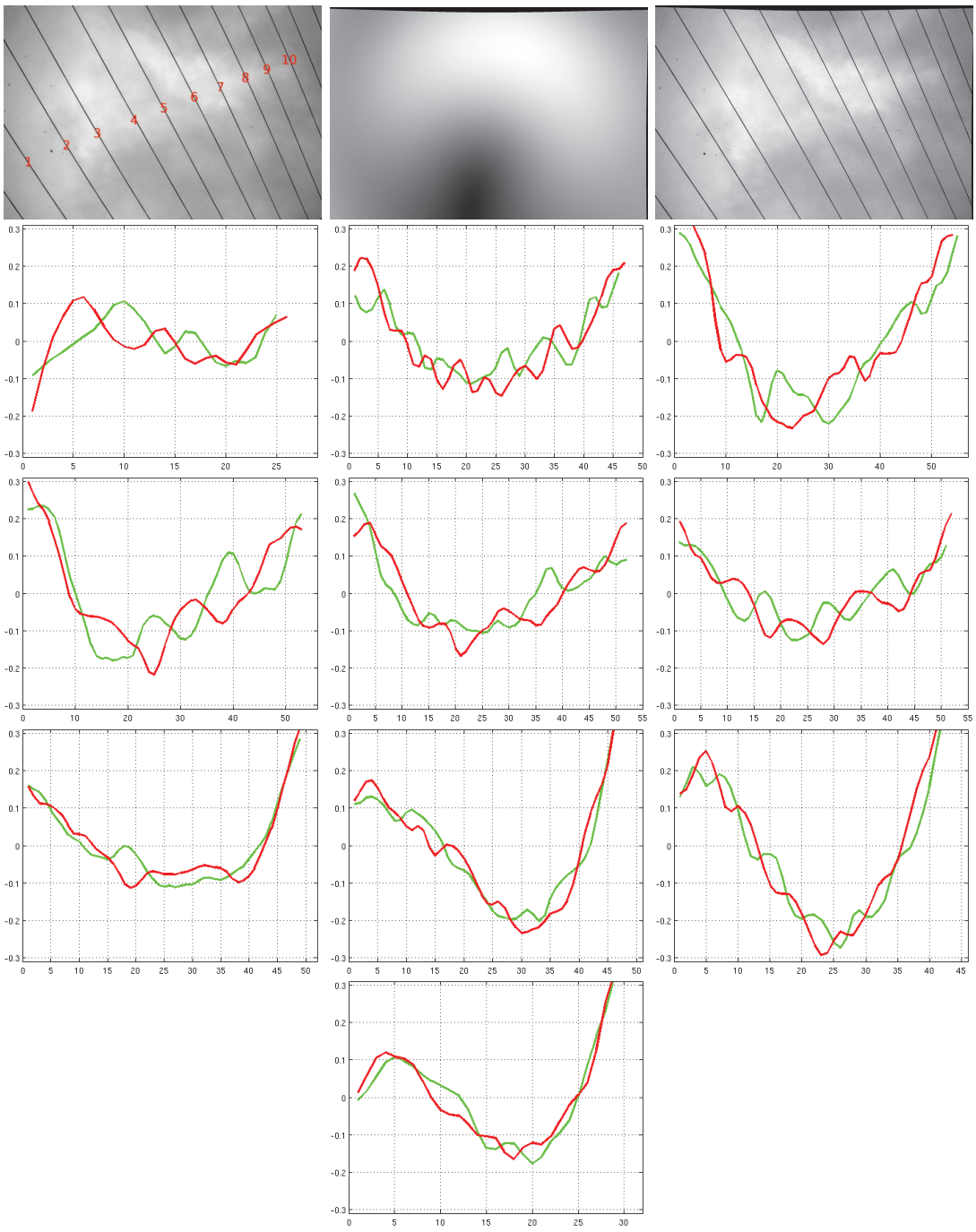


Figure 6.6: Correction performance of a textured flat pattern with the polynomial model. On the first row, from left to right: the independent distorted image, the distortion field and the corrected image. From the second row to the last row, from left to right: the distance in pixels from the edge points to their regression line on lines 1 to 10 in the distorted image on top-left, after correction. Note that each figure contains two curves because there are two sides for one string. The x-axis is the index of edge points. The range of y-axis is from -0.3 pixels to 0.3 pixels.

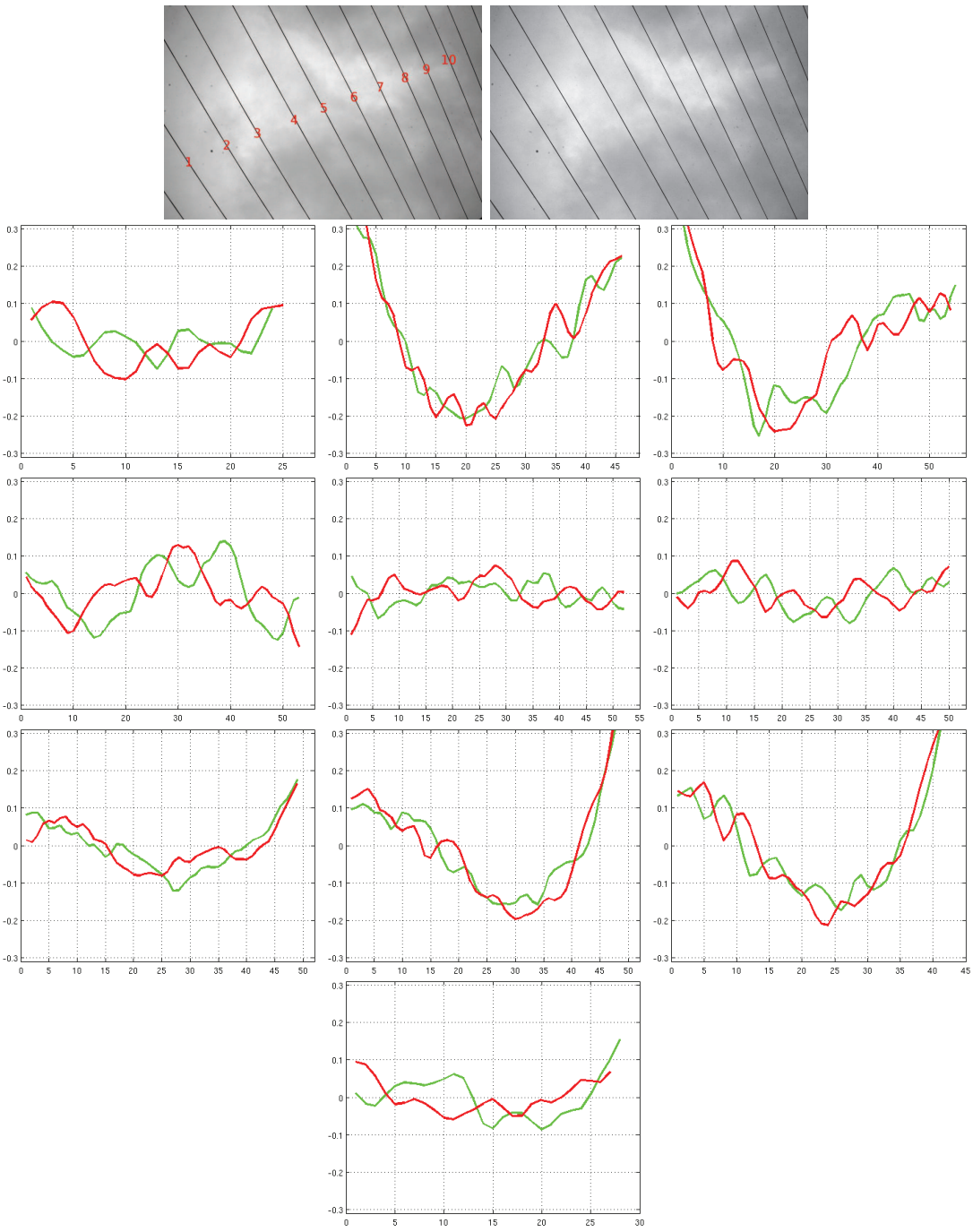


Figure 6.7: Correction performance of the Alvarez *et al.* method. On the first row, from left to right: the independent distorted image and the corrected image. From the second row to the last row, from left to right: the distance in pixels from the edge points to their regression line on lines 1 to 10 in the distorted image on top-left, after correction. Note that each figure contains two curves because there are two sides for one string. The x-axis is the index of edge points. The range of y-axis is from -0.3 pixels to 0.3 pixels.

line No.	RMSE (in pixels)				
	polynomial model	non-parametric	Lavest method	polynomial textured pattern	Alvarez method
1	0.046/0.036	0.048/0.042	0.035/0.062	0.055/0.065	0.038/0.069
2	0.050/0.068	0.088/0.082	0.217/0.218	0.081/0.112	0.157/0.171
3	0.057/0.054	0.166/0.168	0.267/0.270	0.152/0.174	0.147/0.152
4	0.051/0.073	0.135/0.126	0.156/0.139	0.129/0.128	0.076/0.060
5	0.061/0.076	0.082/0.080	0.118/0.137	0.092/0.103	0.029/0.035
6	0.052/0.056	0.069/0.062	0.108/0.099	0.075/0.088	0.041/0.037
7	0.039/0.017	0.095/0.080	0.090/0.072	0.100/0.102	0.067/0.058
8	0.042/0.054	0.133/0.143	0.117/0.127	0.163/0.180	0.129/0.152
9	0.035/0.036	0.154/0.162	0.080/0.095	0.197/0.204	0.131/0.146
10	0.010/0.008	0.040/0.014	0.122/0.120	0.058/0.043	0.058/0.043

Table 6.5: RMS distance from the edge points of the corrected lines to their corresponding regression line. The proposed method is compared to the non-parametric pattern-based method [80], the Lavest *et al.* method [95], the textured pattern polynomial fitting method, and the Alvarez method [4, 3]. Note that each cell in the table contains two values because there are two sides for each string. The lines are numbered in the top-left image in Fig. 6.3.

6.4 Current Limitations and Potential Improvement

As explained before, the main motivation to build a "calibration harp" for distortion correction was to circumvent the difficulty in obtaining a very flat pattern with flatness error of the order of 10 micron for a 40 centimeters width. Even though it is easier to build a "calibration harp" with tightly stretched strings, the quality of the strings plays an important role. In the previous experiments, ordinary sewing strings were used to build a first "calibration harp". But the sewing strings are not very smooth and their thickness oscillates in a braid pattern along the strings, due to their twisted structure (see Fig. 6.9a). This oscillation being local does not necessarily influence the global distortion correction precision. However, it alters the evaluation of the attained precision. We tried alternative choices such as tennis racket strings (Fig. 6.9b) and opaque fishing strings (Fig. 6.9c). Among the three types of strings, the tennis racket string and the opaque fishing string are apparently more smooth than the sewing string and have a more uniform thickness. But, as we shall see, tennis racket strings are rigid and would require an extreme tension to become straight. The opaque fishing strings are both smooth and flexible and give the best result once tightly stretched on a wooden framework (see Fig. 6.8).

To ensure the extraction precision of the edge points from the string images, an uniform background with contrast to the string color must be preferred. The first idea was to use an uniform wall as background. However, the projected shadows of the strings on the wall are a nuisance which can cause an edge detector failure (see Fig. 6.8a and 6.8c). The uniform wall should therefore be far away from the harp, which is not easy to realize. Next to a uniform wall, the sky is the only distant and uniform background that comes into mind. Yet, in the experiments, we found that it was difficult to take photos of the harp against the sky. When

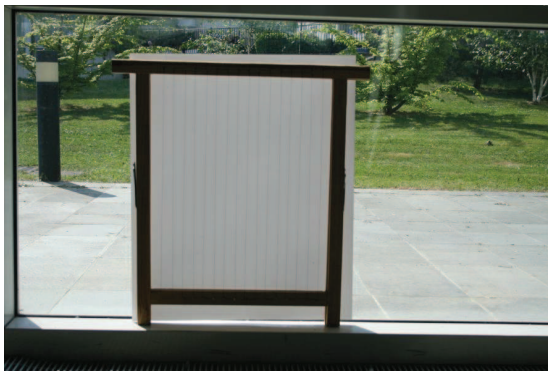
the angle of view of the camera is large, it is difficult to photograph only the sky and to avoid the interference of buildings, trees, etc. in the photos. In addition, even if at first sight the sky looks uniform, it turns out to be often inhomogeneous: See Fig. 6.2 or Fig. 6.10a. The final solution was to fix a translucent paper on the back of the harp and to use back lighting (see Fig. 6.8b and 6.8d for the harp with the translucent paper). This setup allows us to take photos anywhere, provided the back of the harp is sufficiently lit.

A Canon EOS 30D reflex camera was installed on a tripod with 10 seconds timer to avoid hand shakes and motion blur. Photos of different orientations were taken by rotating the camera on the tripod. The camera is always parallel to the harp and has the same distance to the harp (see Fig. 6.12 for photos of different orientations). Compared to the photos of sewing strings taken by hand against the sky (Fig. 6.10a), the photos of opaque fishing strings with a translucent paper (Fig. 6.10b) have a more uniform background. In addition, the images taken by hand (Fig. 6.10a) suffer from inhomogeneous blur or variation of strings thickness caused by the inconstant distance from camera to the harp or the hand motion, while the images taken by tripod have better quality.

The improvement of the string quality and of the photographic quality leads to a better accuracy evaluation. As we have seen before, even though the plumb-line method with a polynomial model can correct the distortion up to about 0.05 pixels (Fig. 6.3) and no big global tendency can be observed, the maximal amplitude of oscillation can still be larger than 0.05 pixels. For the second line in Fig. 6.3, a small global tendency is observed. But the amplitude of this small global tendency is of the same order of the amplitude of the oscillation. So the minimization algorithm could not tell apart the two types of error. So it is better to reduce this oscillation so that the minimization algorithm can concentrate on the global tendency minimization. The observed oscillation is not due to any lens distortion. It is in fact related to the quality of the strings. Indeed, the observed oscillation inherits the high frequency of the distorted lines, while lens distortion alters only the low frequency of the distorted lines. In Fig. 6.11, the high frequency of the distorted sewing string, distorted tennis racket string and distorted opaque fishing string are compared to the straightness error of their corresponding corrected strings. The almost superimposing high frequency oscillation confirms that the high frequency of the distorted strings is not changed by the lens distortion correction. Among the three types of strings, the opaque fishing string shows the smallest such oscillation. The larger oscillation of the sewing string is due to a variation of the thickness related to its twisted structure, while the tennis racket string is simply too rigid to be stretched, even if this is not apparent in Fig. 6.9b).

The comparison of the correction accuracy for a opaque fishing strings harp with a sewing strings harp is instructive. In Fig. 6.13, the straightness error is shown in the same form as before. Compared to Fig. 6.3, it is evident that we have a significantly more precise correction, and that the residual oscillation is reduced. The correction precision can be again verified by doing the same experiments with the largest focal length (55 mm) and the camera-object distance about 100 cm. The result is shown in Fig. 6.14. The average RMS distance in Table 6.6 is about 0.02, evidently better than the first column in Table 6.5.

(a) The harp with an uniform opaque object as background

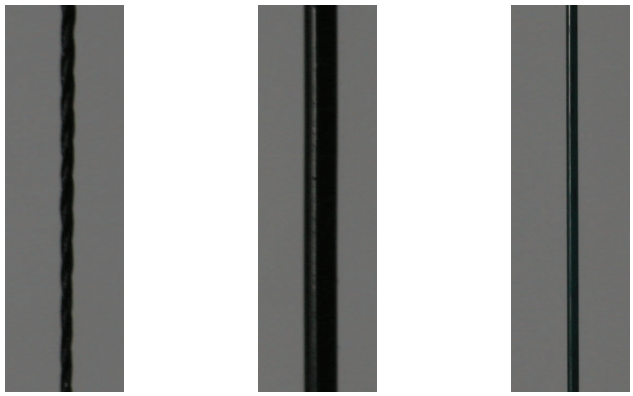


(b) The harp with a translucent paper as background

(c) A close-up of the harp with an uniform opaque object as background

(d) A close-up of the harp with a translucent paper as background

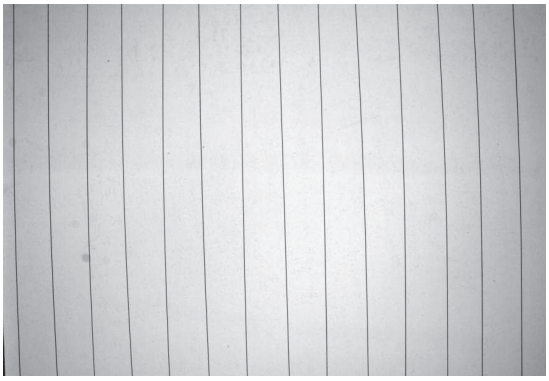
Figure 6.8: The harp with an opaque object or a translucent paper as background. (a) The harp with an uniform opaque object as background (see a close-up in (c)). (b) The harp with a translucent paper as background (see a close-up in (d)). The shadow can be seen in (a) and (c), while there is no shadow in (b) or (d).



(a) (b) (c)

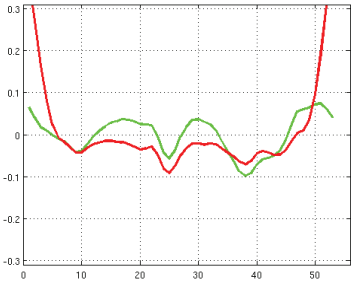
Figure 6.9: The quality of lines. (a) sewing line. (b) tennis racket line. (c) opaque fishing line.

(a) The photo of the harp is taken against the sky

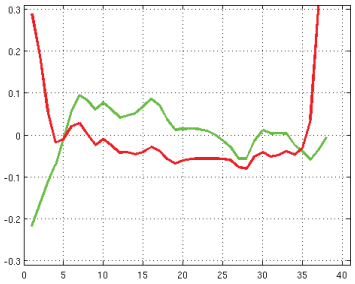


(b) The photo of the harp is taken against a translucent paper by a tripod

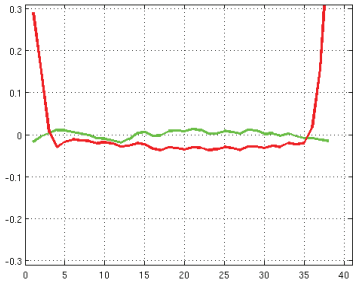
Figure 6.10: The quality of photos depends on the harp, its background and also the stability of camera for taking photos.



(a) The sewing string



(b) The tennis racket string



(c) The opaque fishing string

Figure 6.11: The small oscillation of the corrected lines is related to the quality of the strings. The green curve shows the RMS distance (in pixels) from the edge points of a corrected line to its regression line. The red curve shows the high frequency of the corresponding distorted line. The corrected line inherits the oscillation from the corresponding distorted line. (a) the sewing string. (b) the tennis racket string. (c) the opaque fishing line. The x-axis is the index of edge points. The range of y-axis is from -0.3 pixels to 0.3 pixels.

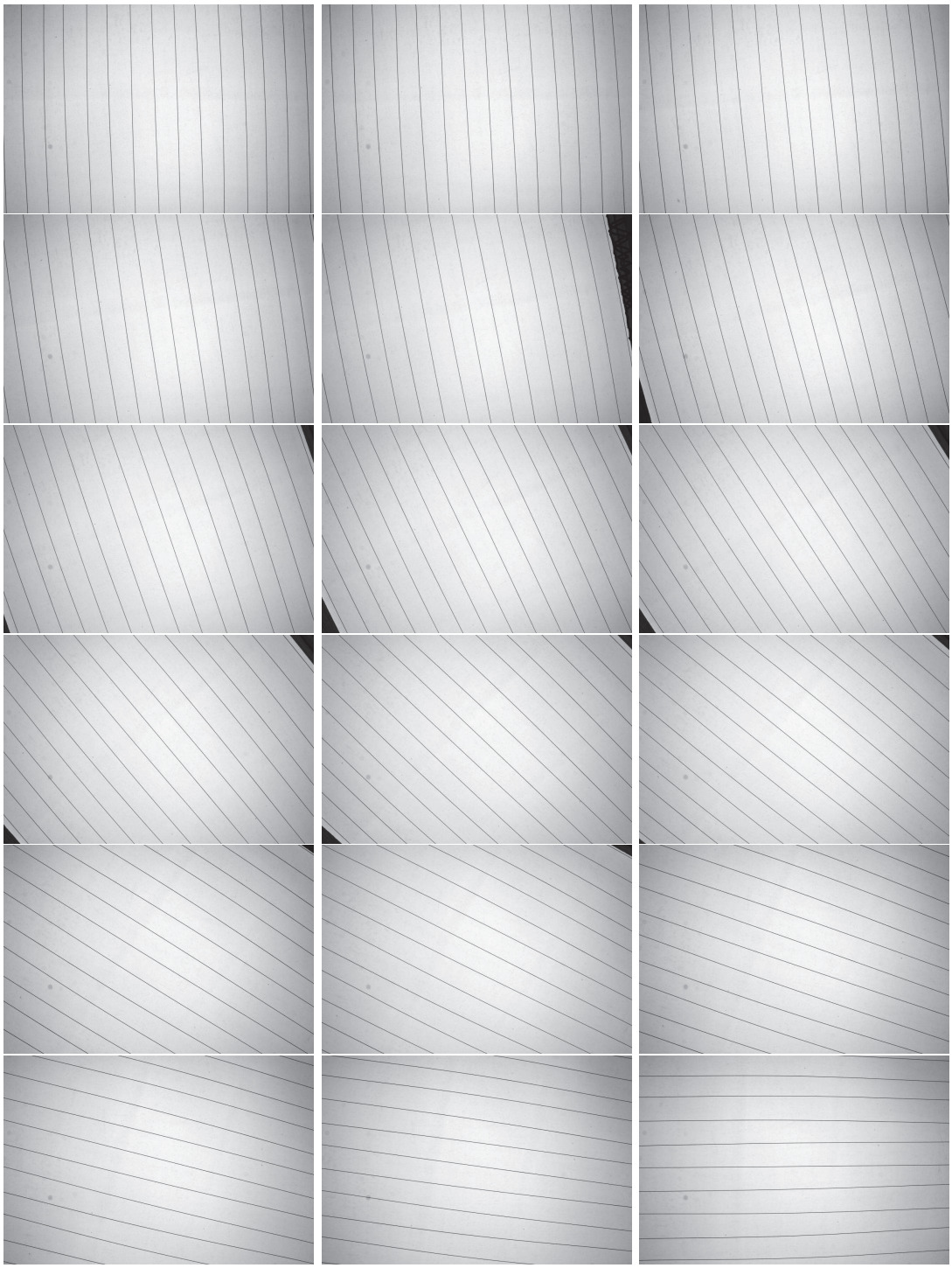


Figure 6.12: Distorted opaque fishing strings taken by the camera fixed on a tripod with different orientations.

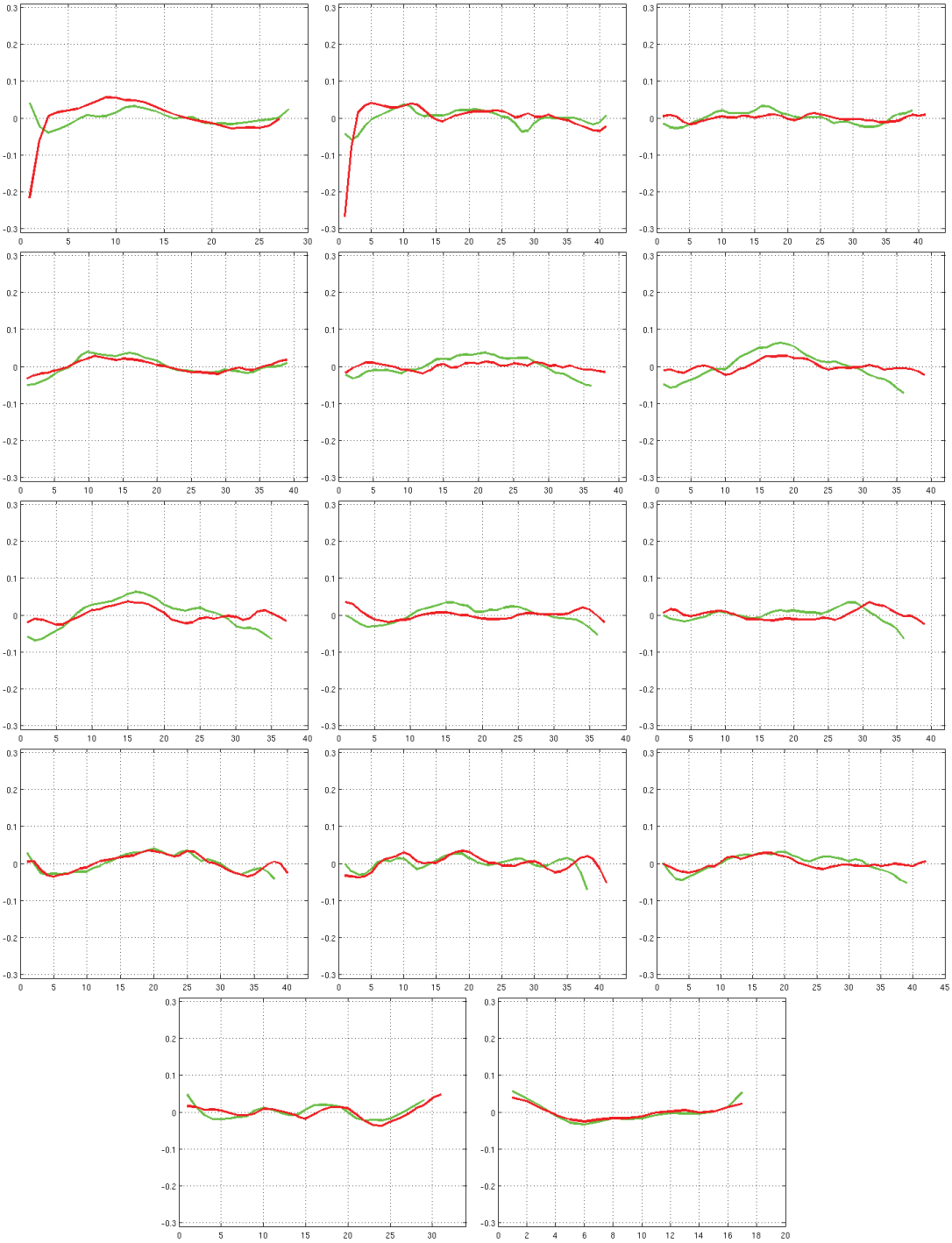


Figure 6.13: Correction performance of the proposed plumb-line method with a harp made up of opaque fishing strings. The camera focal length is fixed 18 mm and the distance between camera and object is about 30 cm. The x-axis is the index of edge points. The range of y-axis is from -0.3 pixels to 0.3 pixels.

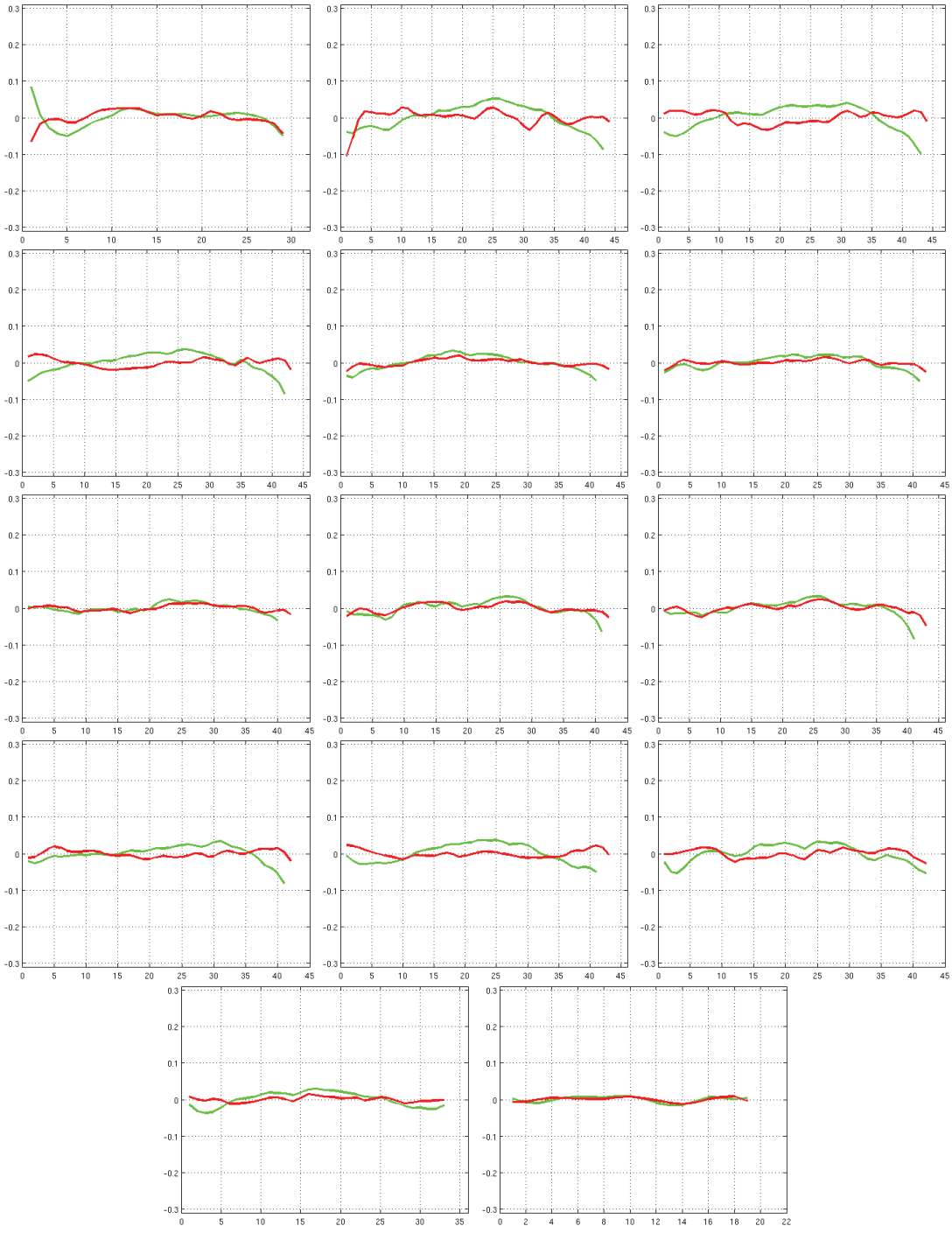


Figure 6.14: Correction performance of the proposed plumb-line method with a harp made up of opaque fishing strings. The camera focal length is fixed 55 mm and the distance between camera and object is about 100 cm. The x-axis is the index of edge points. The range of y-axis is from -0.3 pixels to 0.3 pixels.

line No.	RMSE (in pixels)	
	Near view	Far view
1	0.019/0.052	0.027/0.020
2	0.021/0.049	0.034/0.022
3	0.017/0.007	0.034/0.016
4	0.024/0.016	0.027/0.012
5	0.025/0.009	0.021/0.010
6	0.038/0.014	0.018/0.008
7	0.040/0.018	0.012/0.008
8	0.022/0.012	0.020/0.012
9	0.019/0.014	0.021/0.013
10	0.024/0.022	0.023/0.009
11	0.018/0.021	0.026/0.010
12	0.024/0.014	0.025/0.012
13	0.018/0.018	0.020/0.007
14	0.026/0.018	0.008/0.006

Table 6.6: RMS distance from the edge points of the corrected lines to their corresponding regression line. A harp made of opaque fishing strings is used. The first column corresponds to Fig. 6.13, where the camera focal length is fixed at 18 mm and the distance between camera and object is about 30 cm. The second column corresponds to Fig. 6.14, where the camera focal length is fixed at 55 mm and the distance between camera and object is about 100 cm. Note that each cell in the table contains two values because there are two sides for each string.

6.5 The Correction Performance of Global Camera Calibration is Unstable

We have shown a correction example by the global camera calibration method (Lavest *et al.* method). It does not correct the distortion as well as the plumb-line based method (see the comparison in Table 6.5 or Fig. 6.3 and Fig. 6.5). But a probably more worrying drawback of global camera calibration is that it does not give a stable correction, due to the error compensation among many parameters. When different distortion models are used in the global calibration, different corrections will be obtained. This is a serious problem because *a priori* we do not know which distortion model and which parameters are more appropriate for a given camera. The results in Fig. 6.5 were obtained by using Brown's classic distortion model [25] with 2 radial parameters and 2 tangential parameters in Lavest *et al.* method. Table 6.7 compares the correction precision by using different number of distortion parameters. It is clear that the correction precision varies. We get the best precision by using 5 radial parameters, but no way to predict it *a priori*. Furthermore, an increase in the number of parameters of model should never decrease the model fitness.

line No.	RMSE (in pixels)			
	2 radial params	5 radial params	2 radial + 2 tangent	5 radial + 2 tangent
1	0.038/0.068	0.032/0.061	0.035/0.062	0.031/0.057
2	0.172/0.173	0.139/0.140	0.217/0.218	0.172/0.175
3	0.190/0.191	0.154/0.160	0.267/0.270	0.219/0.220
4	0.101/0.087	0.099/0.089	0.156/0.139	0.131/0.115
5	0.068/0.085	0.062/0.077	0.118/0.137	0.086/0.106
6	0.055/0.042	0.044/0.035	0.108/0.099	0.076/0.066
7	0.023/0.054	0.029/0.028	0.090/0.072	0.085/0.057
8	0.085/0.072	0.064/0.048	0.117/0.127	0.085/0.103
9	0.073/0.062	0.060/0.036	0.080/0.095	0.073/0.083
10	0.057/0.039	0.047/0.027	0.122/0.120	0.030/0.014

Table 6.7: The RMSE of corrected lines under estimated parameters by Lavest *et al.* method with different number of distortion parameters.

6.6 Conclusion

Global camera calibration methods are not reliable to correct the distortion with very high accuracy. By combining the advantages of a model-free polynomial approximation and of a real plumb line pattern, the proposed lens distortion correction is significantly more accurate than parametric methods on flat patterns. The "calibration harp" construction only requires the acquisition of a string with decent quality. It is far simpler than realizing a flat plate with highly accurate patterns engraved on it. (The calibration of such patterns is not easier than lens calibration itself!) The high number of degrees of freedom in the unstructured model explains why we can call the method model-free. The only assumption on the lens distortion

is its smoothness, implying that a polynomial with high enough order approximates it. In our experiments, the approximation error stabilizes for polynomials of degree 7 to 11. It might be objected that the high number of parameters in the polynomial interpolation (156 for an 11-order polynomial) could cause some bias in the result. Yet, the number of control points is far higher: There were about 10 strings for each orientation, some 30 control points on each string side, and some 18 orientations. Thus the number of control points is about 10000 and therefore 60 times more than the number of polynomial coefficients. A visual examination of the two sides of the strings confirms that no artificial simultaneous bias has been introduced by the polynomial distortion correction. This observation seems to indicate that a good part of the 0.05 pixels remaining oscillation is due either to image processing factors, or to background inhomogeneity, to aliasing in the edge detector, or to string diameter variations. By building a harp of better quality with more smooth opaque fishing strings, the residual oscillation is largely reduced and we indeed gain a factor about 2 or even 3.

Chapter 7

Three-Step Image Rectification

Contents

Abstract *Image stereo-rectification is the process by which two images of the same solid scene undergo homographic transforms, so that their corresponding epipolar lines coincide and become parallel to the x-axis of image. A pair of stereo-rectified images is helpful for dense stereo matching algorithms. It restricts the search domain for each match to a line parallel to the x-axis. Due to the redundant degrees of freedom, the solution to stereo-rectification is not unique and actually can lead to undesirable distortions or be stuck in a local minimum of the distortion function. In this chapter a robust geometric stereo-rectification method by a three-step camera rotation is proposed and mathematically explained. Unlike other methods which reduce the distortion by explicitly minimizing an empirical measure, the intuitive geometric camera rotation angle is minimized at each step. For un-calibrated cameras, this method uses an efficient minimization algorithm by optimizing only one natural parameter, the focal length. This is in contrast with all former methods which optimize between* 3 *and* 6 *parameters. Comparative experiments show that the algorithm has an accuracy comparable to the state-of-art, but finds the right minimum in cases where other methods fail, namely when the epipolar lines are far from horizontal.*

7.1 Introduction

The stereo rectification of an image pair is an important component in many computer vision applications. The precise 3D reconstruction task requires an accurate dense disparity map, which is obtained by image registration algorithms. By estimating the epipolar geometry between two images and performing stereo-rectification, the search domain for registration algorithms is reduced and the comparison simplified, because horizontal lines with the same y component in both images are in one to one correspondence. Stereo-rectification methods simulate rotations of the cameras to generate two coplanar image planes that are in addition parallel to the baseline.

From the algebraic viewpoint, the rectification is achieved by applying 2D projective transformations (or homographies) on both images. This pair of homographies is not unique, because a pair of stereo-rectified images remains stereo-rectified under a common rotation of both cameras around the baseline. This remaining degree of freedom can introduce an undesirable distortion to the rectified images. In the literature, several methods have been proposed to reduce this distortion. In [88], authors first rectify one image and find another "matched" homography to rectify the other image. The distortion is reduced by imposing that one homography is approximately rigid around one point and by minimizing the x-disparity between both rectified images. In [116], the distortion reduction is improved by decomposing the homographies into three components: homograhy, similarity and shear. A projective transformation is sought, as affine as possible to reduce projective distortion, but the affine distortion is not treated. In [75], a new parametrization of the fundamental matrix based on two rectification homographies is used to fit the feature correspondences. The rectification problem is formulated as a 6-parameter non-linear minimization problem. This method is very compact but no special attention is paid to the distortion reduction. In [78], the distortion is interpreted as local loss or creation of pixels in the rectified images. Thus the local area change in the rectified images is minimized. A similar idea is exposed in [122], whose solution is a homography that can be locally well approximated by a rigid transformation through the whole image domain. The rectification problem is also studied in the special situation

where the camera projection matrix is known without explicitly reducing distortion [76, 6]. Although different measures for rectification distortion are proposed in the above methods, the distortion is minimized explicitly. However it is not clear which measure would be more appropriate for image rectification. In the method proposed here, the distortion is not minimized explicitly. The rectification process is decomposed into three steps. The first step sends both epipoles to infinity; then the epipoles are sent to infinity in the x-direction; eventually the residual rotation between both cameras around their baseline is compensated to achieve the rectification. At each step, the camera rotation induces a homography on each image whose rotation angle is minimized to reduce the distortion. In contrast with the one-step rotation proposed in [75], we shall see that the three-step rotation makes the algorithm more robust. Even in extreme cases where the initial epipolar lines are far from horizontal, the algorithm works well while all other algorithms fail. The method yields a result, no matter whether the camera calibration matrix is known or not. In the latter case the proposed method can be easily formulated as a one-parameter minimization problem under the assumption of square aspect-ratio, zero skewness and image center as principal point. But unlike some methods treating arbitrary geometry [148, 140], our method can only treat the case where the epipoles are outside the image domain. The method is detailed in Section 7.2. Some results are compared and commented in Section 7.3 followed by a conclusion in Section 8.4.

7.2 Description of the Method

Space or image points will be denoted by lowercase bold letters and matrices by uppercase bold letters. The rectification works in the two-dimensional projective space $\mathcal{P}^2$. A point is a 3-vector in $\mathcal{P}^2$, for example, $\mathbf{m} = (x, y, w)^T$, corresponding to the Euclidean point $(x/w, y/w)^T$. If $w = 0$, then the point is at infinity in the (x, y) direction. A transformation in the two-dimensional projective space $\mathcal{P}^2$ is a 3×3 matrix. Examples of such transformations are the fundamental matrix, denoted by $\mathbf{F}$ and homographies denoted by $\mathbf{H}$.

As usual in stereo-rectification, a set of non-degenerate correspondences between image I_1 and I_2 are given, permitting to compute the correct fundamental matrix $\mathbf{F}$. For that purpose, the SIFT algorithm [117] followed by a RANSAC-like algorithm [67, 129, 72] yields a reliable enough input with inliers only.

7.2.1 Rectification Geometry

The fundamental matrix corresponds to two stereo-rectified images if and only if it has the special form (up to a scale factor)

$$[\mathbf{i}]_\times = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}_\times = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}. \tag{7.1}$$

Having both cameras pointing to the same direction with their image planes co-planar and parallel to the baseline is still not sufficient to achieve rectification. Assume the cameras to have the form $\mathbf{P} = \mathbf{K}[\mathbf{I} \mid \mathbf{0}]$ and $\mathbf{P}' = \mathbf{K}'[\mathbf{I} \mid \mathbf{i}]$ with the motion between both cameras being

only the translation along the x-axis and

$$\mathbf{K} = \begin{bmatrix} f_x & \alpha & u \\ 0 & f_y & v \\ 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{K}' = \begin{bmatrix} f'_x & \alpha' & u' \\ 0 & f'_y & v' \\ 0 & 0 & 1 \end{bmatrix} \tag{7.2}$$

calibration matrices. Then the fundamental matrix is:

$$\mathbf{F} = \mathbf{K}^{-T}[\mathbf{i}]_\times \mathbf{K}^{-1} = \frac{1}{f_y f'_y} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -f_y \\ 0 & f'_y & v' f_y - v f'_y \end{bmatrix} \tag{7.3}$$

So the fundamental matrix has the form $[\mathbf{i}]_\times$ only when the following condition is satisfied:

$$f_y = f'_y \ , \ v = v'. \tag{7.4}$$

This is equivalent to say that $\mathbf{K}$ and $\mathbf{K}'$ have the same second row. So to achieve the rectification, both the orientation and calibration matrix of cameras need to be ajusted.

The orientation of the camera can be adjusted by applying a homography on the image, which has the form:

$$\mathbf{H} = \mathbf{K}\mathbf{R}\mathbf{K}^{-1} \tag{7.5}$$

where $\mathbf{R}$ is the relative rotation before and after rectification. Is the rectification achieved by finding a pair of homographies which sends the epipoles in each image to $(1, 0, 0)^T$ (x-direction)? The answer is NO. Having the epipoles at $(1, 0, 0)^T$ only means the relationship between two cameras is a rotation around the baseline, which corresponds to a fundamental matrix of the form:

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & a & b \\ 0 & c & d \end{bmatrix} \tag{7.6}$$

with $ad - bc \neq 0$, which is still different from $[\mathbf{i}]_\times$. However, under the condition that K is correct and the camera model is ideal pinhole, the remaining rotation matrix around the baseline between two cameras can be extracted from this special fundamental matrix in Eq. (7.6).

7.2.2 Calibrated Case

We first treat the case where cameras are calibrated. This is the same situation as [76] except that in their case cameras projective matrices are also known. To simplify the problem, assume the same calibration matrix $\mathbf{K}$ for two cameras before and after rectification and no lens distortion is considered. Then a perfect epipolar geometry can be computed from a group of non-degenerate correspondences between two images. Note $\mathbf{F}$ the fundamental matrix, $\mathbf{x}_i, \mathbf{x}'_i$ the correspondences. The epipole for the left image $\mathbf{e} = (e_x, e_y, 1)^T$ and right image $\mathbf{e}' = (e'_x, e'_y, 1)^T$ can be computed as right and left null vector: $\mathbf{F}\mathbf{e} = 0$ and $\mathbf{e}'^T\mathbf{F} = 0$. The idea is to transform two images such that the fundamental matrix becomes $[\mathbf{i}]_\times$. Unlike the other methods which directly parametrizes the homographies from the constraint $\mathbf{H}\mathbf{e} = \mathbf{i}$, $\mathbf{H}'\mathbf{e}' = \mathbf{i}$ and $\mathbf{H}'^T[\mathbf{i}]_\times\mathbf{H} = \mathbf{F}$ and find an optimal one by minimizing the measure of distortion, we compute the homography by explicitly rotating the camera around its optical center. The algorithm is decomposed into three steps (Fig. 7.1):

1. Compute homographies $\mathbf{H}_1$ and $\mathbf{H}'_1$ by rotating two cameras respectively such that left epipole $(e_x, e_y, 1)$ is transformed to $(e_x, e_y, 0)$ and right epipole $(e'_x, e'_y, 1)$ to $(e'_x, e'_y, 0)$.

2. Continue to rotate two cameras such that $(e_x, e_y, 0)$ is transformed to $(1, 0, 0)$ and $(e'_x, e'_y, 1)$ to $(1, 0, 0)$. The corresponding homographies are $\mathbf{H}_2$ and $\mathbf{H}'_2$.

3. Rotate one camera or two cameras together to compensate the remaining relative rotation between two cameras around baseline. The corresponding homographies are $\mathbf{H}_3$ and $\mathbf{H}'_3$.

For the first step, we have the relationship: $\mathbf{H}_1\mathbf{e} = (e_x, e_y, 0)^T$ and $\mathbf{H}'_1\mathbf{e}' = (e'_x, e'_y, 0)^T$ with $\mathbf{H}_1 = \mathbf{K}\mathbf{R}\mathbf{K}^{-1}$ and $\mathbf{H}'_1 = \mathbf{K}\mathbf{R}'\mathbf{K}^{-1}$. The only unknown is two rotation matrices but the solutions to them are not unique. We find a rotation matrix which need a minimal rotation angle, then introduces less distortion. By rewriting $\mathbf{H}_1\mathbf{e} = (e_x, e_y, 0)^T$ as $\mathbf{R}\mathbf{K}^{-1}\mathbf{e} = \mathbf{K}^{-1}(e_x, e_y, 0)^T$, the problem is in fact to find a rotation matrix which rotate the vector $\mathbf{K}^{-1}\mathbf{e}$ to $\mathbf{K}^{-1}(e_x, e_y, 0)^T$. So the minimal angle θ is $\text{acos}(\frac{\mathbf{a}\cdot\mathbf{b}}{|\mathbf{a}||\mathbf{b}|})$ and the rotation axis t is $\frac{\mathbf{a}\times\mathbf{b}}{|\mathbf{a}||\mathbf{b}|}$. with $\mathbf{a} = \mathbf{K}^{-1}(e)$, $\mathbf{b} = \mathbf{K}^{-1}(e_x, e_y, 0)^T$. (see Fig. ??). By Rodrigues' formula, the rotation can be written as:

$$\mathbf{R}(\theta, \mathbf{t}) = \mathbf{I} + \sin\theta[\mathbf{t}]_\times + (1 - \cos\theta)[\mathbf{t}]^2_\times \tag{7.7}$$

This process can be repeated in step 2. After two steps, two epipoles are both transformed to $(1, 0, 0)^T$, the rectification has not finished and the fundamental matrix has the form like equation (7.6). With the assumption that $\mathbf{K}$ is correct and no other effects deviating the camera from ideal pinhole camera is considered, the following proposition proves that the remaining relationship between two cameras is a rotation $\hat{\mathbf{R}}$ around the baseline. So $\hat{\mathbf{F}}$ can be written as $\mathbf{K}^{-T}[\mathbf{i}]_\times\hat{\mathbf{R}}\mathbf{K}^{-1}$.

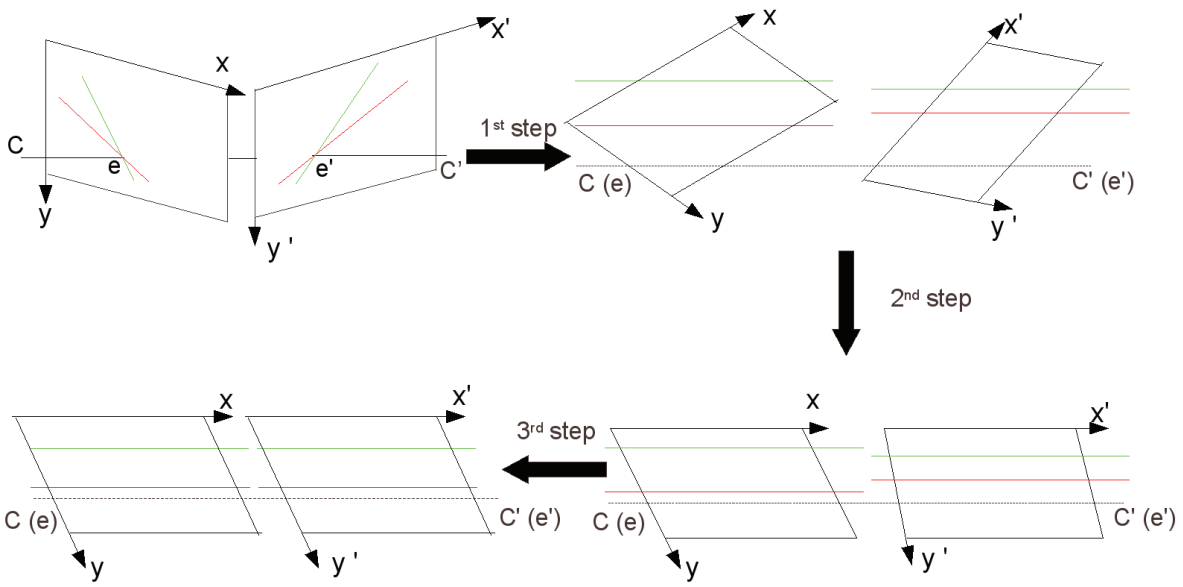


Figure 7.1: Three-step rectification.

Someone maybe argues that one step of rotation is enough to rectify the images instead of three steps. The reason is that three steps of rotations makes the algorithm more robust. In fact, the idea of parametrization of the fundamental matrix by just one step of camera rotation is used in [75]. As we will see in the experiments, this parametrization is not robust when the initial epipolar lines are far from horizontal.

Theorem 3 *Given the fundamental matrices* $\mathbf{F}$ *of two calibrated cameras, after the first two steps of camera rotation, the cameras have the same orientation except a rotation around the baseline.*

Proof Write the original $\mathbf{F} = \mathbf{K}^{-T}[\mathbf{t}]_\times \mathbf{R}\mathbf{K}^{-1}$ with $\mathbf{t}$, $\mathbf{R}$ the translation and rotation between two cameras. A new fundamental matrix is generated after two steps of camera rotations: $\hat{\mathbf{F}} = \mathbf{H}_2'^{-T}\mathbf{H}_1'^{-T}\mathbf{F}\mathbf{H}_1^{-1}\mathbf{H}_2^{-1}$ with $\mathbf{H}_1 = \mathbf{K}\mathbf{R}_1\mathbf{K}^{-1}$, $\mathbf{H}_1' = \mathbf{K}\mathbf{R}_1'\mathbf{K}^{-1}$ and $\mathbf{H}_2 = \mathbf{K}\mathbf{R}_2\mathbf{K}^{-1}$, $\mathbf{H}_2' = \mathbf{K}\mathbf{R}_2'\mathbf{K}^{-1}$. So finally, we have:

$$\begin{aligned}\hat{\mathbf{F}} &= \mathbf{H}_2'^{-T}\mathbf{H}_1'^{-T}\mathbf{F}\mathbf{H}_1^{-1}\mathbf{H}_2^{-1} = \mathbf{K}^{-T}\mathbf{R}_2'\mathbf{R}_1'[\mathbf{t}]_\times \mathbf{R}\mathbf{R}_1^{-1}\mathbf{R}_2^{-1}\mathbf{K}^{-1} \\ &= \mathbf{K}^{-T}[\mathbf{R}_2'\mathbf{R}_1'\mathbf{t}]_\times \mathbf{R}_2'\mathbf{R}_1'\mathbf{R}\mathbf{R}_1^{-1}\mathbf{R}_2^{-1}\mathbf{K}^{-1} \\ &= \mathbf{K}^{-T}[\mathbf{R}_2'\mathbf{R}_1'\mathbf{R}(\mathbf{C}-\mathbf{C}')]_\times \mathbf{R}_2'\mathbf{R}_1'\mathbf{R}\mathbf{R}_1^{-1}\mathbf{R}_2^{-1}\mathbf{K}^{-1} \\ &= \mathbf{K}^{-T}[\mathbf{i}]_\times \hat{\mathbf{R}}\mathbf{K}^{-1}\end{aligned} \tag{7.8}$$

with $\hat{\mathbf{R}} = \mathbf{R}_2'\mathbf{R}_1'\mathbf{R}_1^{-1}\mathbf{R}_2^{-1}$ the remaining rotation between two cameras around baseline after two steps of camera rotations. Note that the translation vector $\mathbf{t} = \mathbf{R}(\mathbf{C}-\mathbf{C}')$ with $\mathbf{C}$, $\mathbf{C}'$ the optical center of cameras. □

Once $\hat{\mathbf{F}} = \mathbf{K}^{-T}[\mathbf{i}]_\times \hat{\mathbf{R}}\mathbf{K}^{-1}$, the essential matrix is: $\hat{\mathbf{E}} = [\mathbf{i}]_\times \hat{\mathbf{R}}$. According to [89], two possible rotations $\hat{\mathbf{R}}$ can be retrieved from $\hat{\mathbf{E}}$ and only one is physical. $\hat{\mathbf{R}}$ can be compensated by rotating one camera or two together.

All of the above discussion is based on the assumptions of perfect camera calibration matrix, ideal pinhole model and correct fundamental matrix estimation. But in real case, none of them is true. This will make fundamental matrix ($\mathbf{F}$), calibration matrix ($\mathbf{K}$) not compatible. This is the problem we want to resolve for uncalibrated cameras in the following section.

Definition 1 *We call* $\mathbf{F}$, $\mathbf{K}$ *are compatible if* $\mathbf{E} = \mathbf{K}^T\mathbf{F}\mathbf{K}$ *is an essential matrix, that is,* $\mathbf{E}$ *has two equal singular values and the third one is zero.*

7.2.3 Uncalibrated Case

In practice the calibration matrices are not always available for images. In such case, the rectification can be performed in the same framework as the calibrated case by finding an appropriate calibration matrix. To simplify the context, we assume two cameras have the same calibration matrix with the form:

$$\mathbf{K} = \begin{bmatrix} f & 0 & \frac{w}{2} \\ 0 & f & \frac{h}{2} \\ 0 & 0 & 1 \end{bmatrix} \tag{7.9}$$

with principal point at the center of the image $(\frac{w}{2}, \frac{h}{2})$ and f the unknown focal length. So $\mathbf{H}_1$, $\mathbf{H}'_1$, $\mathbf{H}_2$, $\mathbf{H}'_2$ can be parametrized by f. By applying these homographies on images, we obtain a new fundamental matrix: $\hat{\mathbf{F}} = \mathbf{H}_2'^{-T}\mathbf{H}_1'^{-T}\mathbf{F}\mathbf{H}_1^{-1}\mathbf{H}_2^{-1}$. Unlike the ideal calibrated case, $\hat{\mathbf{F}}$ is not necessarily compatible with $\mathbf{K}$. In other words, $\hat{\mathbf{E}} = \mathbf{K}^T\hat{\mathbf{F}}\mathbf{K}$ is not an essential matrix. So $\hat{\mathbf{E}}$ cannot be decomposed into the form like $[\mathbf{t}]_\times\mathbf{R}$. Thus the third step of rectification will fail. By writing $\hat{\mathbf{E}} = \mathbf{U}\mathbf{D}\mathbf{V}'$, one possible modification is to force two singular values of $\hat{\mathbf{E}}$ equal and the third one 0, which gives $\tilde{\hat{\mathbf{E}}} = \mathbf{U}\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}\mathbf{V}^T$ and $\tilde{\hat{\mathbf{F}}} = \mathbf{K}^{-T}\tilde{\hat{\mathbf{E}}}\mathbf{K}^{-1}$. This modification is the smallest in the sense of Frobenius norm. Then The third step of rectification can be continued by using the extracted rotation matrix from $\tilde{\hat{\mathbf{E}}}$. But this modification can also change a lot the epipolar geometry. This can be evaluated by checking the sum of distance from the points to the modified epipolar lines:

$$\begin{aligned} S(f) &= \sum_{i=1}^{N} d(\mathbf{x}'_i, \tilde{\mathbf{F}}(f)\mathbf{x}_i) + d(\mathbf{x}_i, \tilde{\mathbf{F}}(f)^T\mathbf{x}'_i) \\ &= \sum_{i=1}^{N}\left(\left(\mathbf{x}_i'^T\tilde{\mathbf{F}}\mathbf{x}_i\right)^2\left(\frac{1}{(\tilde{\mathbf{F}}\mathbf{x}_i)_1^2 + (\tilde{\mathbf{F}}\mathbf{x}_i)_2^2} + \frac{1}{(\tilde{\mathbf{F}}^T\mathbf{x}'_i)_1^2 + (\tilde{\mathbf{F}}^T\mathbf{x}'_i)_2^2}\right)\right) \end{aligned} \tag{7.10}$$

with $\tilde{\mathbf{F}} = \mathbf{H}_1'^T\mathbf{H}_2'^T\tilde{\hat{\mathbf{F}}}\mathbf{H}_2\mathbf{H}_1$. An optimal $\mathbf{K}$ can be found by minimizing the distance function $S(f)$. $S(f)$ is a non-linear function of focal length f. Levenberg-Marquardt minimization algorithm (see Appendix A.3) is chosen to find an optimal f. For more stability and efficiency, the Jacobian matrix $\frac{\partial S(f)}{\partial f}$ is computed explicitly instead of using finite difference scheme. The delicate part of the Jacobian computation is $\frac{\partial\tilde{\hat{\mathbf{E}}}}{\partial f}$, which can be resolved by using a method proposed in [146]. By deriving $\hat{\mathbf{E}} = \mathbf{U}\mathbf{D}\mathbf{V}'$ with respect of f, we have:

$$\frac{\partial\hat{\mathbf{E}}}{\partial f} = \frac{\partial\mathbf{U}}{\partial f}\mathbf{D}\mathbf{V}^T + \mathbf{U}\frac{\partial\mathbf{D}}{\partial f}\mathbf{V}^T + \mathbf{U}\mathbf{D}\frac{\partial\mathbf{V}^T}{\partial f} \tag{7.11}$$

By multiplying $\mathbf{U}^T$ at left and $\mathbf{V}$ at right:

$$\mathbf{U}^T\frac{\partial\hat{\mathbf{E}}}{\partial f}\mathbf{V} = \mathbf{U}^T\frac{\partial\mathbf{U}}{\partial f}\mathbf{D} + \frac{\partial\mathbf{D}}{\partial f} + \mathbf{D}\frac{\partial\mathbf{V}^T}{\partial f}\mathbf{V} \tag{7.12}$$

On the other hand, since $\mathbf{V}^T\mathbf{V} = \mathbf{I}$, we have $\frac{\partial\mathbf{V}^T}{\partial f}\mathbf{V} + \mathbf{V}^T\frac{\partial\mathbf{V}}{\partial f} = 0$. That means $\frac{\partial\mathbf{V}^T}{\partial f}\mathbf{V}$ is anti-symmetric, so is $\mathbf{U}^T\frac{\partial\mathbf{U}}{\partial f}$. By noting $\mathbf{U}^T\frac{\partial\mathbf{U}}{\partial f} = \Omega_\mathbf{U}$ and $\frac{\partial\mathbf{V}^T}{\partial f}\mathbf{V} = \Omega_\mathbf{V}$, Eq. (7.12) becomes:

$$\mathbf{U}^T\frac{\partial\hat{\mathbf{E}}}{\partial f}\mathbf{V} = \Omega_\mathbf{U}\mathbf{D} + \frac{\partial\mathbf{D}}{\partial f} + \mathbf{D}\Omega_\mathbf{V} \tag{7.13}$$

Since $\Omega_\mathbf{U}$ and $\Omega_\mathbf{V}$ are anti-symmetric, their diagonal elements are all zeros. The following equations can be obtained by observing the off-diagonal elements of Eq. (7.13):

$$\begin{cases} (\Omega_\mathbf{U})_{k,l}\,\mathbf{D}_{l,l} + \mathbf{D}_{k,k}\,(\Omega_\mathbf{V})_{k,l} = \left(\mathbf{U}^T\frac{\partial\hat{\mathbf{E}}}{\partial f}V\right)_{k,l} \\ (\Omega_\mathbf{U})_{k,l}\,\mathbf{D}_{k,k} + \mathbf{D}_{l,l}\,(\Omega_\mathbf{V})_{k,l} = \left(\mathbf{U}^T\frac{\partial\hat{\mathbf{E}}}{\partial f}V\right)_{k,l} \end{cases} \quad k = 1,2,3; l = k+1,\cdots,3 \tag{7.14}$$

This is a linear equation system for each (k, l) pair:

$$\begin{pmatrix} \mathbf{D}_{l,l} & \mathbf{D}_{k,k} \\ \mathbf{D}_{k,k} & \mathbf{D}_{l,l} \end{pmatrix} \begin{pmatrix} (\Omega_{\mathbf{U}})_{k,l} \\ (\Omega_{\mathbf{V}})_{k,l} \end{pmatrix} = \begin{pmatrix} \left(\mathbf{U}^T \frac{\partial \hat{\mathbf{E}}}{\partial f} \mathbf{V}\right)_{k,l} \\ \left(\mathbf{U}^T \frac{\partial \hat{\mathbf{E}}}{\partial f} \mathbf{V}\right)_{k,l} \end{pmatrix} \tag{7.15}$$

Once $\Omega_{\mathbf{U}}$ and $\Omega_{\mathbf{V}}$ are solved, we have:

$$\frac{\partial \mathbf{U}}{\partial f} = \mathbf{U}\Omega_{\mathbf{U}}, \quad \frac{\partial \mathbf{V}}{\partial f} = -\mathbf{V}\Omega_{\mathbf{V}} \tag{7.16}$$

So finally, $\frac{\partial \tilde{\hat{\mathbf{E}}}}{\partial f} = \frac{\partial \mathbf{U}}{\partial f} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \mathbf{V}^T + \mathbf{U} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \frac{\partial \mathbf{U}^T}{\partial f}$. Once the optimal $\mathbf{K}$ is found, $\mathbf{H}_3$ and $\mathbf{H}'_3$ can be computed from the residual rotation.

It might be argued that one step of rotation is enough to rectify the images instead of three steps. But the three steps procedure makes the algorithm much more robust and, as we shall see, the residual distortion is equal or only slightly higher. In fact, the idea of parametrization of the fundamental matrix by just one step of camera rotation is used in [75]. As we will see in the experiments, this parametrization is not robust when the initial epipolar lines are far from horizontal.

7.3 Results

In this section, the algorithm is tested on several pairs of real images. In Table 7.1, the first three pairs of images are from Mallon's test set [122], which can be taken by the same camera under a fixed lens configuration. Concerning the other three pairs of images, the camera motion causes the initial epipolar lines to be far from the horizontal direction. For all pairs, enough correct correspondences are available. The fundamental matrix was computed by the normalized 8-point algorithm [87] and the calibration matrix is unknown. The performance of the algorithm is compared with Hartley [88][1], Fusiello *et al.* [75][2] and Mallon *et al.* [122][3] methods.

The performance is evaluated on two aspects: the rectification error and the introduced distortion. The rectification error is measured as the average and standard deviation of the y-disparity of rectified correspondences. The same statistics are computed for the original epipolar geometry with the distance from points to the corresponding epipolar lines as metric.

The distortion reduction is measured by two traditional criteria: orthogonality and aspect ratio. Consider the rectification homography $\mathbf{H}$ and four cross points $\mathbf{a} = (\frac{w}{2}, 0, 1)^T$, $\mathbf{b} = (w, \frac{h}{2}, 1)^T$, $\mathbf{c} = (\frac{w}{2}, h, 1)^T$, $\mathbf{d} = (0, \frac{h}{2}, 1)^T$. with w and h the width and the height of image. The orthogonality θ_o is defined as the angle between the vector $\mathbf{m} = \mathbf{Hb} - \mathbf{Hd}$ and $\mathbf{n} = \mathbf{Hc} - \mathbf{Ha}$: $\theta_o = \cos^{-1}\left(\frac{\mathbf{m}\cdot\mathbf{n}}{|\mathbf{m}||\mathbf{n}|}\right)$. The ideal value of orthogonality is 90°. By redefining $\mathbf{a} = (0, 0, 1)^T$, $\mathbf{b} = (w, 0, 1)^T$, $\mathbf{c} = (w, h, 1)^T$, $\mathbf{d} = (0, h, 1)^T$, the aspect ratio r_d is the length ratio between diagonals: $r_d = \left((\mathbf{m}^T\mathbf{m})/((\mathbf{n}^T\mathbf{n}))\right)^{1/2}$. The ideal value of the aspect ratio is 1.

[1] Du Huynh's version: `http://www.csse.uwa.edu.au/~du/Software/rectification/`

[2] Code available at: `http://profs.sci.univr.it/~fusiello/demo/rect/`.

[3] Only the first three example in Table 7.1 are tested by Mallon *et al.*'s since the code is not available.

We argue that the above criteria are not sufficient. A rectification should also have a geometric meaning. For some pairs of images, we can deduce the rectified images since the camera motion is evident. This gives an empirical evaluation of the geometric meaning of the rectification.

The results are gathered in Table 7.1. In the first two examples ("Boxes" and "Arch", Fig. 7.2), the performance of the three algorithms are similar except that Hartley's method introduces more distortion. This is not surprising because the distortion is just reduced by minimizing the disparity in x-axis, which is not directly related to distortion. In the third example ("Drive"), the proposed method has a slightly larger rectification error, but the rectification error is still coherent with the original error and in a reasonable range.

Fusiello *et al.*'s method is very competitive, in particular for the rectification precision. But its result does not always have a correct geometric meaning. In the example of "Building" (Fig. 7.3) and "Tower", two pairs of aerial images were taken by a camera installed on a helicopter. Since the motion is close to the y-axis of the camera, the initial epipolar lines are close to vertical. In such situation, a correct rectification algorithm should rotate both cameras so that the baseline is parallel to the x-axis. Only our algorithm rotates the images and therefore gives a small rectification error. Neither Fusiello's method nor Hartley's method rotates the images, producing bigger rectification error. In the example of "Cournot" (Fig. 7.4), the initial epipolar lines were also far from the horizontal direction. The images should have been rotated to achieve a good rectification. Even though Fusiello's method gives a result with a small rectification error, the geometry of the rectified images is not correct.

Notice that for Hartley's method the orthogonality of the homography for the right image is always 90°. Indeed the right homography has the form $\mathbf{GRT}$ where $\mathbf{R}$ is a rotation matrix and $\mathbf{T}$ a translation matrix. $\mathbf{G}$ is close to a rigid transformation if the epipole is far from the image domain, which is the case of the images in the experiments. The same phenomenon can be observed for the orthogonality for the left homography of Fusiello *et al.*'s method. In their algorithm, the left camera does not rotate around x-axis. And the rotation around y-axis and z-axis is also very small if the epipole is far away.

7.4 Conclusion

A new image rectification algorithm was proposed. This algorithm is decomposed into three steps of camera rotation. By computing the minimal rotation angle at each step, the distortion is implicitly limited. This algorithm performs as well as state-of-art algorithms, but for image pairs where the initial epipolar lines are far from horizontal, the fact that we have a unique parameter to estimate (focal length) makes the algorithm more robust by reducing the risks of reaching a local minimum.

Figure 7.2: Image pair "Arch" rectified by different methods. From top to bottom: original images, proposed method, Hartley method, Fusiello *et al.* method and Mallon *et al.* method. A horizontal line is added to images to check the rectification. The third column represents an image average of each pair.

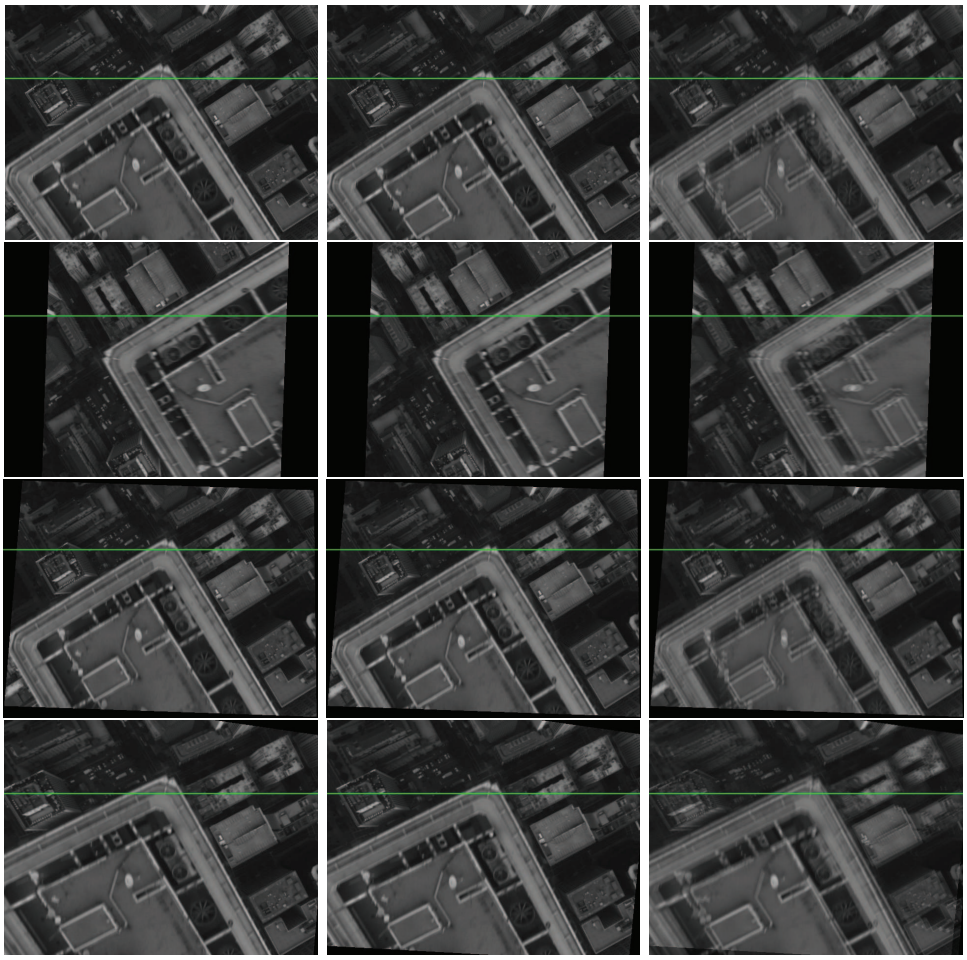


Figure 7.3: Image pair "Building" rectified by different methods. From top to bottom: original images, proposed method, Hartley method and Fusiello *et al.* method. A horizontal line is added to images to check the rectification. The third column represents an image average of each pair.

Figure 7.4: Image pair "Cournot" rectified by different methods. From top to bottom: original images, proposed method, Hartley method and Fusiello *et al.* method. A horizontal line is added to images to check the rectification. The third column represents an image average of each pair.

Sample	**F** Mat.	Method	Orthogonality		Aspect ratio		Rectification	
			H	**H′**	**H**	**H′**	mean	std
Boxes	0.1213 0.0963	Proposed	89.60	89.63	0.9884	0.9892	0.1293	0.0887
		Hartley	94.07	90.00	1.0639	0.9948	0.1194	0.0968
		Fusiello	90.00	90.16	1.0000	1.0043	0.1055	0.0891
		Mallon	89.33	88.78	0.9889	0.9878	0.44	0.33
Arch	0.2107 0.2247	Proposed	89.80	90.05	0.9942	1.0014	0.2520	0.2349
		Hartley	82.80	90.00	0.8841	1.0002	0.2089	0.2199
		Fusiello	90.00	90.19	1.0000	1.0051	0.2134	0.2593
		Mallon	90.26	91.22	1.0045	1.0175	0.22	0.33
Drive	0.5111 0.7445	Proposed	89.95	90.00	0.9977	1.0001	0.7139	0.8253
		Hartley	91.96	90.00	1.0320	1.0001	0.5132	0.7462
		Fusiello	90.00	90.11	1.0000	1.0026	0.4962	0.7851
		Mallon	90.12	90.44	1.0021	1.0060	0.18	0.91
Building	0.1308 0.1221	Proposed	89.96	89.95	0.9990	0.9989	0.1330	0.1242
		Hartley	90.02	90.00	1.0002	0.9998	6.4910	5.4584
		Fusiello	90.00	89.85	1.0000	0.9970	3.0508	2.4809
		Mallon	n/a	n/a	n/a	n/a	n/a	n/a
Tower	0.1370 0.1154	Proposed	89.99	89.98	0.9998	0.9995	0.1438	0.1192
		Hartley	89.94	90.00	0.9990	0.9999	11.1079	3.5541
		Fusiello	90.00	89.89	1.0000	0.9979	3.2698	1.7266
		Mallon	n/a	n/a	n/a	n/a	n/a	n/a
Cournot	0.2055 0.1563	Proposed	89.66	89.29	0.9920	0.9833	0.3242	0.2082
		Hartley	89.70	90.00	0.9928	0.9950	39.9008	2.0386
		Fusiello	90.00	89.81	1.0000	0.9951	0.3315	0.2486
		Mallon	n/a	n/a	n/a	n/a	n/a	n/a

Table 7.1: The performance comparison between the proposed method, Hartley's method [88], Fusiello *et al.*'s method [75] and Mallon *et al.*'s method [122]. The comparison is based on rectification error, orthogonality and aspect ratio. The ideal value of orthogonality and aspect ratio are 90° and 1 respectively.

Chapter 8

The Matching Precision of the SIFT Method

Contents

Abstract *Image features detection and matching is a fundamental step in many computer vision tasks. Many methods have been proposed in recent years, with the aim to extract image features fully invariant to any geometric and photometric transformation. Even though the state-of-art has not achieved a full invariance, many methods, like SIFT, Harris-affine and Hessian-affine combining a robust and distinctive descriptor, give sufficient invariance for many practical applications. In contrast to the advance in the invariance of feature detectors, the matching precision has not been paid enough attention, even though the repeatability and stability are extensively studied. The matching precision is evaluated on a pair of images and reflects to some extent the average relative localization precision between two images. It depends on the localization precision of feature detector, the scale change between the two images, the descriptor construction and matching protocol. In this chapter, we focus on the SIFT method and measure its matching precision as the average residual error under different geometric transformations. For scale invariant feature detectors, the matching precision decreases with the scale of features. This drawback can be avoided by canceling the sub-sampling in SIFT scale space. This first scheme improves the matching precision only when there is no scale change between both images. An iterative scheme is thus proposed to treat the scale change. For real images, a local filtering technique is used to improve the matching precision if two images are related by a smooth transformation.*

8.1 Introduction

Recent years have seen invariant feature points/regions detector thrive. Even though they are more and more invariant to image geometric transformations and illumination changes, the matching precision has not been carefully studied. Nevertheless, the matching precision is of the first importance in many computer vision applications, like super-resolution, image mosaicing, camera calibration, etc. Even if a feature detection/matching method can find many correspondences between two images, this does not mean that the matching precision is high. One feature is matched to another according to some protocol based on their descriptor. But the matching protocol is not the only factor which influences the matching precision. The matching precision is computed as the average residual error between the matchings of two images under a certain transformation. So to be more accurate, it depends on the localization precision of feature detector, the scale change between two images, the descriptor construction and the matching protocol.

8.1.1 Localization Uncertainty, Localization Precision and Matching Precision

For feature detection and matching algorithms, three concepts about precision coexist: localization uncertainty, localization precision and matching precision. More explanations are needed to not confuse them.

The localization uncertainty is the variation of the feature position introduced in the feature detection process, while the localization precision is used to describe how close *on average* the position of the detected features is to their *true position*. In statistical terms, the localization precision is called bias and the localization uncertainty is called variance. But feature detection is a deterministic process in the sense that given an image and a certain

feature detection algorithm, the position of detected features is always the same even if the detection is repeated infinitely many times. There seems to be a contradiction, but in fact a particular image or a particular feature detection algorithm is just a sample of an underlying *statistical ensemble*. This explains why a certain feature detection algorithm always extracts the same features from a certain image. In the case of feature detection, the underlying statistical ensemble can be interpreted as a collection of images and a collection of different feature detection algorithms. The collection of images can be obtained by disturbing an ideal image with noise, blur, illumination changes, etc. Assume the true position of a feature point is known. The localization precision (bias) can be measured as the distance from the average position of the detected features by a particular feature detector on differently disturbed images to the true position (taking a model for each disturbance). The localization uncertainty (variance) can be measured as the variation of the position of a certain feature point by different feature detectors on a particular image. The localization precision (bias) is related to a specific feature detector, interpreted as its robustness to precisely locate feature against the image disturbance. The localization uncertainty (variance) is related to the image local property, interpreted as the uncertainty of feature detection in that image. Neither the localization precision nor the localization uncertainty are uniform in the image domain. They can be different from one feature point to another.

In geometric structure inference, the localization uncertainty is measured by a covariance matrix for each detected feature individually. In [98], the author used two methods to estimate the covariance matrix for corner features and concluded that the accuracy of the geometric computation was not improved by incorporating the covariance matrix into the optimization. This is understandable, since the detector they tested selects always corner-like features. In contrast, Brooks *et al.* [24] observed some accuracy improvement in the fundamental matrix computation by incorporating the estimated covariance matrix of Harris corners. Steele and Jaynes [160] on the other hand focus on the detector and address the problem of feature inaccuracy based on pixel noise. They use several different noise models for pixel intensities and propagate the related covariances through the detection process of the Förstner-corner detector to come up with a covariance estimate for each feature point. Orguner and Gustafsson [141] evaluate the accuracy for Harris corner points. The analysis is built on the probability that pixels are the true corner in the region around the corner estimate. For scale-invariant region features, the covariance matrix has different properties from the corner features [187]: *First, due to the focus on interest regions, the shape of covariances will be in general anisotropic; second, the magnitude of covariances will vary significantly due to detection in scale space.*

Unlike the localization uncertainty, the localization precision is difficult to evaluate. The computation of the localization precision first requires knowing the ideal position of feature points, which is in fact ill-defined. Even if the ideal position of the feature points is known, it is not clear what an appropriate *statistical ensemble* would be, which could be deduced from an underlying noise model, a camera model or a model of lighting condition. So in practice, it is impossible to measure the localization precision. From another viewpoint, the absolute localization precision is less interesting in computer vision tasks where the correspondences are usually required. This motivated us to measure the matching precision instead of the localization precision. In fact, the matching precision reflects to some extent the localization precision. Given enough correspondences between two images, assume that all the feature points in one image are ideal and that the ground truth transformation between both images

is known, then the matching precision is close to the localization precision under the hypothesis that the feature detector has the same localization precision on all feature points in the other image and that the local properties of all the features composes an appropriate *statistical ensemble*. In reality feature points in both images are not ideal and their localization precision is different from point to point. So the matching precision measures the *average relative localization precision* of matchings between two images. Of course some inaccurate matchings or "outliers" can decrease the matching precision.

8.1.2 Organization

For scale-invariant feature detectors with sub-sampling like SIFT, the localization error and the localization uncertainty both increase through scales. We propose to cancel the sub-sampling to make images more blurred to gain some improvement. In theory, for continuous infinite resolution images, the localization precision will not be improved by canceling the sub-sampling. But in practice, only digital images can be used and the features can be more precisely located if the sub-sampling is canceled. The improvement can be observed through the matching precision, if there was no scale change between both images. We begin with a review of the SIFT method in section 8.2, followed by an improved SIFT scheme. Synthetic test of matching precision is shown in section 8.3 for Lowe's SIFT and improved SIFT with some comparison and analysis, followed by an iterative scheme to treat the case of scale change. Finally a local filtering technique is used to improved the matching precision for real images which are related by a homography plus a non-linear lens distortion. A brief conclusion is in section 8.4.

8.2 SIFT Method Review

The SIFT method [117] is one of the most widely used feature detectors. It is a good candidate for our analysis of multi-scale matching precision due to its scale invariance. The SIFT method is a complete algorithm, including scale-invariant feature detector, gradient-based descriptor construction and descriptor matching based on nearest neighbor distance ratio. The scale-invariant feature detector relies on 3D scale-space of normalized Laplacian implemented by difference of Gaussians due to its computational efficiency:

$$\begin{aligned} D(x,y,\sigma) &= \Big(G(x,y,k\sigma) - G(x,y,\sigma)\Big) \circledast I(x,y) \\ &\approx (k-1)\sigma^2 \Delta\Big(G(x,y,\sigma) \circledast I(x,y)\Big) \end{aligned} \tag{8.1}$$

$G(\cdot,\cdot,\sigma)$ is the Gaussian function with standard deviation σ and $\circledast$ is the convolution operation. Remark that the normalized Laplacian gives scale invariance to the Laplacian threshold in SIFT method. The SIFT method gives stable performance when k is smaller than $\sqrt{2}$. To simulate camera zoom, a 2-subsampling is also added in scale space. Several images of the same size compose one octave in scale space. The SIFT scale space consists of several octaves (Fig. 8.1): one octave contains $Ninter + 3$ Gaussian blurred images of the same size which are used to compute $Ninter + 2$ difference-of-Gaussian images. The local extrema are only detected on the $Ninter$ DoG images in the middle ($Ninter$ is the number of intervals in one octave, $Ninter = 3$ by default). The Gaussian blur is increased with the multiplicative factor

$2^{1/Ninter}$ and thus $k = 2^{1/Ninter}$. The 2-subsampling is performed on the image in octave which contains two times the blur of the initial image in the same octave. This convolution-subsampling procedure is repeated until the image is too small for feature detection. It is easy to see that the sampling with respect to the blur is the same for all octaves. So one image has the same nature as its counterparts in the other octaves. This process simulates camera zoom and explains why SIFT method is scale invariant.

In the 3D scale-space, features are selected as local extrema by comparing with 26 neighbors (see Fig. 8.1). Once a feature is extracted, its 3D position (position and scale) is refined by a 3D interpolation, where comes from SIFT sub-pixel precision. Each feature is assigned a principal direction by using the gradient information of the pixels in the neighborhood. A fixed-size (16×16 pixel) region around image feature along its principal direction is extracted to construct the descriptor. This region is divided into 4×4 sub-regions; in each sub-region, an orientation histogram containing 8 directions is created by quantizing the gradient direction of each sample weighted by gradient magnitude (Fig. 8.2). To make the detected features useful, their 3D coordinate (location and scale) should be propagated back to the original image.

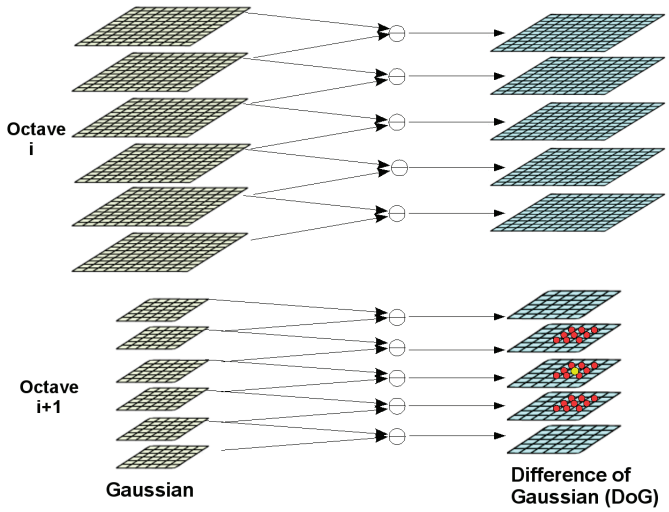


Figure 8.1: Pyramid-like SIFT scale space: a feature is selected as local extrema (yellow point) by comparing 26 neighboring samples (red point).

8.2.1 Blur

Blur is an important point because the scale invariance of the SIFT method is in fact achieved by simulating the blur under different resolution in scale space. The SIFT method is based on the assumption that a Gaussian convolution can well approximate the blur introduced by camera system and gives an aliasing-free image sub-sampling. In [132], it is proved that a well-sampled image should contain a Gaussian blur of standard deviation $\beta = 0.8$. Therefore an aliasing-free t-subsampling should be preceded by a Gaussian blur of about $\beta \times \sqrt{t^2 - 1}$. Here the Gaussian blur means the standard deviation of the Gaussian kernel.

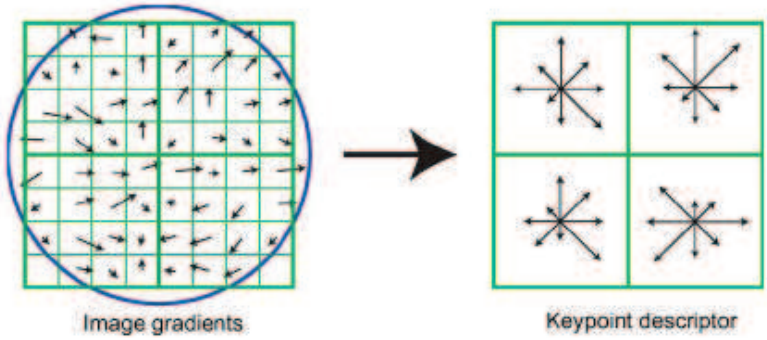


Figure 8.2: A descriptor is constructed on a square region around a feature point whose side direction is given by the principal gradient direction. Example of a 2×2 descriptor array of orientation histograms (right) computed from an 8×8 set of samples (left). The orientation histograms are quantized into 8 directions and the length of each arrow corresponds to the magnitude of the histogram entry.

The above discussion deals with aliasing-free sub-sampling. However, the blur conditions become different for the up-sampling case. This argument is based on the following simple equations:

$$\Delta\left(u(\frac{x}{2},\frac{y}{2})\right) = \frac{1}{4}(\Delta u)\left(\frac{x}{2},\frac{y}{2}\right) \tag{8.2}$$

$$\frac{\partial\left(u\left(\frac{x}{2},\frac{y}{2}\right)\right)}{\partial\,\bullet} = \frac{1}{2}\frac{\partial u}{\partial\,\bullet}\left(\frac{x}{2},\frac{y}{2}\right) \tag{8.3}$$

which means that the Laplacian is 4 times smaller and the gradient is 2 times smaller if an image is up-sampled by 2. Remark that the Laplacian and gradient are computed by finite difference schemes, but not by Eq. (8.1), which includes a scale normalization factor. This relation is only valid when the image u is smooth enough. Fig. 8.3 shows a test for a natural image. The image is first convolved by a Gaussian blur, then followed by an up-sampling by factor 2. The Laplacian and gradient module are computed on the original image and the up-sampled image respectively. Note m the ratio of Laplacian before and after 2-upsampling and n the ratio of gradient norm before and after 2-upsampling:

$$m = \frac{(\Delta u)\left(\frac{x}{2},\frac{y}{2}\right)}{\Delta\left(u(\frac{x}{2},\frac{y}{2})\right)} \tag{8.4}$$

$$n = \frac{\sqrt{\left(\frac{\partial u}{\partial x}(\frac{x}{2},\frac{y}{2})\right)^2 + \left(\frac{\partial u}{\partial y}(\frac{x}{2},\frac{y}{2})\right)^2}}{\sqrt{\left(\frac{\partial u(\frac{x}{2},\frac{y}{2})}{\partial x}\right)^2 + \left(\frac{\partial u(\frac{x}{2},\frac{y}{2})}{\partial y}\right)^2}} \tag{8.5}$$

By computing the average and standard deviation of m and n with respect to the added blur, it appears that Eq. (8.2) and (8.3) are satisfied only if the added Gaussian blur was bigger than 1.6. This makes the image blur equal to $\sqrt{1.6^2 + 0.8^2} \approx 1.8$ if the original image is assumed to already contain a blur 0.8. This experiment is complementary to the one dealing

with aliasing-free sub-sampling in [132]. Our conclusion is that a good image for feature analysis must contain at least a 1.8 Gaussian blur. Similarly, in Eq. (8.1), the difference of Gaussians can correctly approximate the scale normalized Laplacian only if the image $G(x, y, \sigma) \circledast I(x, y)$ contains a blur bigger than 1.8. In Lowe's SIFT, to increase the number of features, the input image is first pre-zoomed by 2. For an image containing initial Gaussian blur 0.8, a 2-upsampling increases the blur to be 1.6, which is close to 1.8, as required.

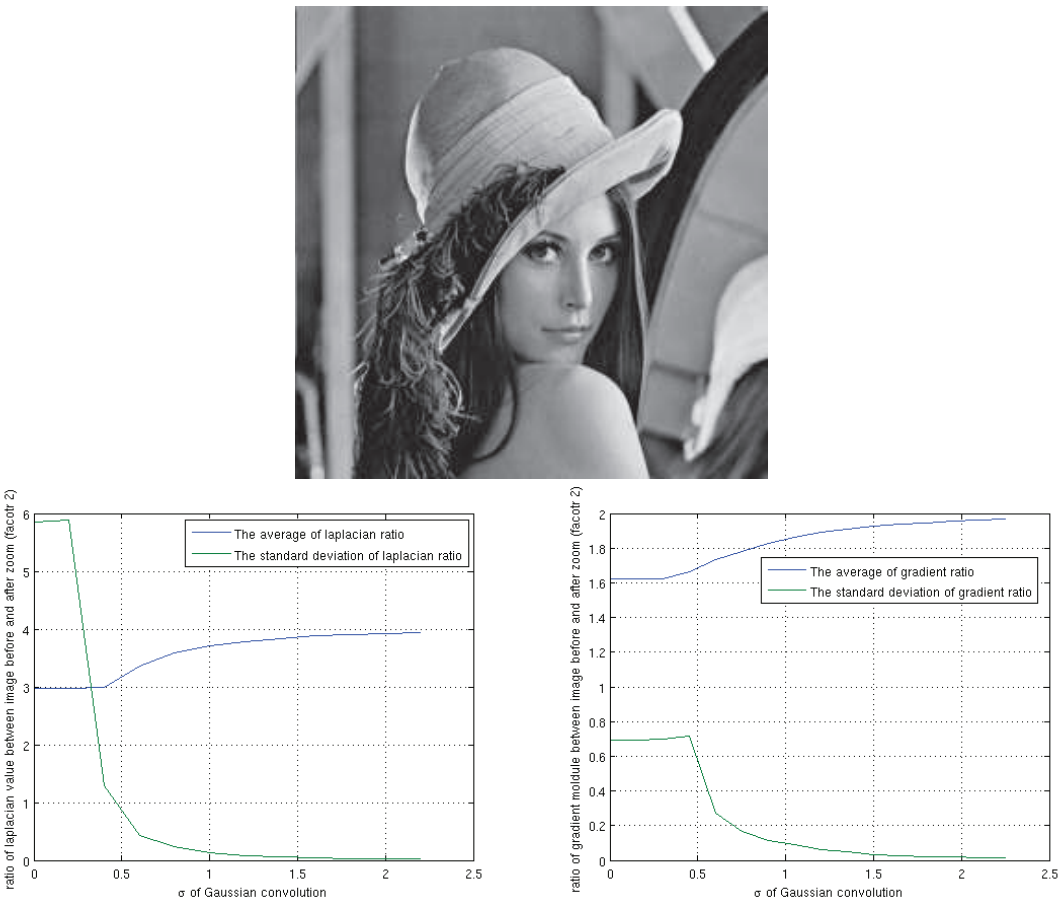


Figure 8.3: The top image is convolved by a Gaussian function with standard deviation σ before it is up-sampled by factor 2. The Laplacian value and gradient modulus before and after the up-sampling are compared. Bottom left: the average and standard deviation of ratio of the Laplacian value before and after 2-upsampling. Bottom right: the average and standard deviation of ratio of gradient modulus before and after 2-upsampling.

8.2.2 3D Localization Refinement

Once the local extrema are extracted in the 3D scale space, their position can be refined under the assumption that the image can be locally approximated by a second order Taylor expansion. Given a local extremum located at $\mathbf{x} = (x, y, \sigma)$, the DoG function $D(\mathbf{x})$ is

expanded at $\mathbf{x}$ by:

$$D(\mathbf{x}+\mathbf{\Delta x}) = D(\mathbf{x}) + \mathbf{\Delta x}^T \frac{\partial D}{\partial \mathbf{x}} + \mathbf{\Delta x}^T \frac{\partial^2 D}{\partial \mathbf{x}^2} \mathbf{\Delta x} \tag{8.6}$$

The peak of this function is attained when its derivative is set to be zero, which gives the offset $\mathbf{\Delta x} = (\Delta x, \Delta y, \Delta\sigma)^T$:

$$\mathbf{\Delta x} = (\Delta x, \Delta y, \Delta\sigma)^T = \left(\frac{\partial^2 D}{\partial \mathbf{x}^2}\right)^{-1} \frac{\partial D}{\partial \mathbf{x}} \tag{8.7}$$

Sub-pixel precision is thus obtained and the final position is $\mathbf{x} + \mathbf{\Delta x}$. The refined scale with respect to the original image resolution is $(\sigma + \Delta\sigma) \cdot 2^{oct}$ with $\sigma = \sigma_0 \cdot 2^{inter/Ninter}$ (σ_0 is the blur of the first image in one octave, *oct* is the octave index, *inter* is the interval index and $Ninter$ is the total number of intervals in one octave). This refinement can also be extended to other feature detectors because the method is relatively independent of the detector. The gradient and Hessian matrix in Eq. (8.6) is computed by a finite difference scheme from neighboring samples on the DoG images. A more robust method is to estimate the offset $\mathbf{\Delta x}$ by least square minimization [114].

8.2.3 Improvement

We remark that the principal error source in SIFT method is that the detected features in scale space are projected back to the original image. Assume a feature located at $\mathbf{x}$ with ideal position $\mathbf{x_0}$ disturbed by the error ε: $\mathbf{x} = \mathbf{x_0} + \varepsilon$. If this feature is detected in the i-th octave in SIFT scale space, then its final position is $2^i\mathbf{x} = 2^i\mathbf{x_0} + 2^i\varepsilon$. The error is increased by the factor 2^i. To gain accuracy, this suggests to cancel the sub-sampling between octaves. The new scheme is shown in Fig. 8.4. Although this seems to be an one-step modification to SIFT method, there are some details to discuss.

First the Laplacian threshold, which is the most important threshold in SIFT to select stable features, must be changed. Because of the scale normalized Laplacian computed as difference-of-Gaussian in Eq. (8.1), the threshold on the Laplacian can be kept constant through octaves. This is still true when the sub-sampling is canceled. This can be seen by upsampling a DoG function $D(x, y)$ at a certain scale σ by factor 2:

$$\begin{aligned} D(\tfrac{x}{2}, \tfrac{y}{2}, \sigma) &= \Big(G(\frac{x}{2}, \frac{y}{2}, k\sigma) - G(\frac{x}{2}, \frac{y}{2}, \sigma)\Big) \circledast I(\frac{x}{2}, \frac{y}{2}) \\ &= \Big(G(x, y, 2k\sigma) - G(x, y, 2\sigma)\Big) \circledast I(\frac{x}{2}, \frac{y}{2}) \\ &\approx (k-1)(2\sigma)^2 \Delta\Big(G(x, y, 2\sigma) \circledast I(\frac{x}{2}, \frac{y}{2})\Big) \\ &= (k-1)(2\sigma)^2 \Delta\Big(G(\frac{x}{2}, \frac{y}{2}, \sigma) \circledast I(\frac{x}{2}, \frac{y}{2})\Big) \\ &= (k-1)\frac{4}{m}\sigma^2\Big(\Delta(G \circledast I)\Big)(\frac{x}{2}, \frac{y}{2}) \end{aligned} \tag{8.8}$$

According to Eq. (8.2), $m = 4$ when the image $G \circledast I$ contains a blur bigger than 1.8. Then, $D(\frac{x}{2}, \frac{y}{2}, \sigma) = (k-1)\sigma^2\Big(\Delta(G \circledast I)\Big)(\frac{x}{2}, \frac{y}{2})$, which means that the Laplacian value does not change through octaves when the sub-sampling is canceled.

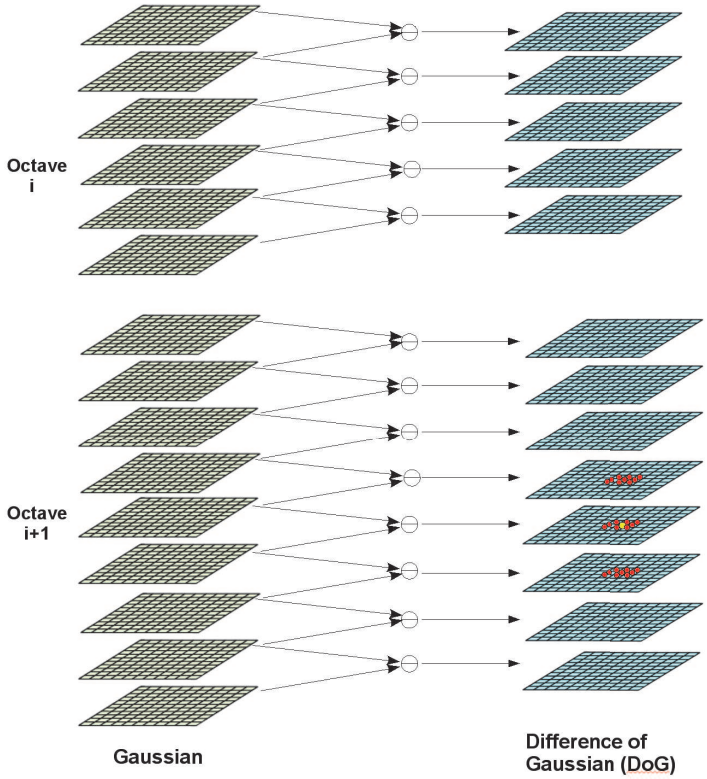


Figure 8.4: Improved SIFT scale space with sub-sampling canceled. The number of intervals in octave is increased through octaves.

Second, Lowe's SIFT descriptor is constructed from a region with fixed size 16×16 around extracted features. There is no problem if the scale change between both images is 2^i $(i \in \mathcal{N})$ because their scale spaces superpose up to an i octaves shift. So one feature in one image will find its correspondence in the other image at the same interval up to this i octaves shift. So any fixed-size descriptor capturing enough distinctive local information works. But if the scale changes between both images is $2^{i+\delta}$ $(i \in \mathcal{N}, 0 < \delta < 1)$, then one feature will not find its correspondence at the same interval up to an i octaves shift. In such a case, the fixed-size descriptor will not cover the same image region and thus can introduce some false matchings. By considering this default, we propose that the descriptor region have its size proportional to the Gaussian blur where the feature is actually detected. This increases the overlap of covered regions between correspondences when the scale change is $2^{i+\delta}$ (Fig. 8.5). On the other hand, if the sub-sampling is canceled, the size of descriptor becomes bigger with the increase of blur. And the sampling step becomes smaller and smaller with respect to blur. This is not consistent with scale invariance. To keep SIFT scale invariance, the new descriptor is sub-sampled again to make the sampling step proportional to blur, just like the original SIFT (see Fig. 8.6). Thus this new SIFT framework is still scale-invariant.

Figure 8.5: The right image is a $2^{2/3}$-subsampling of the left image. A feature in the right image with blur σ corresponds to a feature with $\sigma \times 2^{2/3}$ in the left image. The fixed-size descriptor gives evidently different patches for a common feature in both images (yellow patch in the left image via green patch in the right image). The two patches (in green) look almost the same by using a descriptor with size proportional to the blur.

Third, with increased blur through octaves, the step between two adjacent intervals in the scale direction increases also by factor 2, 4, 16, $\cdots$. Then the scale space is sampled more and more sparsely through octaves. This makes it more difficult for the 3D interpolation refinement to produce a precise result. In addition, the SIFT descriptor is constructed approximately on these sparse intervals without really interpolating a new interval. This makes descriptors less accurate. To compensate this effect, the number of intervals is increased with the same factor through octaves (see Fig. 8.4). This means that the up-sampling is performed also in the scale

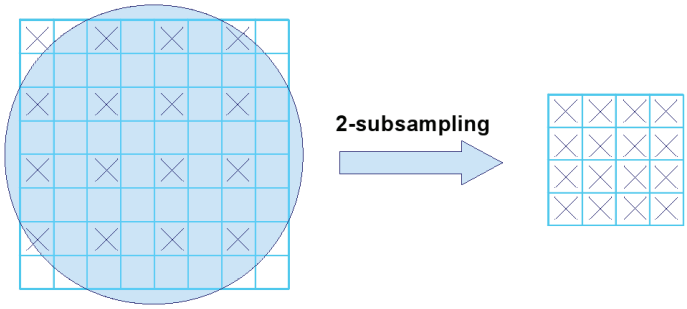


Figure 8.6: The SIFT descriptor is constructed by summarizing the gradient information of a region around the detected feature. This region has a fixed size in Lowe's SIFT. The sampling step of the region is proportional to the Gaussian blur. The gradient information is weighted by a Gaussian weighting function indicated by the overlaid circle. In the improved SIFT, the region has a size proportional to the blur and the sampling step is no longer proportional to the Gaussian blur. The descriptor region needs to be sub-sampled to maintain the scale invariance. A 2-subsampling is shown on the right.

direction, just as that in the x and y directions in image. Remark that this up-sampling in the scale direction still keeps the scale invariance of the Laplacian threshold.

For a sub-pixel refinement, the sub-sampling removal has a twofold effect. On the one hand, the assumption that the image is locally a 2-order polynomial is more valid with the increase of Gaussian blur. This makes the 3D interpolation more precise. On the other hand, the increase of Gaussian blur also makes it more difficult to localize features. There is no back-projection since the sub-sampling is canceled. Are these two factors completely compensated by each other? Theoretically the answer is yes. Given a local extremum $\mathbf{x} = (x_0, y_0, \sigma_0)$, its offset is given by Eq. (8.7): $\mathbf{\Delta x} = \left(\frac{\partial^2 D}{\partial \mathbf{x}^2}\right)^{-1} \frac{\partial D}{\partial \mathbf{x}}$. If the factor to back project to the original resolution is 2, the final offset is $2\mathbf{\Delta x}$. If the error is directly computed on 2-upsampled image, by Eq. (8.4), (8.5), (8.7) and (8.8), it gives: $\left(\frac{4}{m^2}\frac{\partial^2 D}{\partial \mathbf{x}^2}\right)^{-1} \frac{4}{mn}\frac{\partial D}{\partial \mathbf{x}} = \frac{m}{n}\left(\frac{\partial^2 D}{\partial \mathbf{x}^2}\right)^{-1} \frac{\partial D}{\partial \mathbf{x}} = \frac{m\mathbf{\Delta x}}{n}$. According to Fig. 8.3, by adding a Gaussian blur larger than 1.6, $\frac{m}{n}$ is about 2 and the variation is ignorable. Then the offset is about $2\mathbf{\Delta x}$, which means that no improvement in precision is obtained by canceling the sub-sampling between octaves. But in practice, we do gain something due to the fact that the used image is digital. First, the local extrema at integer pixel position are more precise in the up-sampled image, which means that the 3D refinement has a better departure point. Second, the assumption that the image is locally 2-order is more valid with the increase of Gaussian blur. So the gradient and the Hessian computation are more precise on the up-sampled images. Remark that if no blur is added to the image, the average of $\frac{m}{n}$ is also about 2, but the variance is rather high. Thus the computed offset is not reliable.

8.3 Matching Precision Evaluation

The matching precision of Lowe's SIFT method is evaluated on a pair of images in this section. We first review different matching precision evaluation procedures and point out their drawbacks. We compute the matching precision more directly under different geometric transformations by average residual error.

8.3.1 Evaluation Method

The most popular evaluation criterion is *repeatability* introduced in [128, 155], and defined as the ratio between the number of correspondences and the minimum number of points detected in both images. Feature x_a and x_b correspond if:

- the error in relative point location is less than ϵ_p pixels: $x_a - H \cdot x_b < \epsilon_p$, where ϵ_p is typically 1.5 pixels and H is the ground truth homography between two images;
- $1 - \frac{R_{\mu_a} \cap R_{H^T \mu_b H}}{R_{\mu_a} \cup R_{H^T \mu_b H}} < \epsilon_o$ where R_μ the detected region determined by the shape matrix μ given by affine invariant interest point detector [128]; $H^T \mu_b H$ is shape matrix projected to the other image; $R_{\mu_a} \cap R_{H^T \mu_b H}$ is the intersection of regions and $R_{\mu_a} \cup R_{H^T \mu_b H}$ is their union.

We argue that "repeatability" is not well adapted to the matching precision due to the following reasons:

- It is not sure that x_a and x_b is a good correspondence if the two above criteria are satisfied. In fact the above two measures depend on the detected scale. So it is not easy to find a universal threshold.
- It is assumed that the scene is planar and that the ground truth homography between both images is estimated from some control points. But in practice, we cannot have a completely planar scene. Even if so, the camera lens would introduce some non-linear distortion. Thus, a pair of real images are always related by a homography up to some error. This means that the ground truth itself is problematic.
- The features detected in different scales do not have the same precision. So it is better to do the evaluation in different octaves separately.
- More precise matchings can be obtained by using a Ransac-based algorithm.

Some other criteria not requiring the exact position of interest points exist. In [40, 93], authors used projective invariants to evaluate interest point precision. But this kind of method needs a scene composed of simple geometric objects, like polygons or polyhedrons, since the reference values are computed from scene measurements. In [143], the authors used four different criteria: alignment of the extracted points, accuracy of the 3D reconstruction, accuracy of the epipolar geometry and stability of the cross-ratio. In [8], four other global criteria are proposed: collinearity, intersection at a single point, parallelism and localization on an ellipse.

None of the above methods consider lens distortion or other optic system aberrations in their precision evaluation procedure. For the methods requiring scenes composed of simple geometric objects, they are designed for some specific model-based interest point detectors and not very adequate for SIFT points. Moreover, since the reference measures are from the scene, the objects should be constructed with high precision, which is difficult in practice.

Due to the above problems, we shall evaluate the matching precision in a more direct way. Synthetic images are used in the test to avoid any disturbance like lens distortion. A transformation is applied on a reference image to obtain the second image. SIFT is applied on both images to extract feature points. The SIFT descriptors are matched by nearest neighbor distance ratio. A Ransac-like parameter-free algorithm [129] is applied to eliminate false matchings. A global homography is computed from the Ransac-verified matchings by using a least-square method. The matching precision is evaluated as the average residual error to this homography.

8.3.2 Tests

The tests are performed for six types of transformation: translation, rotation, zoom, tilt, affine transformation and homography. The pre-zoom in SIFT method makes the blur of the initial image about 1.6. The number of octaves is fixed to be four. The proposed evaluation procedure is used and the error is computed respectively on different octaves. Remark that the number of features decreases through octaves. There can be not enough matchings on the third or fourth octave and the evaluation is not very reliable, in particular in case of tilt, affine transformation and homography, since SIFT is only similarity invariant.

Translation

Translation is the simplest case to test. For an integer pixel translation, the evaluation always gives zero error. This is because both images are identical same up to an integer pixel translation. Errors are observed when the translation is not an integer. The reason can be twofold: first, if the reference image itself is a little aliased, the generated second image will also contain some artifacts; second, the feature position refinement is performed by a 3D interpolation in SIFT, which is based on the assumption that the image can be locally approximated by second-order Taylor expansion. Thus the performance of the 3D refinement depends on the image regularity around local extrema. In addition, the implementation of sub-pixel translation can also pose a problem: Fourier interpolation is exact but introduces "ringing", namely artifacts near the contours and image borders, while a spline interpolation is just an approximation to the Fourier interpolation. As a compromise, a 7-order spline interpolation was used (see images in Fig. 8.7). Even though this test is very simple, it gives us the idea about the best matching precision that SIFT method can achieve. The average matching error is shown in Fig. 8.10a. Remark that the integer translation $(45, 32)$ gives very small error on the first two octaves, but on the other octaves, errors are observed because the images are sub-sampled and the translation becomes non-integer. In fact, the matching error should increase by 2 from one octave to the next due to back projection to the original image. But in the translation case, the matching error is largely dominated by the quantity of translation: the matching error is smaller when the translation is close to an integer. This explains why the error increases by a varying factor.

Rotation

Rotation is another basic geometric transform in image processing. An exact implementation by Fourier is feasible by decomposing a rotation into three shear transforms. But it suffers also from "ringing" artifacts, like for the translation case. Thus a 7-order spline interpolation is used again (see images in Fig. 8.7). As the evaluation shows in Fig. 8.10b, the closer to 0° or 90° the rotation, the smaller the matching error. This is due to the imprecision in orientation estimation of features in SIFT method and the interpolation introduces less artifacts. In the rotation case, the error is dominated by the back projection. So the matching error increases by a factor approximately 2 from one octave to the next.

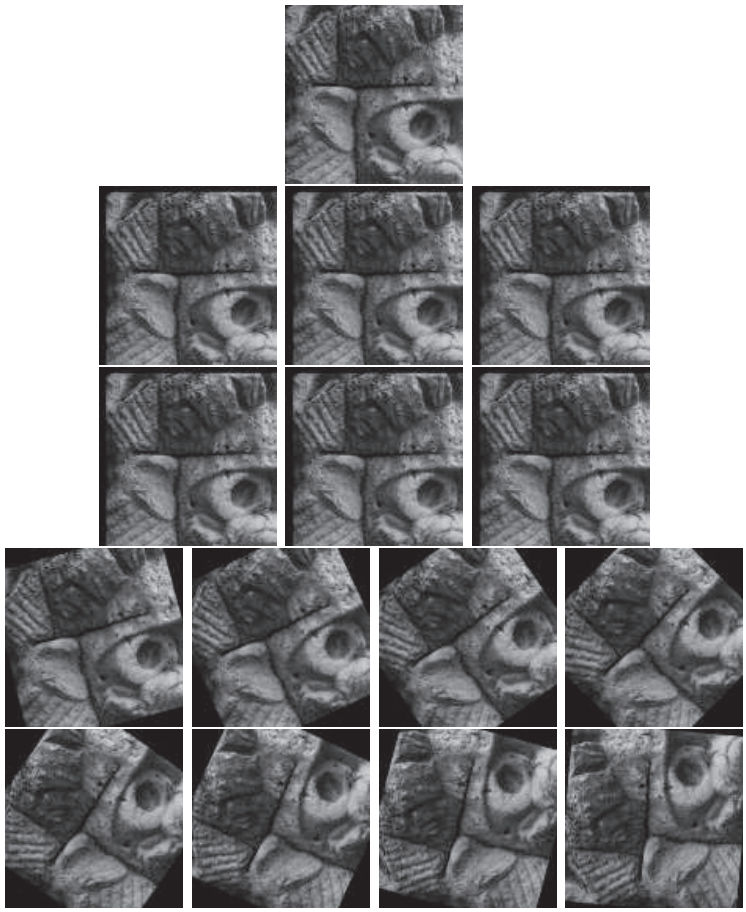


Figure 8.7: First row: original image. Second and third row: image translated by $(45, 32)$, $(45.1, 32.1)$, $(45.3, 32.3)$, $(45.5, 32.5)$, $(45.7, 32.7)$, $(45.9, 32.9)$. Fourth and fifth row: image rotated by 15°, 25°, 35°, 45°, 55°, 65°, 75°, 85°.

Zoom

In the case of zooms, the second image is generated by blurring the reference image with $\sqrt{t^2 - 1} \times 0.8$ followed by a t-subsampling where t is the scale change between two images (Fig. 8.8).

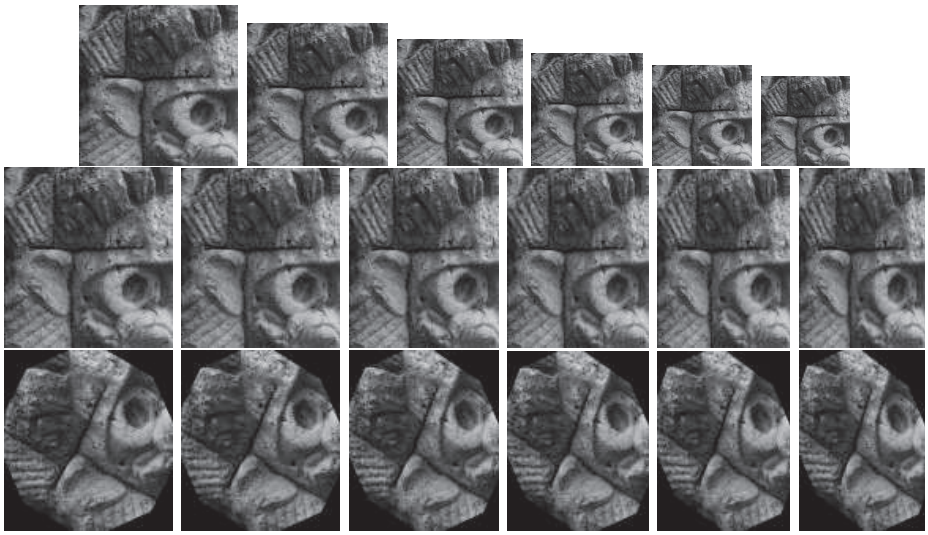


Figure 8.8: First row: image sub-sampled by factor $2^{1/6}$, $2^{2/6}$, $2^{3/6}$, $2^{4/6}$, $2^{5/6}$, 2. Second row: image transformed by tilt with t equal to $2^{1/12}$, $2^{2/12}$, $2^{3/12}$, $2^{4/12}$, $2^{5/12}$, $2^{6/12}$. Third row: image transformed by affine transformation A in Eq. (8.9) with $\phi = 37°$, $\psi = 24°$ and t equal to $2^{1/12}$, $2^{2/12}$, $2^{3/12}$, $2^{4/12}$, $2^{5/12}$, $2^{6/12}$.

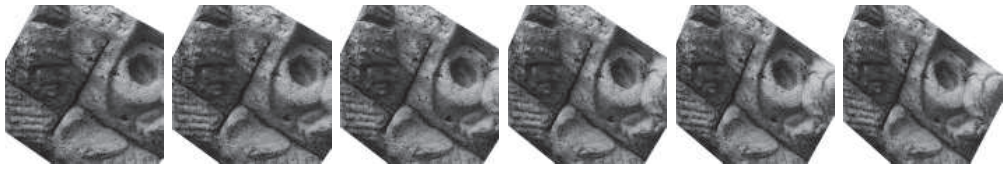


Figure 8.9: Images transformed by homography. The homography is obtained by adding a projective component to the affine transformation matrix in Eq. (8.9) (see Eq. (8.10)).

It is more difficult to deal with zoom than with rotation and translation. Even though theoretically SIFT is scale invariant, in practice the scale invariance is disturbed by the blur inconsistency caused by the scale quantization in the SIFT scale space (see Fig. 8.1). To be clearer, we use the following lemma in [132].

Lemma 2 *Let u and v be two digital images that are frontal snapshots of the same continuous flat image $\mathbf{u}_0$, $u := \mathbf{S_1}\mathbf{G}_\beta\mathbf{H}_\lambda\mathbf{u}_0$ and $v =: \mathbf{S_1}\mathbf{G}_\delta\mathbf{H}_\mu\mathbf{u}_0$, taken at different distances, with different Gaussian blurs and possibly different sampling rates. Let $\mathbf{w}(\sigma, \mathbf{x}) = (\mathbf{G}_\sigma\mathbf{u}_0)(\mathbf{x})$ denote the scale space of $\mathbf{u}_0$. Then the scale spaces of u and v are*

$$\mathbf{u}(\sigma, \mathbf{x}) = \mathbf{w}(\lambda\sqrt{\sigma^2 + \beta^2}, \lambda\mathbf{x}) \text{ and } \mathbf{v}(\sigma, \mathbf{x}) = \mathbf{w}(\mu\sqrt{\sigma^2 + \delta^2}, \mu\mathbf{x})$$

If $(s_0, \mathbf{x_0})$ is a key point of $\mathbf{w}$ satisfying $s_0 \geq max(\lambda\beta, \mu\delta)$, then it corresponds to a key point of $\mathbf{u}$ at the scale σ_1 such that $s_0 = \lambda\sqrt{\sigma_1^2 + \beta^2}$, whose SIFT descriptor is sampled with mesh $\sqrt{\sigma_1^2 + \beta^2}$. In the same way (s_0, x_0) corresponds to a key point of $\mathbf{v}$ at scale σ_2 such that $s_0 = \mu\sqrt{\sigma_2^2 + \delta^2}$, whose SIFT descriptor is sampled with mesh $\sqrt{\sigma_2^2 + \delta^2}$.

$\mathbf{S_1}$ is the 1-sampling operation applied on continuous image to obtain a digital image; $\mathbf{H}_\lambda$ is sub-sampling of factor λ; $\mathbf{G}_\beta$ is the Gaussian convolution with standard deviation β. $\mathbf{G}_\beta$ and $\mathbf{G}_\delta$ are the camera blurs applied on the infinite resolution image (blur free) before 1-sampling to avoid aliasing. The above lemma is proved in the continuous setting under the assumption that the Gaussian blur performed by the SIFT method approximates well the camera blur and gives aliasing-free images.

This lemma is easier to understand by an example with $\mu/\lambda = 2$. Then $s_0 = \sqrt{\sigma_1^2 + \beta^2} = 2\sqrt{\sigma_2^2 + \delta^2}$. Assume a pair of correspondences: $f_\mathbf{u}$ in $\mathbf{u}$ and $f_\mathbf{v}$ in $\mathbf{v}$. $f_\mathbf{u}$ lies on a scale two times coarser than $f_\mathbf{v}$, and $f_\mathbf{u}$'s descriptor has a sampling step twice bigger than $f_\mathbf{v}$'s descriptor. The SIFT dyadic scale space structure is adaptive to this case: $\mathbf{u}$ and $\mathbf{v}$ ideally superposes by one octave shift. This is true for all cases with $\mu/\lambda = 2^i, i \in \mathcal{N}$. So the localization error is zero up to the machine precision. But the situation becomes more complicated when $\mu/\lambda \neq 2^i$. Assume $\mu/\lambda = 2^{i+\epsilon}, i \in \mathcal{N}, 0 < \epsilon < 1$, then $\mathbf{u}$ and $\mathbf{v}$ never superpose in scale space due to the dyadic structure of the SIFT scale space.

A special case is when $\epsilon = s/Ninter, s = 1, \cdots, Ninter - 1$ ($Ninter$ is the number of intervals in one octave). Now assume $i = 0$, $Ninter = 3$ and $s = 1$, then $\mu/\lambda = 2^{1/3}$, then $f_\mathbf{u}$ lies on the scale $2^{1/3}$ times blurred than $f_\mathbf{v}$. This coincides with the fact that one SIFT octave is divided into $Ninter = 3$ intervals (Fig. 8.1). For $f_\mathbf{u}$ on interval s of octave o, $f_\mathbf{v}$ is on interval $s + 1$ of octave o containing blur $2^{1/3}$ bigger. But the problem occurs when the 3D refinement (in space and scale) is performed on local extrema. The 3D refinement being sensitive to blur, the precision of the refined position and scale for $f_\mathbf{u}$ and $f_\mathbf{v}$ will be different. So the matching error, which can be interpreted to some extent as an average relative localization precision, is higher than that in the case of a translation and rotation. Moreover, the refined scale is used to decide the size of region for the descriptor construction. The discrepancy in the precision of the refined scale for $f_\mathbf{u}$ and $f_\mathbf{v}$ can also cause problem in the descriptor matching. The false matchings can possibly be introduced in the following situation: if there exists a feature $f'_\mathbf{u}$ very close to $f_\mathbf{u}$, then a small error in the descriptor can make $f_\mathbf{v}$ matched by $f'_\mathbf{u}$, instead of $f_\mathbf{u}$. Since this kind of error is particularly small, the Ransac-like algorithm does not guarantee a removal of this kind of false matching.

The most general case occurs when μ/λ is any value. In such a case, no feature extracted in the scale space is a good candidate for matching because the blurs can never be equal: $\lambda\sqrt{\sigma_1^2+\beta^2} \neq \mu\sqrt{\sigma_2^2+\delta^2}$. More precision in the 3D refinement (position and blur) is required. But even with the correct blur ($\lambda\sqrt{\sigma_1^2+\beta^2} = \mu\sqrt{\sigma_2^2+\delta^2}$), the descriptor is always constructed from the region extracted in the closest interval (instead of interpolating a region). In addition, the blur is only used to determine the size of the descriptor region. Yet, the region is not sampled on the mesh mentioned in Lemma 2. This means that the descriptor is just approximative and can lead to false matchings.

The above problem also implies that in the SIFT method the initial blur (like β and δ) in image affects the SIFT matching performance. The Gaussian convolution performed by SIFT in the scale space is based on the initial blur. With an incorrect value of the initial blur, the resulting images in the scale space are either too much, or not enough blurred. This can make the discrepancy in the precision of the 3D refinement bigger. The initial Gaussian blur in SIFT is set to be 0.8 for input images. But not all of the natural images contain a Gaussian blur of 0.8.

The evaluation is shown in Fig. 8.10c. The case of $\mu/\lambda = 2$ gives the best result except in octave -1, where the reference image should find no matching in the second image. The false matchings are kept because the Ransac-based algorithm [129] is not designed for high precision evaluation. The case of $\mu/\lambda = 2^{2/6}$ and $2^{4/6}$ gives a more precise result than the case of $\mu/\lambda = 2^{1/6}$, $2^{3/6}$ and $2^{5/6}$ because one SIFT octave is divided into 3 intervals. Since the back projection is not the only error source, the matching error does not increase through octave with a factor of 2.

Affine Transformation

The affine transformation A can be applied on the reference image $\mathbf{u}$ to obtain the second image $\mathbf{v}$ by decomposing it by SVD (Singular Value Decomposition, see Appendix A.2):

$$A = H_\lambda R_1(\psi) T_t R_2(\phi) = \lambda \begin{pmatrix} \cos\psi & -\sin\psi \\ \sin\psi & \cos\psi \end{pmatrix} \begin{pmatrix} t & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \cos\phi & -\sin\phi \\ \sin\phi & \cos\phi \end{pmatrix} \tag{8.9}$$

with $R_1(\psi)$ and $R_2(\phi)$ rotation matrices, T_t a tilt and H_λ an expansion of λ. Rotation and zoom have been already analyzed before, the only new element here is the tilt. Without loss of generality, assume $t < 1$, meaning that the tilt T_t sub-samples the image by factor t in the x-direction without changing the resolution in the y-direction (see images in Fig. 8.8). The tilt is not consistent with the Gaussian blur performed in the SIFT scale space because the Gaussian blur is isotropic while the tilt isn't. More precisely, for a pair of correspondences $f_\mathbf{u}$ and $f_\mathbf{v}$, to attain scale invariance, on the one hand, in y-direction, $f_\mathbf{u}$ should be on the scale $\frac{1}{t}$ times coarser than $f_\mathbf{v}$; on the other hand, in the x-direction, $f_\mathbf{u}$ should be on the same scale as $f_\mathbf{v}$. This means that SIFT is not designed to be invariant for tilt or affine transformation. So the matching error attached to an affine transformation or to a tilt is larger than that of zoom and rotation. In Fig. 8.10d, a pure tilt transformation is evaluated with t varying from $2^{1/12}$ to $2^{6/12}$. The matching error increases with the increase of t and octave index. In Fig. 8.10e, an affine transformation is tested with $\phi = 37°$, $\psi = 24°$ and t varying from $2^{1/12}$ to $2^{6/12}$. Remark that in case of tilt and affine transformation, fewer matchings are found in coarse octaves. So the matching precision evaluation can be less reliable.

Homography

A homography can describe any transformation between two images of a planar scene taken by an ideal pinhole camera. A homography $H : (x, y) \to (X, Y) = (F_1(x, y), F_2(x, y))$ can be locally approximated by an affine transformation $A(x_0, y_0)$ around each point $(x_0, y_0) \to (X_0, Y_0)$ with 1-order Taylor expansion:

$$\begin{pmatrix} X - X_0 \\ Y - Y_0 \end{pmatrix} = \begin{pmatrix} \frac{\partial F_1}{\partial x}(x_0, y_0) & \frac{\partial F_1}{\partial y}(x_0, y_0) \\ \frac{\partial F_2}{\partial x}(x_0, y_0) & \frac{\partial F_2}{\partial y}(x_0, y_0) \end{pmatrix} \begin{pmatrix} x - x_0 \\ y - y_0 \end{pmatrix} + O \begin{pmatrix} (x - x_0)^2 + (y - y_0)^2 \\ (x - x_0)^2 + (y - y_0)^2 \end{pmatrix}$$

At point (i, j), the local zoom factor in two orthogonal directions $t_1(i, j)$, $t_2(i, j)$ can be computed by decomposing the corresponding affine transformation $A(i, j)$ like Eq. (8.9). If $t_1(i, j)$ and $t_2(i, j)$ are both bigger than 1 for all (i, j), the image is compressed nowhere and H can be directly applied on the image. If $t_1(i, j)$ (or $t_2(i, j)$) is smaller than 1, then there is a sub-sampling in the neighborhood around (i, j) along the corresponding direction. In such a case, a pre-zoom of factor $\frac{1}{t_1}$ (or $\frac{1}{t_2}$) is needed around (i, j) along the corresponding direction before applying H to avoid aliasing. But this point-wise pre-zoom is not feasible since $t_1(i, j)$ (or $t_2(i, j)$) differs from point to point. So the only solution is to first compute $t = \min_{(i,j)}(t_1(i, j), t_2(i, j))$. If $t < 1$, a global pre-zoom $\frac{1}{t}$ is applied on the image. With this adapted anti-aliasing pre-zoom, H can be safely applied on an image. Afterwards, to cancel the pre-zoom, we should again do a zoom-out with the same factor t. As always, a Gaussian blur $0.8 \times \sqrt{\frac{1}{t^2} - 1}$ is required before sub-sampling. The algorithm is recapitulated in Algorithm 2. It can be proven that it is enough to evaluate $t = \min_{(i,j)}(t_1(i, j), t_2(i, j))$ on the four image corners. The resulting image will be a little blurred since the Gaussian blur applied is adapted to the biggest local sub-sampling. In our experiments, the homography matrix is generated by adding a projective component to the affine transformation matrix A used in the previous tests:

$$H = \left[\begin{array}{cc|c} & \mathrm{A} & \begin{matrix}0\\0\end{matrix} \\ \hline -0.0001 & -0.0001 & 1 \end{array}\right]. \tag{8.10}$$

The images in Fig. 8.9 are generated by applying the homography matrix with Algorithm 2. The matching error is shown in Fig. 8.10f.

Algorithm 2 (Anti-aliasing homography)

Input: *image* $\mathbf{I}$, *homography* $\mathbf{H}$
Output: *image* $g(\mathbf{I}, \mathbf{H}) = \mathbf{I} \circ \mathbf{H}^{-1}$

At each corner of $\mathbf{I}$ *compute the Jacobian* $\mathbf{J}$ *of* $\mathbf{H}$ *and the SVD of* $\mathbf{J}$. *Take* t *the smallest among these 8 singular values.*

Let $\mathbf{S} = \begin{pmatrix} s & 0 & 0 \\ 0 & s & 0 \\ 0 & 0 & 1 \end{pmatrix}$ *with* $s = max(\frac{1}{t}, 1)$.

if $t(\mathbf{H}) < 1$ **then**

  $\mathbf{I} = g(\mathbf{I}, \mathbf{SH})$;

  Convolve $\mathbf{I}$ *with the Gaussian kernel of standard deviation* $0.8 \times \sqrt{\frac{1}{t^2} - 1}$;

  Replace $\mathbf{H}$ *by the zoom-out matrix* $\mathbf{S}^{-1} = \begin{pmatrix} t & 0 & 0 \\ 0 & t & 0 \\ 0 & 0 & 1 \end{pmatrix}$

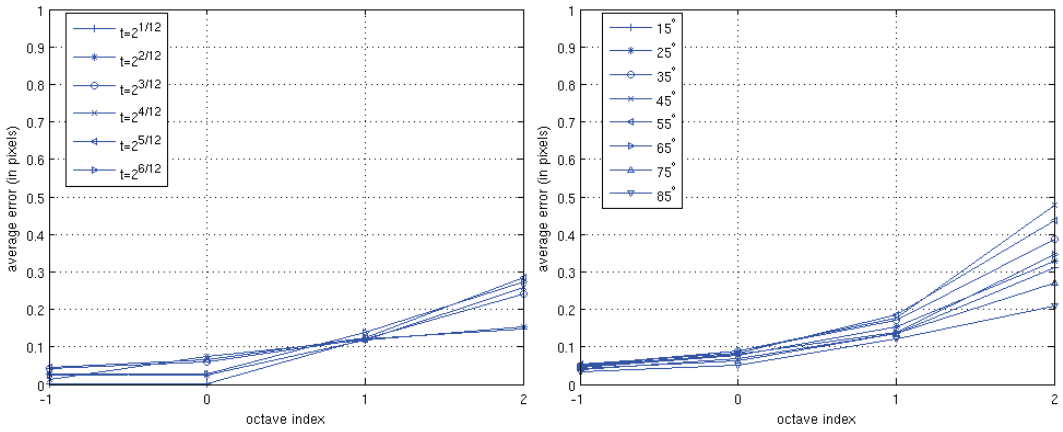
end

Return image $\mathbf{I} \circ \mathbf{H}^{-1}$, *computed by Fourier interpolation or other high-order spline interpolation.*

Conclusion

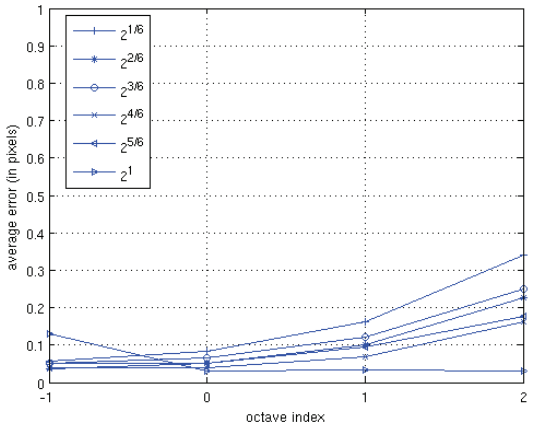
Systematic tests have been performed to evaluate the SIFT matching precision. As we have explained, the matching precision reflects to some extent the *average relative localization precision* between the compared images. The localization precision is mostly determined on the local extrema extraction and the performance of the 3D refinement in the feature point position and scale. Of course, the false matchings caused by inaccurate descriptors can also decrease the matching precision. Blur is a key point in SIFT. It affects not only the performance of the 3D refinement, but also the descriptor construction. The blur value is correct only when the initial image blur is well estimated and the interpolation in scale direction is well performed.

The tests can be divided into two groups. One group is without scale change: translation and rotation. The other with scale change: zoom, tilt, affine transformation and homography. The group without scale change suffers less from the blur inconsistency than the group with scale change because scale changes lead to different performance in 3D interpolation for a feature and its correspondences. For all transformations, the error increases with octaves. This is normal because all features are finally projected back to the original image. For translations and rotations, the localization precision is quite similar for two compared images without scale change. So the matching error is relatively small. Rotations cause a matching error larger than for translations because the rotation case suffers also from the imprecise estimation of the principal orientation of features. We think that in the translation and rotation case, the matching precision is the best precision the SIFT method can achieve. For tilts, affine transformations and homographies, the localization precision of a feature point is different from that of its correspondences due to the blur inconsistency caused by the coarse scale quantization in SIFT scale space. This is the main reason explaining why the transformation with scale changes has a larger matching error. In addition, the matching

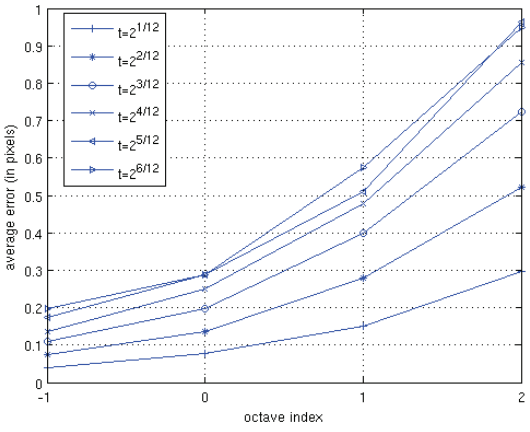


(a) translation

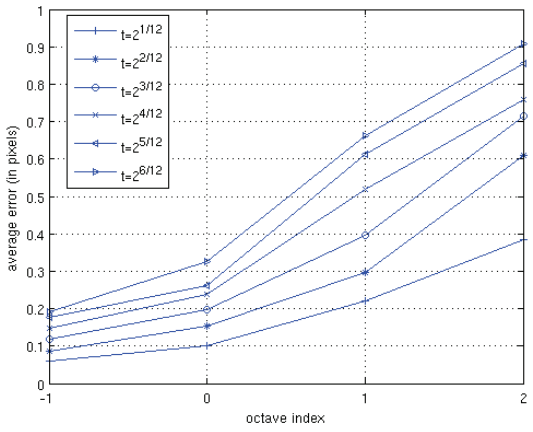
(b) rotation



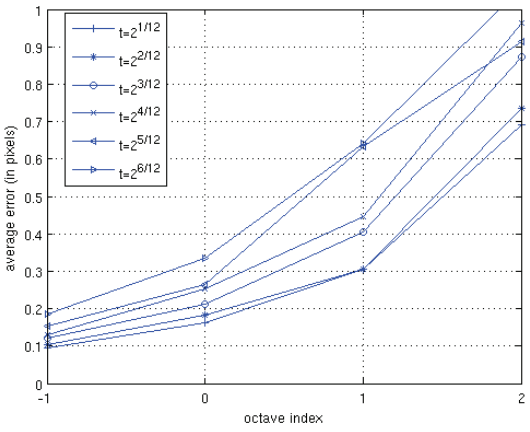
(c) zoom



(d) tilt



(e) affine transformation



(f) homography

Figure 8.10: The average matching error of Lowe's SIFT under different geometric transformations. (a) translation: the translation in x and y direction is $(45, 32)$, $(45.1, 32.1)$, $(45.3, 32.3)$, $(45.5, 32.5)$, $(45.7, 32.7)$ and $(45.9, 32.9)$. (b) rotation: the rotation angle varies from 15° to 85° with the step of 10°. (c) zoom: the zoom factor is $2^{1/6}$, $2^{2/6}$, $2^{3/6}$, $2^{4/6}$, $2^{5/6}$ and $2^{6/6}$. (d) tilt: Eq (8.9) with $\phi = 0°$, $\psi = 0°$, t is $2^{1/12}$, $2^{2/12}$, $2^{3/12}$, $2^{4/12}$, $2^{5/12}$, $2^{6/12}$. (e) affine transformation: Eq (8.9) with $\phi = 37°$, $\psi = 24°$ and t is $2^{1/12}$, $2^{2/12}$, $2^{3/12}$, $2^{4/12}$, $2^{5/12}$ and $2^{6/12}$. (f) homography: Eq. (8.10) by adding a projective component in the affine transformation. The x-axis is the octave index, from -1 to 2. The range of y-axis is from 0 pixels to 1.0 pixels.

error becomes larger when there are not enough matchings in octaves 1 and 2, due to SIFT's partial invariance to affine transformation.

8.3.3 Improvement

In this section, the improvement in section 8.2.3 is evaluated. This scheme gives a smaller matching error than Lowe's SIFT in translation and rotation case (Fig. 8.11a and 8.11b) because the back projection is removed and the precision of 3D refinement is higher on more blurred images. But for the zoom, tilt, affine transformation and homography which contain a scale change, the matching error incurred by blur inconsistency plays a more important role and prevails over the gain by canceling the sub-sampling. So the improvement in precision is limited compared with Lowe's SIFT (Fig. 8.11c, 8.11d, 8.11e, 8.11f).

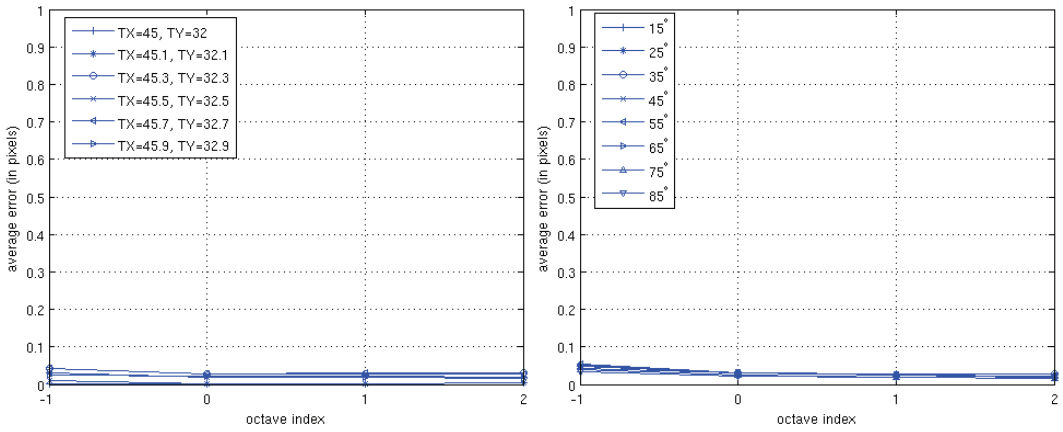
Applying a homography covers all the tested transformations. Even though the precision cannot be evidently improved if there is a scale change between both images, the homography between two images can be estimated by using the most precise matchings in the first octave. In the experiments, the homography from the reference image to the second image, which shrinks the size of image, was estimated. This homography can then be applied on the reference image to compensate the scale change. The SIFT method can be again applied on both images to detect and match features. But the matching precision is finally computed in the original image space. Since there is no longer a scale change between the two images, the improved SIFT scheme can reach a higher precision. Fig. 8.12 shows the matching precision of this iterative scheme on zoom, tilt, affine transformation and homography.

8.3.4 A More Realistic Algorithm

We have seen that the scale changes are a challenge for the SIFT method if precise matchings are at stake. The improved SIFT scheme can hardly improve the precision, while the iterative scheme used in section 8.2.3 improves the precision only when the underlying transformation is a homography. In practice, the transformation between two images can be more complicated than a homography by considering the 3D effect of scene and camera lens distortion. So the iterative scheme is not very useful for real images. This leads us to find a more realistic algorithm.

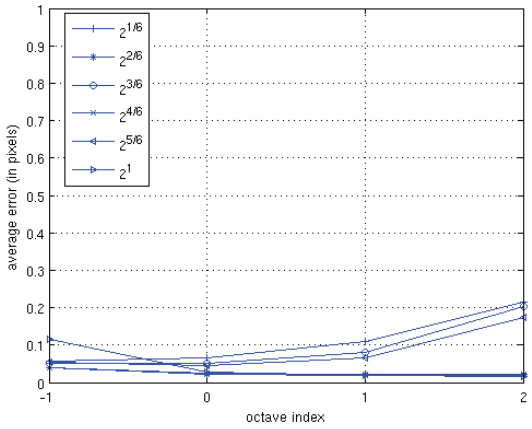
Here we concentrate on the cases where the transformation between two images is smooth. Although this does not apply when the scene is really 3D, there are already many applications based on this assumption, like super-resolution, deformable image registration, camera calibration, etc.

The idea comes from our work about non-parametric lens distoriton correction [80]. In this work, dense SIFT matchings between a digital textured pattern and its photo are obtained to interpolate the distortion field. This field can then be reversely applied on any other photos taken by the same camera to correct the distortion. The improved SIFT scheme cannot give matchings with average error smaller than 0.1 pixels. This precision does not estimate a sufficiently precise distortion field to rectify the distorted lines. In fact, the zig-zag effect is observed on the corrected lines. To further increase the SIFT matchings precision, a local filter is used. This is based on the assumption that any smooth transformation can be locally approximated by a homography. If the SIFT matchings densely and uniformly distribute over all of the domain of image, the point-wise local homography can be estimated by using

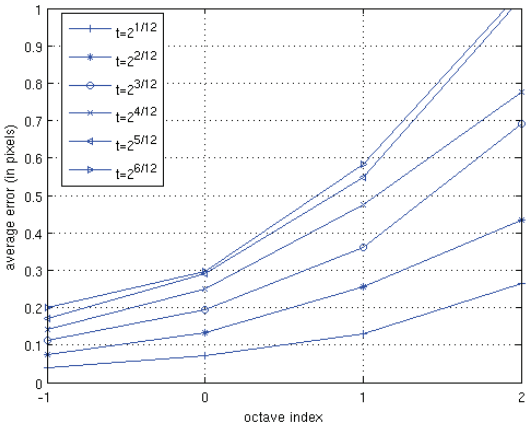


(a) translation

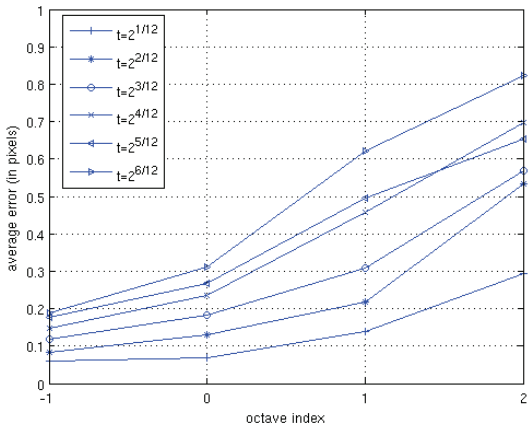
(b) rotation



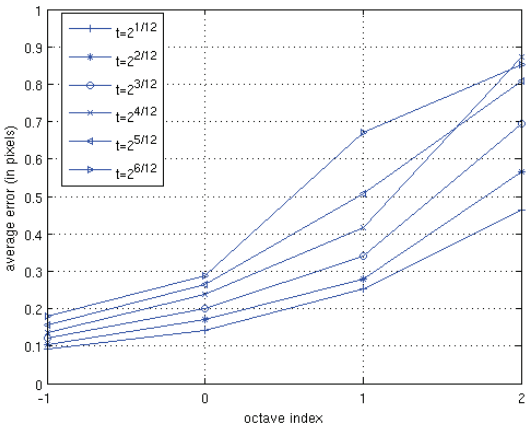
(c) zoom



(d) tilt

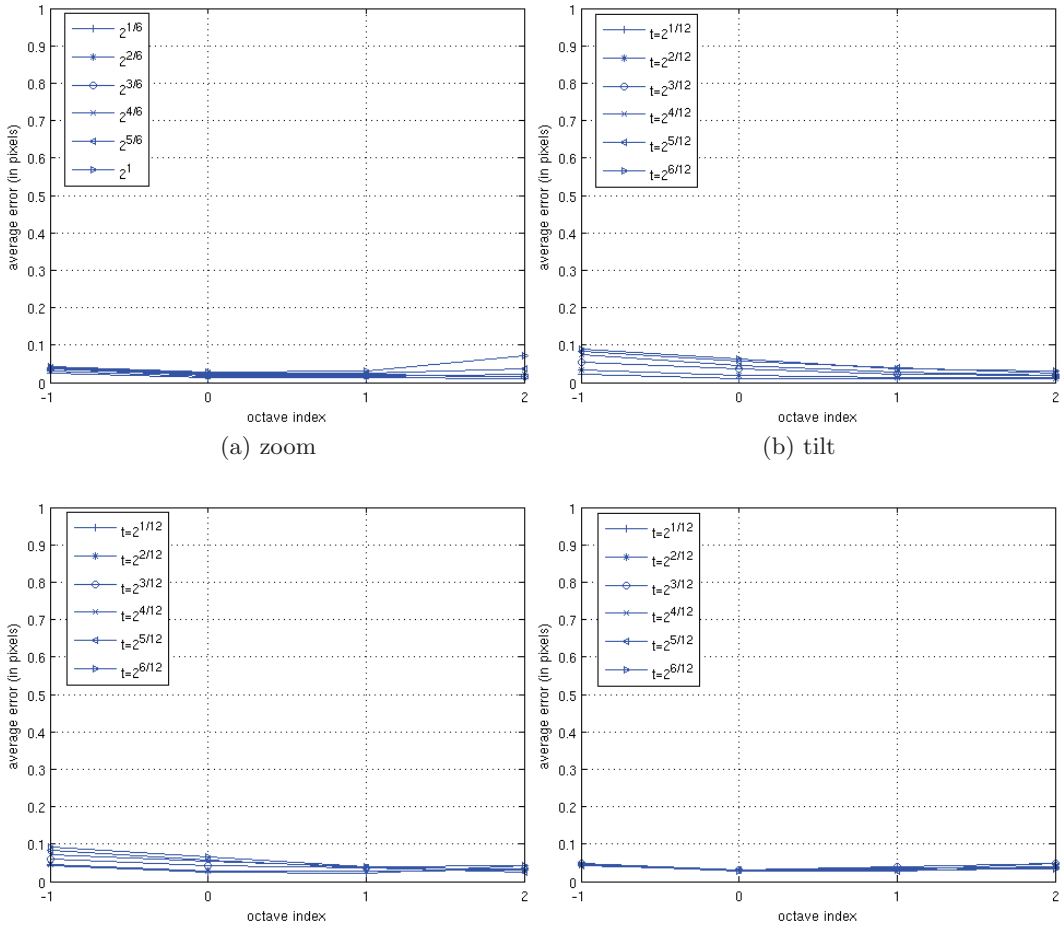


(e) affine transformation



(f) homography

Figure 8.11: The average matching error of improved SIFT scheme by canceling the sub-sampling in scale space, under different geometric transformations. (a) translation: the translation in x and y direction is $(45, 32)$, $(45.1, 32.1)$, $(45.3, 32.3)$, $(45.5, 32.5)$, $(45.7, 32.7)$ and $(45.9, 32.9)$. (b) rotation: the rotation angle varies from $15°$ to $85°$ with the step of $10°$. (c) zoom: the zoom factor is $2^{1/6}$, $2^{2/6}$, $2^{3/6}$, $2^{4/6}$, $2^{5/6}$ and $2^{6/6}$. (d) tilt: Eq (8.9) with $\phi = 0°$, $\psi = 0°$, t is $2^{1/12}$, $2^{2/12}$, $2^{3/12}$, $2^{4/12}$, $2^{5/12}$, $2^{6/12}$. (e) affine transformation: Eq (8.9) with $\phi = 37°$, $\psi = 24°$ and t is $2^{1/12}$, $2^{2/12}$, $2^{3/12}$, $2^{4/12}$, $2^{5/12}$ and $2^{6/12}$. (f) homography: Eq. (8.10) by adding a projective component in the affine tranformation. The x-axis is the octave index, from -1 to 2. The range of y-axis is from 0 pixels to 1.0 pixels.



(a) zoom (b) tilt

(c) affine transformation (d) homography

Figure 8.12: The average matching error of iterative SIFT scheme under different geometric transformations. (a) zoom: the zoom factor is $2^{1/6}$, $2^{2/6}$, $2^{3/6}$, $2^{4/6}$, $2^{5/6}$ and $2^{6/6}$. (b) tilt: Eq (8.9) with $\phi = 0°$, $\psi = 0°$, t is $2^{1/12}$, $2^{2/12}$, $2^{3/12}$, $2^{4/12}$, $2^{5/12}$, $2^{6/12}$. (c) affine transformation: Eq (8.9) with $\phi = 37°$, $\psi = 24°$ and t is $2^{1/12}$, $2^{2/12}$, $2^{3/12}$, $2^{4/12}$, $2^{5/12}$ and $2^{6/12}$. (f) homography: Eq. (8.10) by adding a projective component in the affine tranformation. The x-axis is the octave index, from -1 to 2. The range of y-axis is from 0 pixels to 1.0 pixels.

neighbor matchings around one matching. By using each point-wise homography to correct the position of feature points in one image, the precision of matchings will be increased. This method works well in lens distortion correction.

This method requires a dense set of SIFT matching between two images, which is only possible for highly textured images or images containing many details. To have enough matchings for natural images, we consider every pixel as a local extremum in the first two octaves of improved SIFT scheme, followed by the 3D refinement and descriptor construction. Only a low Laplacian threshold is used to eliminate very unstable feature points. This is similar to the SIFT flow [112], which is used to align images at scene level. The difference is that the SIFT flow uses all of the pixels at the image initial scale without exploiting the other scales. The positions are neither refined by 3D interpolation. The "outlier" removal by vector filter and matching acceleration technique are also used to obtain more reliable SIFT matchings. Someone might argue that this technique introduces more imprecise SIFT matchings than Lowe's SIFT because only a low Laplacian threshold is applied. But the experimental results show that the local homography filter is enough to refine these coarse matchings to reach an average precision better than 0.1 pixel. We evaluate the whole procedure on the different geometric transformations as before. The matchings in octave -1 and 0 are mixed together to have a dense matchings over image domain. The evaluation then gives the average matching error without distinguishing their octave index. In Table 8.3.4, the average matching error is shown for several geometric transformations. For transformations without scale changes, the precision is comparable or better than the improved SIFT, while for transformations with scale changes, the precision is evidently improved compared to Lowe's SIFT and improved SIFT. The default of this method is that not many features can be matched by SIFT under the transformations too "affine", even though each pixel is considered as a local extremum. This will degrade the filtering performance of local homographies. This explains why the precision decreases in the case of tilt, affine transformation and homography with the increase of the parameter from $2^{1/12}$ to $2^{6/12}$ in Table 8.3.4. This is in fact not a problem because we can always make two images to be more similar by transforming one image using a coarse homography between them, even if the relation between both images is not a homography. ASIFT [131] can also be tried to find more matchings. More precision can be obtained by just keeping the locally filtered matchings whose Laplacian value is bigger than a higher threshold.

For real images, the matching precision is also affected by the noise in the image. To test the performance of the algorithm for real images, a Canon EOS 30D SLR camera with EFS $18-55$mm lens was used to take images. Three pairs of images were tested (Fig. 8.13): the first pair for a planar abstract painting, the second pair for an infinite homography, the third pair for a distant wall with small camera motion. Ideally the underlying geometric transformation is a homography. But for real images, the homography cannot be used to measure the residual error due to the non-linear distortion. Even though the maximal focal length (55mm) was chosen to avoid the lens distortion as much as possible, there still exists small distortion. The polynomial model, an universal and practical distortion model, which is also consistent with a homography, was used here to evaluate the matching precision. The result is recapitulated in Table 8.3.4. It seems that the local filtering technique is very efficient to increase the matching precision even with noisy images.

		avg. without local filter (in pixels)			avg. with local filter (in pixels)		
		100%	50% estim.	50% verif.	100%	50% estim.	50% verif.
translation	$(45, 32)$	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001
	$(45.1, 32.1)$	0.0912	0.0922	0.0908	0.0187	0.0189	0.0188
	$(45.3, 32.3)$	0.1962	0.2068	0.1861	0.0311	0.0311	0.0311
	$(45.5, 32.5)$	0.0222	0.0425	0.0023	0.0055	0.0058	0.0052
	$(45.7, 32.7)$	0.1938	0.2055	0.1822	0.0293	0.0297	0.0290
	$(45.9, 32.9)$	0.0915	0.0925	0.0910	0.0190	0.0191	0.0191
rotation	$15°$	0.1589	0.1728	0.1450	0.0271	0.0271	0.0271
	$25°$	0.1674	0.1844	0.1505	0.0271	0.0271	0.0272
	$35°$	0.1738	0.1915	0.1563	0.0296	0.0298	0.0295
	$45°$	0.1749	0.1937	0.1562	0.0288	0.0291	0.0285
	$55°$	0.1770	0.1948	0.1591	0.0296	0.0295	0.0296
	$65°$	0.1704	0.1859	0.1550	0.0287	0.0289	0.0286
	$75°$	0.1670	0.1825	0.1516	0.0280	0.0284	0.0277
	$85°$	0.1589	0.1733	0.1446	0.0272	0.0274	0.0271
zoom	$2^{1/6}$	0.2188	0.2178	0.2199	0.0333	0.0327	0.0338
	$2^{2/6}$	0.1537	0.1799	0.1276	0.0256	0.0260	0.0254
	$2^{3/6}$	0.2009	0.2058	0.1961	0.0331	0.0333	0.0329
	$2^{4/6}$	0.1443	0.1639	0.1251	0.0247	0.0252	0.0242
	$2^{5/6}$	0.1909	0.1968	0.1854	0.0329	0.0334	0.0329
	$2^{6/6}$	0.1362	0.1438	0.1292	0.0217	0.0217	0.0218
tilt	$2^{1/12}$	0.1054	0.1027	0.1081	0.0186	0.0186	0.0187
	$2^{2/12}$	0.1616	0.1527	0.1705	0.0271	0.0277	0.0272
	$2^{3/12}$	0.2176	0.2043	0.2310	0.0359	0.0363	0.0370
	$2^{4/12}$	0.2652	0.2465	0.2843	0.0435	0.0429	0.0440
	$2^{5/12}$	0.2926	0.2776	0.3087	0.0443	0.0434	0.0462
	$2^{6/12}$	0.3223	0.3005	0.3492	0.0590	0.0563	0.0734
affine	$2^{1/12}$	0.2002	0.2133	0.1874	0.0340	0.0340	0.0341
	$2^{2/12}$	0.2299	0.2314	0.2287	0.0370	0.0372	0.0370
	$2^{3/12}$	0.2686	0.2646	0.2736	0.0423	0.0425	0.0429
	$2^{4/12}$	0.2963	0.2910	0.3036	0.0518	0.0533	0.0604
	$2^{5/12}$	0.3146	0.2961	0.3348	0.0511	0.0492	0.0555
	$2^{6/12}$	0.3377	0.3239	0.3529	0.0564	0.0530	0.0626
homography	$2^{1/12}$	0.2444	0.2368	0.2541	0.0426	0.0408	0.0455
	$2^{2/12}$	0.2497	0.2376	0.2629	0.0438	0.0427	0.0455
	$2^{3/12}$	0.2794	0.2564	0.3046	0.0495	0.0451	0.0543
	$2^{4/12}$	0.3161	0.3055	0.3283	0.0661	0.0655	0.0730
	$2^{5/12}$	0.3161	0.2966	0.3357	0.0516	0.0491	0.0550
	$2^{6/12}$	0.3639	0.3812	0.3631	0.0855	0.0938	0.0903

Table 8.1: The average residual error of improved SIFT with/without local filter under different geometric transformations. Column 1 is the type of transformation. Column 2 is the transformation parameters. The average residual error without local filter is recapitulated in column 3 to column 5. Column 3 is the residual error with 100% matchings. Column 4 is the residual error with 50% matchings. The other 50% matchings are used for verification in column 5. With the local filter, the same statistics are shown in column 6 to column 8.

Figure 8.13: Three pairs of images taken by Canon EOS 30D SLR camera with focal length 55mm.

image	order	avg. without local filter (in pixels)			avg. local filter (in pixels)		
		100%	50% estim.	50% verif.	100%	50% estim.	50% verif.
drawing	3	0.51	0.51	0.50	0.11	0.11	0.11
	4	0.50	0.51	0.50	0.10	0.09	0.10
	5	0.50	0.51	0.50	0.09	0.09	0.09
	6	0.50	0.51	0.51	0.08	0.08	0.09
	7	0.50	0.51	0.51	0.08	0.08	0.08
	8	0.50	0.51	0.51	0.07	0.07	0.07
	9	0.50	0.52	0.51	0.07	0.07	0.07
	10	0.50	0.52	0.52	0.06	0.06	0.06
house	3	0.35	0.35	0.35	0.08	0.08	0.08
	4	0.35	0.35	0.35	0.07	0.07	0.07
	5	0.35	0.35	0.35	0.07	0.07	0.07
	6	0.35	0.35	0.35	0.06	0.06	0.06
	7	0.35	0.35	0.35	0.06	0.06	0.06
	8	0.35	0.35	0.34	0.06	0.05	0.06
	9	0.34	0.35	0.34	0.05	0.05	0.05
	10	0.34	0.35	0.34	0.05	0.05	0.05
wall	3	0.28	0.29	0.28	0.03	0.03	0.03
	4	0.28	0.29	0.28	0.03	0.03	0.03
	5	0.28	0.29	0.28	0.03	0.03	0.03
	6	0.28	0.29	0.28	0.03	0.03	0.03
	7	0.28	0.29	0.28	0.03	0.03	0.03
	8	0.28	0.29	0.28	0.03	0.03	0.03
	9	0.28	0.29	0.28	0.03	0.03	0.03
	10	0.28	0.29	0.28	0.03	0.03	0.03

Table 8.2: The polynomial model is used to evaluate the average residual error (in pixels) of the improved SIFT with/without local filter for three pairs of images in Fig. 8.13. Column 2 is the polynomial order. The average residual error without local filter is recapitulated in column 3 to column 5. Column 3 is the residual error with 100% matchings. Column 4 is the residual error with 50% matchings. The other 50% matchings are used for verification in column 5. With the local filter, the same statistics are shown in column 6 to column 8.

8.4 Conclusion

The matching precision of SIFT and its improved scheme have been evaluated under several geometric transformations. Precision improvements were observed when there is no scale change between two images. But the precision still suffers from the blur inconsistency caused by the scale change between two images, even though the sub-sampling is canceled in the improved scheme. An iterative scheme is thus proposed to first estimate the homography between two images, then apply improved SIFT scheme again to improve the matching precision. In practice, the lens distortion should also be considered. Under the assumption that the transformation between two images is smooth, we have applied a local homography filter to refine the precision of each pair of correspondences between two images. This technique works well both in synthetic and real images.

Chapter 9

Burst Denoising

Contents

Abstract *Photon accumulation on a fixed surface is the essence of photography. In the times of chemical photography this accumulation required the camera to move as little as possible, and the scene to be still. Yet, most recent reflex and compact cameras propose a burst mode, permitting to capture quickly dozens of short exposure images of a scene instead of a single one. This new feature permits in principle to obtain by simple accumulation high quality photographs in dim light, with no motion or aperture blur. It also gives the right data for an accurate noise model. Yet, both goals are attainable only if an accurate cross-registration of the burst images has been performed. The difficulty comes from the non negligible image deformations caused by the slightest camera motion, in front of a 3D scene, and from the light variations or motions in the scene. This chapter proposes a numerical processing chain permitting to achieve jointly the two mentioned goals: an accurate noise model for the camera, which is used crucially to obtain a state of the art multi-images denoising. The key feature of the proposed processing chain is a reliable multi-image noise estimator, whose accuracy will be demonstrated by three different procedures. Thanks to the signal dependent noise model obtained from the burst itself, a faithful detection of the well registered pixels can be made. The denoising by simple accumulation of these pixels, which are an overwhelming majority, permits to extend the Nicéphore Niepce photon accumulation method to image bursts. The denoising performance by accumulation is shown to reach the theoretical limit, namely a $\sqrt{n}$ denoising factor for n frames. Comparison with state of the art denoising algorithms will be shown on several bursts taken with reflex cameras in dim light.*

9.1 Introduction

The accumulation of photon impacts on a surface is the essence of photography. The first Nicephore Niepce photograph [36] was obtained after an eight hours exposure. The serious objection to a long exposure is the variation of the scene due to changes in light, camera motion, and incidental motions of parts of the scene. The more these variations can be compensated, the longer the exposure can be, and the more the noise can be reduced. It is a frustrating experience for professional photographers to take pictures under bad lighting conditions with a hand-held camera. If the camera is set to a long exposure time, the photograph gets blurred by the camera motions and aperture. If it is taken with short exposure, the image is dark, and enhancing it reveals the noise. Yet, this dilemma can be solved by taking a burst of images, each with short-exposure time, as shown in Fig. 9.1, and by averaging them after registration. This observation is not new and many algorithms have been proposed, mostly for stitching and super-resolution. These algorithms have thrived in the last decade, probably thanks to the discovery of a reliable algorithm for image matching, the SIFT algorithm [117]. All of the multi-image fusion algorithms share three well separated stages, the search and matching of characteristic points, the registration of consecutive image pairs and the final accumulation of images. All methods perform some sort of multi-image registration, but surprisingly do not propose a procedure to check if the registration is coherent. Thus, there is a non-controlled risk that the accumulation blurs the final accumulation image, due to wrong registrations. Nevertheless, as we shall see, the accurate knowledge of noise statistics for the image sequence permits to detect and correct all registration incoherences. Furthermore, this noise statistics can be most reliably extracted from the burst itself, be it for raw or for JPEG images. In consequence, a stand alone algorithm which denoises any image burst is doable.

As experiments will show, it even allows for light variations and moving objects in the scene, and it reaches the $\sqrt{n}$ denoising factor predicted for the sum of the n independent (noise) random variables.

We call in the following "burst", or "image burst" a set of digital images taken from the same camera, in the same state, and quasi instantaneously. Such bursts are obtained by video, or by using the burst mode proposed in recent reflex and compact cameras. The camera is supposed to be held as steady as possible so that a large majority of pixels are seen through the whole burst. Thus, no erratic or rash motion of the camera is allowed, but instead incident motions in the scene do not hamper the method.

There are other new and promising approaches, where taking images with different capture conditions is taken advantage of. Liu et al. [185] combine a blurred image with long-exposure time, and a noisy one with short-exposure time for the purpose of denoising the second and deblurring the first. Beltramio and Levine [16] improve the dynamic range of the final image by combining an underexposed snapshot with an overexposed one. Combining again two snapshots, one with and the other without flash, is investigated by Eisemann *et. al.* [58] and Fattal *et. al* [64]. Another case of image fusion worth mentioning is [14], designed for a 3D scanning system. During each photography session, a high-resolution digital back is used for photography, and separate macro (close-up) and ultraviolet light shots are taken of specific areas of text. As a result, a number of folios are captured with two sets of data: a "dirty" image with registered 3D geometry and a "clean" image with the page potentially deformed differently to which the digital flattening algorithms are applied.

Our purpose here is narrower. We only aim at an accurate noise estimation followed by denoising for an image burst. No super-resolution will be attempted, nor the combination of images taken under different apertures, lightings or positions. The main assumption on the setting is that a hand-held camera has taken an image burst of a still scene, or from a scene with a minority of moving objects. To get a significant denoising, the number of images can range from 9 to 64, which grants a noise reduction by a factor 3 to 8. Since the denoising performance grows like the square root of the number of images, it is less and less advantageous to accumulate images when their number grows. But impressive denoising factors up to 6 or 8 are reachable by the simple algorithm proposed here, which we shall call *average after registration* (AAR). Probably the closest precursor to the present method is the multiple image denoising method by Zhang *et. al.* [189]. Their images are not the result of a burst. They are images taken from different points of views by different cameras. Each camera uses a small aperture and a short exposure to ensure minimal optical defocus and motion blur, to the cost of very noisy output. A global registration evaluating the 3D depth map of the scene is computed from the multi-view images, before applying a patch based denoising inspired by NL-means [29]. Thus the denoising strategy is more complex than the simple accumulation after registration which is promoted in the present chapter. Nevertheless, the authors remark that their denoising performance stalls when the number of frames grows, and write that this difficulty should be overcome. Yet, their observed denoising performance curves grow approximately like the square root of the number of frames, which indicates that the real performance of the algorithm is due to the accumulation. The method proposed here therefore goes back to accumulation, as the essence of photography. It uses, however, a hybrid scheme which decides at each pixel between accumulation and block denoising, depending on the reliability of the match. The comparison of temporal pixel statistics with the noise model

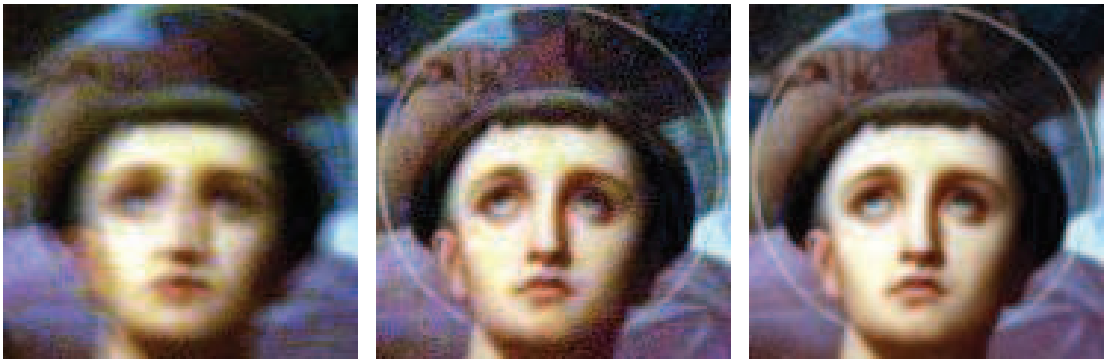


Figure 9.1: From left to right: one long-exposure image (time = 0.4 sec, ISO=100), one of 16 short-exposure images (time = 1/40 sec, ISO = 1600) and the average after registration. All images have been color-balanced to show the same contrast. The long exposure image is blurry due to camera motion. The middle short-exposure image is noisy, and the third one is some **four times** less noisy, being the result of averaging 16 short-exposure images. Images may need to be zoomed in on a screen to compare details and textures.

extracted from the scene itself permits a reliable conservative decision so as to apply or not the *accumulation after registration* (AAR). Without the accurate nonparametric noise estimation, this strategy would be unreliable. Therefore estimating accurately the noise model in a burst of raw or JPEG images is the core contribution of this chapter. A more complex and primitive version of the hybrid method was announced in the conference paper [31]. It did not contain the noise estimation method presented here.

Plan and Results The chapter requires a rich bibliographical analysis for the many aspects of multi-image processing (Section 9.3). This survey shows that most super-resolution algorithms do in fact much more denoising than they do super-resolution, since they typically only increase the size of the image by a factor 2 or 3, while the number of images would theoretically allow for a 5 to 8 factor. Section 9.2 reviews the other pilar of the proposed method, the noise estimation literature. (This corpus is surprisingly poor in comparison to the denoising literature.)

Section 9.4 is key to the proposed technique, as it demonstrates that a new variant of static noise blind estimate gives results that exactly coincide with Poisson noise estimates taken from registered images in a temporal sequence. It is also shown that although JPEG images obtained by off-the-shelf cameras have no noise model, a usable substitute to this noise model can be obtained: It simply is the variance of temporal sequences of registered images.

Section 9.5 describes the proposed multi-image denoising method, which in some sense trivializes the denoising technology, since it proposes to go back as much as possible to a mere accumulation, and to perform a more sophisticated denoising only at dubiously registered pixels. Section 9.6 compares the proposed strategy with two state of the art multi-images denoising strategies.

9.2 Noise Estimation, a Review

As pointed out in [110], "Compared to the in-depth and wide literature on image denoising, the literature on noise estimation is very limited". Following the classical study by Healey et al. [90], the noise in CCD sensors can be approximated by an additive, white and signal dependent noise model. The noise model and its variance reflect different aspects of the imaging chain at the CCD, mainly dark noise and shot noise. Dark noise is due to the creation of spurious electrons generated by thermal energy which become indistinguishable from photoelectrons. Shot noise is a result of the quantum nature of light and characterizes the uncertainty in the number of photons stored at a collection site. This number of photons follows a Poisson distribution so that its variance equals its mean. The overall combination of the different noise sources therefore leads to an affine noise variance $a + bu$ depending on the original signal value u. Yet, this is only true for the raw CCD image. Further processing stages in the camera hardware and software such as the white balance, the demosaicking, the gamma correction, the blur and color corrections, and eventually the compression, correlate the noise and modify its nature and its standard deviation in a non trivial manner. There is therefore no noise model for JPEG images. However, as we shall see, a signal dependent noise variance model can still be estimated from bursts of JPEG images (section 9.4.2.) It is enough to perform reliably the average after registration (AAR).

9.2.1 Additive Gaussian Noise Estimation

Most computer vision algorithms should adjust their parameters according to the image noise level. Surprisingly, there are few papers dealing with the noise estimation problem, and most of them only estimate the variance of a signal independent additive white Gaussian noise (AWGN). This noise statistics is typically measured on the highest-frequency portion of the image spectrum, or on homogenous image patches. In the AWGN case a spectral decomposition through an orthonormal transform such as wavelets or the DCT preserves the noise statistics. To estimate the noise variance, Donoho et. al [54] consider the finest scale wavelet coefficients, followed by a median filter to eliminate the outliers. Suppose $\{y_i\}_{i=1,\cdots N}$ be N independent Gaussian random variables of zero-mean and variance σ^2, then

$$E\{\text{MED}(|y_i|)\} \approx 0.6745\sigma.$$

It follows immediately that the noise standard deviation σ is given by

$$\tilde{\sigma} = \frac{1}{0.6745}\text{MED}(|y_i|) = 1.4826\text{MED}(|y_i|).$$

The standard procedure of the local approaches is to analyze a set of local estimates of the variance. For example, Rank et. al [153] take the maximum of the distribution of image derivatives. This method is based on the assumption that the underlying image has a large portion of homogeneous regions. Yet, if an image is highly textured, the noise variance can overestimated. To overcome this problem, Ponomarenko et. al [149] have proposed to analyze the local DCT coefficients. A segmentation-based noise estimation is carried out in [2], which considers both i.i.d. and spatially correlated noise.

The algorithm in [150] is a modification of the early work [149] dealing with AVIRIS (Airborne Visible Infrared Imaging Spectrometer) images, in which the evaluation of the noise

variance in sub-band images is addressed. The idea is to divide each block into low frequency and high frequency components by thresholding, and to use K blocks of the smallest variance of the low frequency coefficients to calculate a noise variance, where K is adaptively selected so that it is smaller for highly-textured images.

[45] proposed an improvement of the estimate of the variance of AWGN by using transforms creating a sparse representation (via BM3D [42]) and using robust statistics estimators (MAD and ICI). For a univariate data set $X_1, X_2, ..., X_n$, the MAD is defined as the median of the absolute deviations from the data's median: $\text{MAD} = \text{median}_i\,(\,|X_i - \text{median}_j(X_j)|\,)$. The algorithm is as follows.

1. for each 8×8 block, group together up to 16 similar non-overlapping blocks into 3D array. The similarity between blocks in evaluated by comparing corresponding blocks extracted from a denoised version by BM3D.

2. apply a 3-D orthonormal transform (DCT or wavelet) on each group and sort the coefficients according to the zig-zag scan.

3. collect the first 6 coefficients $c_1, \cdots, c_6$ and define their empirical energy as the mean of the magnitude of the (up to 32) subsequent coefficients:

$$E\{|c_j|^2\} = \text{mean}\{|c_{j+1}^2, \cdots, c_{j+32}^2|\}$$

4. Sort the coefficients from all the groups (6 coefficients per group) according to their energy

5. do MAD and Intersection of Confidence Intervals (ICI) [79] to achieve the optimal bias-variance trade-off in the MAD estimation.

All the above mentioned algorithms give reasonable estimates of the standard deviation **when the noise is uniform**. Yet, when applying these algorithms to estimate signal dependent noise, the results are poor. The work of C. Liu *et. al.* [111] estimates the upper bound on the noise level fitting to a camera model. The noise estimation from the raw data is discussed in [69, 70]. The former is a parametric estimation by fitting the model to the additive Poissonian-Gaussian noise from a single image, while the latter measures the temporal noise based on an automatic segmentation of 50 images.

9.2.2 Poisson Noise Removal

This chapter deals with real noise, which in most real images (digital cameras, tomography, microscopy and astronomy) is a Poisson noise. The Poisson noise is inherent to photon counting. This noise adds up to a thermal noise and an electronic noise which are approximately AWGN. In the literature algorithms considering the removal of AWGN are dominant but, if its model is known, Poisson noise can be approximately reduced to AWGN by a so called variance stabilizing transformation (VST). The standard procedure follows three steps,

1. apply VST to make the data homoscedastic

2. denoise the transformed data

3. apply the inverse VST.

The square-root operation is widely used as a VST,

$$f(z) = b\sqrt{z + c}. \tag{9.1}$$

It follows from the asymptotic unit variance of $f(z)$ that the parameters are given by $b = 2$ and $c = 3/8$, which is the Anscombe transform [5]. A multiscale VST (MS-VST) is studied in [188] along with the conventional denoising schemes based on wavelets, ridgelets and curvelets depending on morphological features (isotropic, line-like, curvilinear, etc) of the given data. It is argued in [121] that the inverse transformation of VST is crucial to the denoising performance. Both the algebraic inverse

$$\mathcal{I}_A(D) = \left(\frac{D}{2}\right)^2 - \frac{3}{8}.$$

and the asymptotically unbiased inverse

$$\mathcal{I}_B(D) = \left(\frac{D}{2}\right)^2 - \frac{1}{8},$$

in [5] are biased for low counts. The authors [121] propose an exact unbiased inverse. They consider an inverse transform $\mathcal{I}_C$ that maps the value $E\{f(z)|y\}$ to the desired value $Ez|y$ that

$$E\{f(z)|y\} = 2\sum_{z=0}^{\infty}\left(\sqrt{z + \frac{3}{8}} \cdot \frac{y^z \exp^{-y}}{z!}\right)$$

where $f(z)$ is the forward Anscombe transform (9.1). In practice, it is sufficient to compute the above equation for a limited set of values y and approximate $\mathcal{I}_C$ by $\mathcal{I}_B$ for large values of y. Furthermore, the state-of-the-art denoising scheme BM3D [69] is applied in the second step.

There are also wavelets based methods [136, 102] or Bayesian [169, 118, 106] removing Poisson noise. In particular, the wavelet-domain Wiener filter [136] uses a cross-validation that not only preserves important image features, but also adapts to the local noise level of the spatially varying Poisson process. The shrinkage of wavelet coefficients investigates how to correct the thresholds [102] to explicitly account for effects of the Poisson distribution on the tails of the coefficient distributions. A recent Bayesian approach by Lefkimmiatis et al. [106] explores a recursive quad-tree image representation which is suitable for Poisson noise degradation and then follows an expectation-maximization technique for parameter estimation and Hidden Markov tree (HMT) structures for inter-scale dependencies. The common denominator to all such methods is that we need an accurate Poisson model, and this will be thoroughly discussed in Section 9.4.

It is, however, a fact that the immense majority of accessible images are JPEG images which contain a noise altered by a long chain of processing algorithms, ending with compression. Thus the problem of estimating noise in a single JPEG image is extremely ill-posed. It has been the object of a thorough study in [110]. This chapter proposes a blind estimation and removal method of color noise from a single image. The interesting feature is that it constructs a "noise level function" which is signal dependent, obtained by computing empirical

standard deviations image homogeneous segments. Of course the remanent noise in a JPEG image is no way white or homogeneous, the high frequencies being notoriously removed by the JPEG algorithm. On the other hand, demosaicking usually acts as a villainous converter of white noise into very structured colored noise, with very large spots. Thus, even the variance of smooth regions cannot give a complete account of the noise damage, because noise in JPEG images is converted in extended flat spots. We shall, however, retain the idea promoted in [110] that, in JPEG images, a signal dependent model for the noise variance can be found. In section 9.4.2 a simple algorithm will be proposed to estimate the color dependent variance in JPEG images from multi-images. All in all, the problem of estimating a noise variance is indeed much better posed if several images of the same scene by the same camera, with the same camera parameters, are available. This technique is classic in lab camera calibration [90].

9.3 Multi-Images and Super Resolution Algorithms

Photo stitching Probably one of the most popular applications in image processing, photo stitching [28, 27] is the first method to have popularized the SIFT method permitting to register into a panorama a set of image of a same scene. Another related application is video stabilization [13]. In these applications no increase in resolution is gained, the final image has roughly the same resolution as the initial ones.

Super-Resolution Super-resolution means creating a higher resolution, larger image from several images of the same scene. Thus, this theme is directly related to the denoising of image bursts. It is actually far more ambitious, since it involves a deconvolution. However, we shall see that most super-resolution algorithms actually make a moderate zoom in, out of many images, and therefore mainly perform a denoising by some sort of accumulation. The convolution model in the found references is anyway not accurate enough to permit a strong deconvolution.

A single-frame super-resolution is often referred to as interpolation. See for example [182, 183]. But several exemplar-based super-resolution methods involve other images which are used for learning, like in Baker and Kanade [9] who use face or text images as priors. Similarly, the patch-example-based approaches stemming from the seminal paper [73], use a nearest-neighbor search to find the best match for local patches, and replace them with the corresponding high-resolution patches in the training set, thus enhancing the resolution. To make the neighbors compatible, the belief-propagation algorithm to the Markov network is applied, while another paper [46] considered a weighted average by surrounding pixels (analogue to nonlocal means [29]). Instead of a nearest-neighbor search, Yang et. al [180] proposed to incorporate the sparsity in the sense that each local patch can be sparsely represented as a linear combination of low-resolution image patches; and a high-resolution image is reconstructed by the corresponding high-resolution elements. The recent remarkable results of [184] go in the same direction. The example-based video enhancement is discussed in [17], where a simple frame-by-frame approach is combined with temporal consistency between successive frames. Also to mitigate the flicker artifacts, a stasis prior is introduced to ensure the consistency in the high frequency information between two adjacent frames.

Focus on Registration In terms of image registration, most of the existing super-resolution methods rely either on a computationally intensive optical flow calculation, or on a parametric global motion estimation. The authors of [194] discuss the effects of multi-image alignment on super-resolution. The flow algorithm they employ addresses two issues: flow consistency (flow computed from frame A to frame B should be consistent with that computed from B to A) and flow accuracy. The flow consistency can be generalized to many frames by computing a consistent bundle of flow fields. Local motion is usually estimated by optical flow, other local deformation models include Delaunay triangulation of features [14] and B-splines [130]. Global motion, on the other hand, can be estimated either in the frequency domain or by feature-based approaches. For example, Vandewalle et. al. [173] proposed to register a set of images based on their low-frequencies, aliasing-free part. They assume a planar motion, and as a result, the rotation angle and shifts between any two images can be precisely calculated in the frequency domain. The standard procedure for feature-based approaches is (1) to detect the key points via Harris corner [35, 7] or SIFT [186, 156], (2) match the corresponding points while eliminating outliers by RANSAC and (3) fit a proper transformation such as a homography. The other applications of SIFT registration are listed in Tab. 9.2.

Reconstruction after Registration A number of papers focus on image fusion, assuming the motion between two frames is either known or easily computed. Elad and Feuer [59] formulate the super-resolution of image sequences in the context of Kalman filtering. They assume that the matrices which define the state-space system are known. For example, the blurring kernel can be estimated by a knowledge of the camera characteristics, and the warping between two consecutive frames is computed by a motion estimation algorithm. But due to the curse of dimensionality of the Kalman filter, they can only deal with small images, e.g. of size 50×50. The work [124] by Marquina and Osher limited the forward model to be spatial-invariant blurring kernel with the down-sampling operator, while no local motion was present. They solved a TV-based reconstruction with Bregman iterations.

A joint approach on demosaicing and super-resolution of color images is addressed in [62], based on their early super-resolution work [63]. The authors use the bilateral-TV regularization for the spatial luminance component, the Tikhonov regularization for the chrominance component and a penalty term for inter-color dependencies. The motion vectors are computed via a hierarchical model-based estimation [15]. The initial guess is the result of the Shift-And-Add method. In addition, the camera PSF is assumed to be a Gaussian kernel with various standard deviation for different sets of experiments.

Methods Joining Denoising, Deblurring, and Motion Compensation Super-resolution and motion deblurring are crossed in the work [11]. First the object is tracked through the sequence, which gives a reliable and sub-pixel segmentation of a moving object [12]. Then a high-resolution is constructed by merging the multiple images with the motion estimation. The deblurring algorithm, which mainly deals with motion blur [94], has been applied only to the region of interest. The recent paper on super-resolution by L. Baboulaz and P. L. Dragotti [7] presents several registration and fusion methods. The registration can be performed either globally by continuous moments from samples, or locally by step edge extraction. The set of registered images is merged into a single image to which either a Wiener or an iterative Modified Residual Norm Steepest Descent (MRNSD) method is applied [134]

to remove the blur and the noise. The super-resolution in [156] uses SIFT + RANSAC to compute the homography between the template image and the others in the video sequence, shifts the low-resolution image with subpixel accuracy and selects the closest image with the optimal shifts.

Implicit Motion Estimation More recently, inspired by the nonlocal movie denoising method, which claims that "denoising images sequences does not require motion estimation" [30], researchers have turned their attention towards super-resolution without motion estimation [57, 56, 152]. Similar methodologies include the steering kernel regression [165], BM3D [44] and its many variants. The forward model in [44] does not assume the presence of the noise. Thus the authors pre-filter the noisy LR input by V-BM3D [41]. They up-sample each image progressively m times, and at each time, the initial estimate is obtained by zero-padding the spectra of the output from the previous stage, followed by filtering. The overall enlargement is three times the original size. Super-resolution in both space and time is discussed in [157, 158], which combine multiple low-resolution video sequences of the same dynamic scene. They register any two sequences by a spatial homography and a temporal affine transformation, followed by a regularization-based reconstruction algorithm.

A Synoptic Table of Super-Resolution Multi-Images Methods Because the literature is so rich, a table of the mentioned methods, classified by their main features, is worth looking at. The methods can be characterized by a) their number k of fused images, which goes from 1 to 60, b) the zoom factor, usually 2 or 3, and therefore far inferior to the potential zoom factor $\sqrt{k}$, c) the registration method, d) the deblurring method, e) the blur kernel. A survey of the table demonstrates that a majority of the methods use many images to get a moderate zoom, meaning that the denoising factor is important. Thus, these methods denoise in some sense by accumulation. But, precisely because all of them aim at super-resolution, none of them considers the accumulation by itself.

Tables 1 and 2 confirm the dominance of SIFT+RANSAC as a standard way to register multi-images, as will also be proposed here in an improved variant. Several of the methods in Table 1 which do not perform SIFT+RANSAC, actually the last four rows, are "implicit". This means that they adhere to the dogma that denoising does not require motion estimation. It is replaced by multiple block motion estimation, like the one performed in NL-means and BM3D. However, we shall see in the experimental section that AAR (average after registration) has a still better performance than such implicit methods. This is one of the main questions that arose in this exploration, and the answer is clear cut: denoising by accumulation, like in ancient photography times still is a valid response in the digital era.

9.4 Noise Blind Estimation

In this section we return to noise estimation and will confront and cross-validate a single frame noise estimation with a multi-images noise estimation.

Table 9.1: comparison of Super Resolution algorithms

Ref.	# of images V.S. factor		Registration	Deblurring	blur kernel
[73] [9]	1	2 to 16	KNN to training set	NO	
[46]	1	2 3		MAP penalty	3×3 5×5
[180]	1	to 4	sparse w.r.t. traning	back-projection	Not mention
[35]	15	2	Harris+RANSAC	Tiknonov	Not mention
[34]	25	3	PCA	NO	
[194]	40	2	consistent flow bundle	NO	
[173]	4	2	frequency domain	NO	
[59]	100	2	assume known motion	Kalman filter	3×3 average
[63, 62]	30	3	hierarchical estimates [15]	bilateral-TV	Gaussian
[156]*	15, 60	2	SIFT+RANSAC	NO	
[186]	20	4	SIFT+RANSAC	Least-square	Gauss($\sigma = 3$)
[11]	10	2	region tracking [12]	motion analysis [94]	motion blur
[7]	20, 40	8	moment-based or Harris + RANSAC	Wiener or MRNSD [134]	B-spline of degree 7
[57] [56]	1 20	2 3	implicit: NLM	NO	
[152] [165]	30	3	implicit: NLM kernel regression	TV bilateral-TV	3×3 average
[44]	9	3	Video-BM3D	zero-padding spectra	3×3 average

Table 9.2: Multi-image SIFT for registration

	Application	# of images	Registration	Blending method
[14]*	manuscript	Not mention	SIFT + RANSAC	Delaunay triangulation
[130]	registration	30 ultrasound 60 MRI	SIFT + threshold + least-square for affine	B-splines deformation
[181] [92]	Mosaic	200 10	SIFT + RANSAC	weighted average
[109]	stitching	6	SIFT + RANSAC	weighted average
[193]	head tracking	1020	SIFT + RANSAC	NA (track 3D motion)

9.4.1 Single Image Noise Estimation

Most noise estimation methods have in common that the noise standard deviation is computed by measuring the derivative or equivalently the wavelet coefficient values of the image. As we mentioned, Donoho et al. [55] proposed to estimate the noise standard deviation as the median of absolute values of wavelet coefficients at the finest scale. Instead of the median, many authors [18, 100] prefer to use a robust median.

Olsen [139] and posteriorly Rank et al. [153] proposed to compute the noise standard deviation by taking a robust estimate on the histogram of sample variances of patches in the derivative image. In order to minimize the effect of edges small windows were preferred, with 3×3 or 5×5 pixels. The sample variance of small patches or the point-wise derivatives provide a non robust measure and require a considerable number of samples with few outliers to guarantee the correct selection of the standard deviation. We observed that the opposite point of view, that is, the use of larger windows 15×15 pixels to 21×21 pixels permits a more robust estimation. However, since larger windows may contain more edges a much smaller percentile will be preferred to the median, in practice the 1% or the 0.5%.

Noise in real photograph images is signal dependent. In order to adapt the noise estimation strategies, the gray level image histogram will be divided adaptively into a fixed number of bins having all the same number of samples. This is preferable to classical approaches where the gray range is divided into equal intervals. Such a uniform division can cause many bins to be almost empty.

To evaluate if a signal dependent noise can be estimated from a single image, 110 images were taken with a Nikon D80, with ISO 100 and very good illumination conditions. These are the best conditions we can expect to have a low noise standard deviation. These color images were converted to gray level by averaging the three color values at each pixel. Finally factor 3 sub-sampling was applied by averaging square groups of nine pixels. These operations having divided the noise standard deviation by slightly more than five, these images can be considered as noise free. Finally, a signal dependent noise was added to them, with variance $8 + 2u$ where u was the noiseless grey level.

The uniform and adaptive divisions of the grey level range in a fixed number of 15 bins were compared, and several noise estimation methods were applied to estimate the noise standard deviation inside each bin. The performance of all methods are compared in Table 9.3 showing the average and standard deviation of the errors between the estimated and original noise curves. The best estimate is obtained by applying the proposed strategy using the variance of large patches rather than small ones or point derivatives. These measurements also confirm that the division of the grey level range into bins with fixed cardinality is preferable to the fixed length interval division. This experiment confirms that a signal dependent noise can be estimated with a high accuracy.

Ground Truth? In order to evaluate the performance of such a noise estimation algorithm in real images we need a ground truth to compare with. This ground truth can be obtained for a given camera by taking a sequence of images of the same pattern, after fixing the camera on a pedestal. All camera parameters remain unchanged for all photographs of the sequence, thus avoiding different exposure times or apertures. The temporal average and standard deviation of the whole sequence of images can therefore be computed without any further registration. The use of a piecewise constant image reduces the effect of small vibrations of the camera, see

	MAD	RMAD	MVPD	MVPD2
$\overline{e}$	1.81	2.87	1.58	0.75
std(e)	1.14	2.59	1.06	0.61

a) Uniform gray division

	MAD	RMAD	MVPD	MVPD2
$\overline{e}$	1.66	1.87	1.36	0.73
std(e)	1.04	1.17	0.90	0.35

b) Adaptive gray division

Table 9.3: A signal dependent noise with variance $8+2u$ is added to 110 noise free images. The uniform and adaptive strategies for dividing the grey level range in a fixed number of 15 bins are compared. For each strategy, the following noise estimation methods in each bin are compared: median of absolute derivatives (MAD), robust median of absolute derivatives (RMAD), median of sample variance of patches 3×3 of the derivative image (MVPD) and 0.005 percentile of sample variance of patches 21×21 of the derivative image (MVPD2). Are displayed the average and standard deviation of the errors between the estimated and original noise curves for the 110 images.

Fig. 9.2. The noise in each channel is estimated independently. Each color range is divided adaptively into a fixed number of bins taking into account the color channel histogram. Inside each bin a percentile is used to estimate the standard deviation.

Fig. 9.3 displays the ground truth estimated curves with this strategy, both in RAW and JPEG format for two different ISO settings. The ground truth curves are compared with the ones estimated in the first image of the sequence by the proposed single image noise estimation algorithm. For the RAW case, the single image and ground truth estimated curves are nearly identical. Fig. 9.2 shows a lack of red in the RAW image of the calibration pattern, even if this pattern is actually gray. This effect is corrected by the white balance as observed in the JPEG image.

The ground truth noise curves estimated from the JPEG images do not agree at all with the classical noise model. This is due to the various image range nonlinear transformations applied by the camera hardware during the image formation, which modify the nature and standard deviation of the noise. The ground truth and single image estimated curves in the JPEG case have a similar shape but a different magnitude. The main new feature is that the interpolation and low pass filtering applied to the originally measured values have strongly altered the high frequency components of the noise. Thus, **the noise statistics are no longer computable from a local patch of the image. The estimation of such a noise curve can only be accomplished by computing the temporal variance in a sequence of images of the same scene.**

9.4.2 Multi-Image Noise Estimation

A temporal average requires the images of the sequence to be perfectly registered. Yet, this registration rises a serious technical objection: how to register globally the images of a burst?

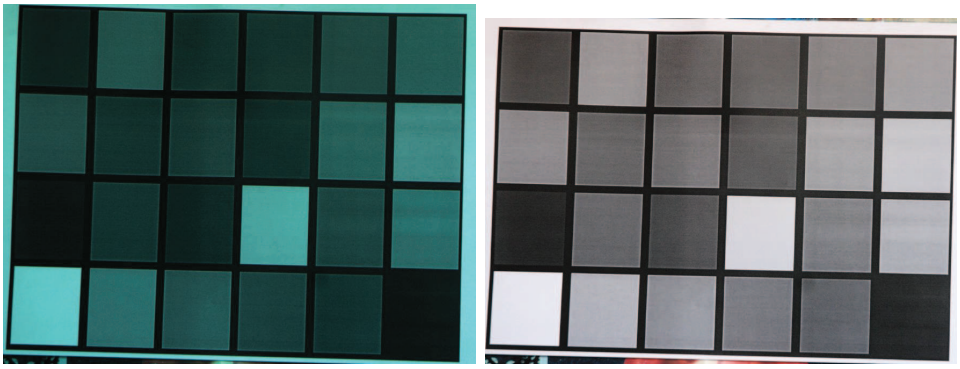


Figure 9.2: Calibration pattern used for noise ground truth estimation. Left: raw image. Right: JPEG image. Even if the calibration pattern is nearly gray the raw image looks blue because the red is less present. This effect is corrected by the white balance applied by the camera image chain leading to the jpeg image.

Fortunately, there are several situations where the series of snapshots are indeed related to each other by a homography, and we shall explore these situations first. The homography assumption is actually valid in any of the following situations:

1. the only motion of the camera is an arbitrary rotation around its optical center;
2. the photographed objects share the same plane in the 3D scene;
3. the whole scene is far away from the camera.

The computation of an homography between a pair of images needs the accurate correspondence of at least four points in each image. Finding key points in images and matching them is a fundamental step for many computer vision and image processing applications. One of the most robust is the Scale Invariant Feature Transform (SIFT) [117], which we will use. Other possible methods allowing for large baselines are [127, 128, 137, 164, 133, 131], but we are here using images taken with only slight changes of view point.

Because wrong matches occur in the SIFT method, an accurate estimate of the dominant homography will require the elimination of outliers. The standard method to eliminate outliers is RANSAC (RANdom SAmple Consensus) [67]. However, it is efficient only when outliers are a small portion of the whole matching set. For this reason several variants have been proposed to improve the performance of outlier elimination, the principal being [166, 192, 170, 135, 129]. The main difference between our approach and the classic outlier elimination is the fact that we dispose of a whole sequence of images and not just of a pair. Instead of choosing a more elaborate version than RANSAC, we preferred to exploit the sequence redundancy in order to improve the registration stage.

The goal is to estimate a dominant homography for the whole set of images, which are typically a few dozens. Only matches which are common to the whole sequence must be kept. In other terms, the keypoints of the first image are kept only if they are matched with another keypoint in any other image of the sequence. This constraint eliminates most of the outliers (see Algorithm 1). In order to apply such a strategy, we assume that the images overlap

considerably. Recall that the purpose is not to make a mosaic or a panorama, but to estimate the noise curve and eventually to denoise the sequence.

A temporal average and standard deviation is computed for the registered sequence. The average values are used to build a histogram and to divide the grey level range adaptively. Inside each bin, the median value of the corresponding standard deviations is taken.

Algorithm 1: Hybrid Accumulation After Registration Algorithm

Input Initial set of images $I_0, I_1, \cdots, I_n$, obtained from a burst

SIFT
Apply the SIFT algorithm between to each pair (I_0, I_j), $j = 1, \cdots, n$. Call S_j the set of matches.
Retain from S_j only the matches for which the matching key point in I_0 has a match in all other images.

RANSAC
Set number of agreed points, m, to 0.
while the number of trials does not exceed N **do**
  Pick up 4 random points from S_0
  for (each $j > 0$) **do**
    Compute the homography using these 4 points and the corresponding ones in S_j
    Add to m the number of points in S_j which agree with this homography up to the precision p.
  end for
  If $m > maxim$, then $maxim = m$ and save the set of agreed points in the whole sequence
end while
Compute for each pair, the homography H_j with the selected points.

FUSION
Apply the homography H_j to each image obtaining $\bar{I}_j$, $j = 1, \cdots, n$.
Average the transformed images obtaining the mean $\mu(x, y)$. Compute also $\sigma(x, y)$, the temporal standard deviation.
Estimate the noise curve using $\sigma(x, y)$, getting $\sigma_n(u)$ the standard deviation associated to each color u.
Obtain the final estimate:

$$(1 - w(\mu, \sigma))\mu(x, y) + w(\mu, \sigma)\, NL(I_0)(x, y),$$

where NL is the NL-means algorithm (Buades et al. [29]) and the function $w(\nu, \sigma)$ is defined by

$$w(\nu, \sigma) = \begin{cases} 0 & \text{if } \sigma < 1.5\sigma_n(\mu) \\ \frac{\sigma - 1.5\sigma_n(\mu)}{1.5\sigma_n(\mu)} & \text{if } 1.5\sigma_n(\mu) < \sigma < 3\sigma_n(\mu) \\ 1 & \text{if } \sigma > 3\sigma_n(\mu) \end{cases}$$

Fig. 9.4 displays three frames from an image sequence with a rotating pattern and a fixed pedestal. The noise curves estimated from the first image with the single image algorithm

and those from the registered and averaged sequence are displayed in the same figure. The estimated curves in the raw image coincide if either of both strategies is applied. However, as previously observed these are quite different when we take into account the JPEG image.

Images taken with indoor lights often show fast variations of the contrast and brightness, like those in Fig. 9.5. This brightness must be rendered consistent through all the images, so that the standard deviation along time is due to the noise essentially and not to the changes of lights. For this reason, a joint histogram equalization must conservatively be applied before the noise estimation chain. The Midway equalization method proposed in [48, 47] is the ideal tool to do so, since it forces all images to adopt a joint *midway* histogram which is indeed a kind of barycenter of the histograms of all images in the burst. Fig. 9.5 illustrates the noise estimation after and before color equalization.

9.5 Average after Registration Denoising

The core idea of the average after registration (AAR) denoising method is that the various values at a cross-registered pixels obtained by a burst are i.i.d.. Thus, averaging the registered images amounts to averaging several realizations of these random variables. An easy calculation shows that this increases the SNR by a factor proportional to $\sqrt{n}$, where n is the number of shots in the burst.

There is a strong argument in favor of denoising by simple averaging of the registered samples instead of block-matching strategies. If a fine non-periodic texture is present in an image, it is virtually indistinguishable from noise, and actually contains a flat spectrum part which has the same Fourier spectrum as the white noise. Such fine textures can be distinguished from noise only if several samples of the same texture are present in other frames and can be accurately registered. Now, state of the art denoising methods (e.g. BM3D) are based on nonlocal block matching, which is at risk to confound the repeated noise-like textures with real noise. A registration process which is far more global than block matching, using strong features elsewhere in the image, should permit a safer denoising by accumulation, provided the registration is sub-pixel accurate and the number of images sufficient.

A simple test illustrates this superior noise reduction and texture preservation on fine non periodic textures. A image was randomly translated by non integer shifts, and signal dependent noise was added to yield an image sequence of sixteen noisy images. Figure 9.6 shows the first image of the sequence and its denoised version obtained by accumulation after registration (AAR). The theoretical noise reduction factor with sixteen images is four. This factor is indeed reached by the accumulation process. Table 9.4 displays the mean square error between the original image and the denoised one by the different methods. Block based algorithms such as NLmeans [29] and BM3D [43], have a considerably larger error, even if their noise reduction could be theoretically superior due to their two dimensional averaging support. But fine details are lost in the local comparison of small image blocks.

As mentioned in the introduction, the registration by using the SIFT algorithm and computing a homography registration is by now a standard approach in the image fusion literature. The main difference of the proposed approach with anterior work is that the mentioned works do not account for registration errors. Yet, in general, the images of a 3D scene are **not** related by a homography, but by an epipolar geometry [89]. Even if the camera is well-calibrated, a 3D point-to-point correspondence is impossible to obtain without computing the depth of

	Barbara	Couple	Hill
noisy	11.30	11.22	10.27
NLM	4.52	3.73	4.50
BM3D	4.33	3.39	3.90
AR	**3.55**	**3.03**	**2.73**

Table 9.4: Mean square error between the original image and the denoised one by the various considered methods applied on the noisy image sequences in Figure 9.6. The block based algorithms, NLmeans [29] and BM3D [43] have a considerably larger error, even if their noise reduction could be in theory superior, due to their two dimensional averaging support. AAR is close to the theoretical reduction factor four.

the 3D scene. However, as we mentioned, a camera held steadily in the hand theoretically produces images deduced from each other by a homography, the principal image motion being due to slight rotations of the camera. Nonetheless, we should not expect that a simple homography will be perfectly accurate everywhere in each pair, but only on a significant part. A coherent registration will be obtained by retaining only the SIFT matches that are common to the whole burst. Therefore the registration applies a joint RANSAC strategy, as exposed in Algorithm 1. This ensures that the same background objects are used in all images to compute the corresponding homographies.

The main new feature of the algorithm is this: The averaging is applied only at pixels where the observed standard deviation after registration is close to the one predicted by the estimated noise model. Thus, there is no risk whatsoever associated with AAR, because it only averages sets of samples whose variability is noise compatible.

At the other pixels, the conservative strategy is to apply a state of the art video denoising algorithm such as the spatiotemporal NL-means algorithm or BM3D. To obtain a smooth transition between the averaged pixels and the NL-means denoised pixels, a weighting function is used. This function is equal to 0 when the standard deviation of the current pixel is lower than 1.5 times the estimated noise standard deviation, and equal to 1 if it is larger than 3 times the estimated noise standard deviation. The weights are linearly interpolated between 1.5 and 3.

9.6 Discussion and Experimentation

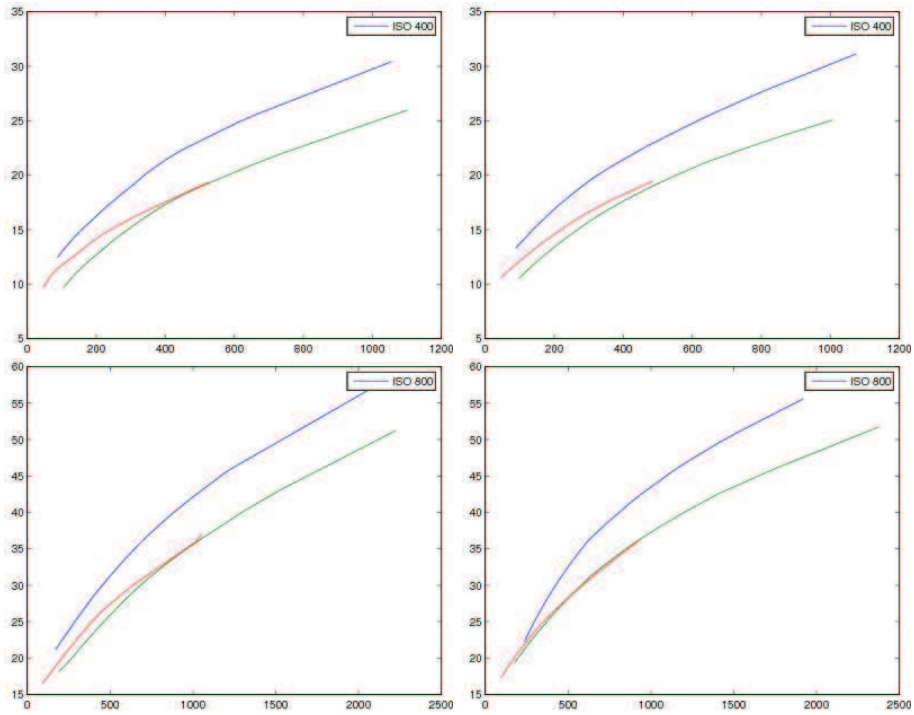
We will compare the visual quality of restored images from real burst sequences. The focus is on JPEG images, which usually contain non white noise and color artifacts. As we illustrated in the previous sections, the variability of the color at a certain pixel cannot be estimated from a single image but from a whole sequence. We will compare the denoised images by using AAR as well as the classical block based denoising algorithms, NL-means. Fig. 9.7 shows the results obtained on three different bursts. Each experiment shows in turn: a) three images extracted from the burst, b) the burst average after registration performed at *all* points, followed by a mask of the image regions in which the temporal standard deviation

is significantly larger than the standard deviation predicted by the noise estimate. At all of these points a block based denoising estimate is used instead of the temporal mean. The final combined image, obtained by an hybridization of the average registration and NL-Means or BM3D, is the right image in each second row.

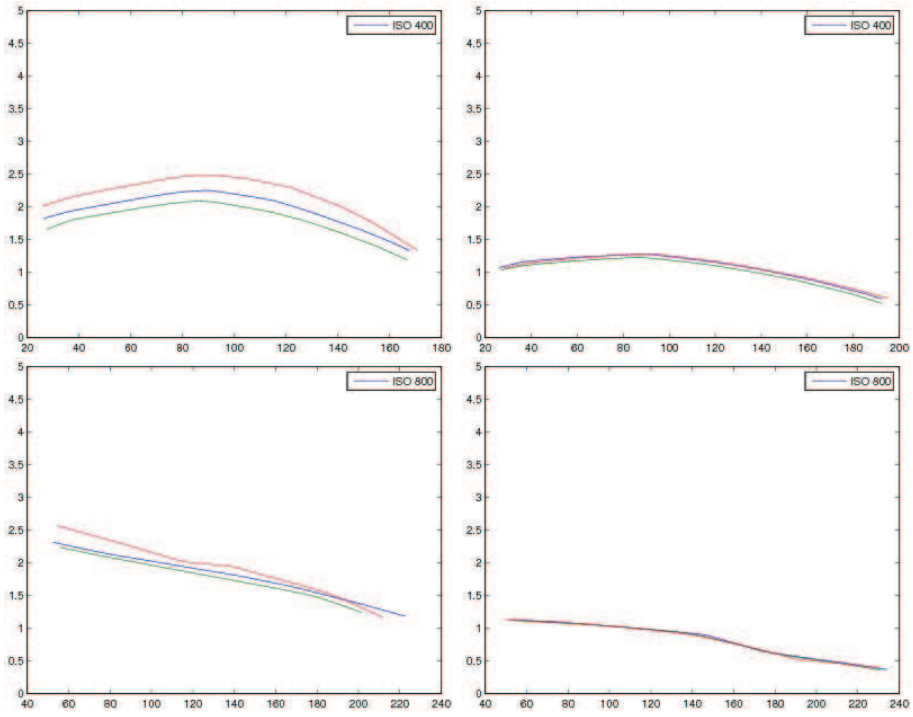
The first experimental data was provided by the company DxO Labs. It captures a rotating pattern with a fixed pedestal. In this case, the dominant homography is a rotation of the main circular pattern, which contains more SIFT points than the pedestal region. Since the proposed algorithm only finds a dominant homography, which is the rotation of the pattern, the simple average fails to denoise the region of the fixed pedestals and of the uniform background. As shown in the white parts of the mask, these regions are detected because they have an excessive temporal standard deviation. They are therefore treated by NL-means or BM3D in the final hybrid result. The whole pattern itself is restored by pure average after registration.

The second burst consists of two books, a newspaper and a moving mouse. Since the dominant homography is computed on still parts, the books and the background, the moving mouse is totally blurred by the averaging after registration, while the rest of the scene is correctly fused. As a consequence, AAR uses the average everywhere, except the part swept by the mouse.

The last burst is a sequence of photographs with short exposure time of a large painting taken in Musée d'Orsay, *Martyrs chrétiens entrant l'amphithéâtre* by Léon Bénouville. Making good photographs of paintings in the dim light of most museums is a good direct application for the proposed algorithm, since the images of the painting are related by a homography even with large changes of view point, the painting being flat. As a result, the average is everywhere favored by the hybrid scheme. Details on the restored images and comparison with BM3D are shown in Fig. 9.8-9.10. Dim light images are displayed after their color values have been stretched to $[0, 255]$.



a) RAW images



b) JPEG images

Figure 9.3: Ground truth and single image noise estimates for the RAW and JPEG images of Fig. 9.2. The estimated curve by the temporal average and standard deviation coincide with the one estimated from the first image by the proposed single image noise estimation algorithm. This is not the case for the JPEG images. The ground truth and single image estimated curves in the JPEG case have a similar shape but a different magnitude. The interpolation and low pass filtering applied to the original measured values have altered the high frequency components of the noise and have correlated its low frequencies. This means that the noise statistics are no longer computable from a local patch of the image. The estimation of a noise curve can only be accomplished by computing the temporal variance in a sequence of images of the same scene.

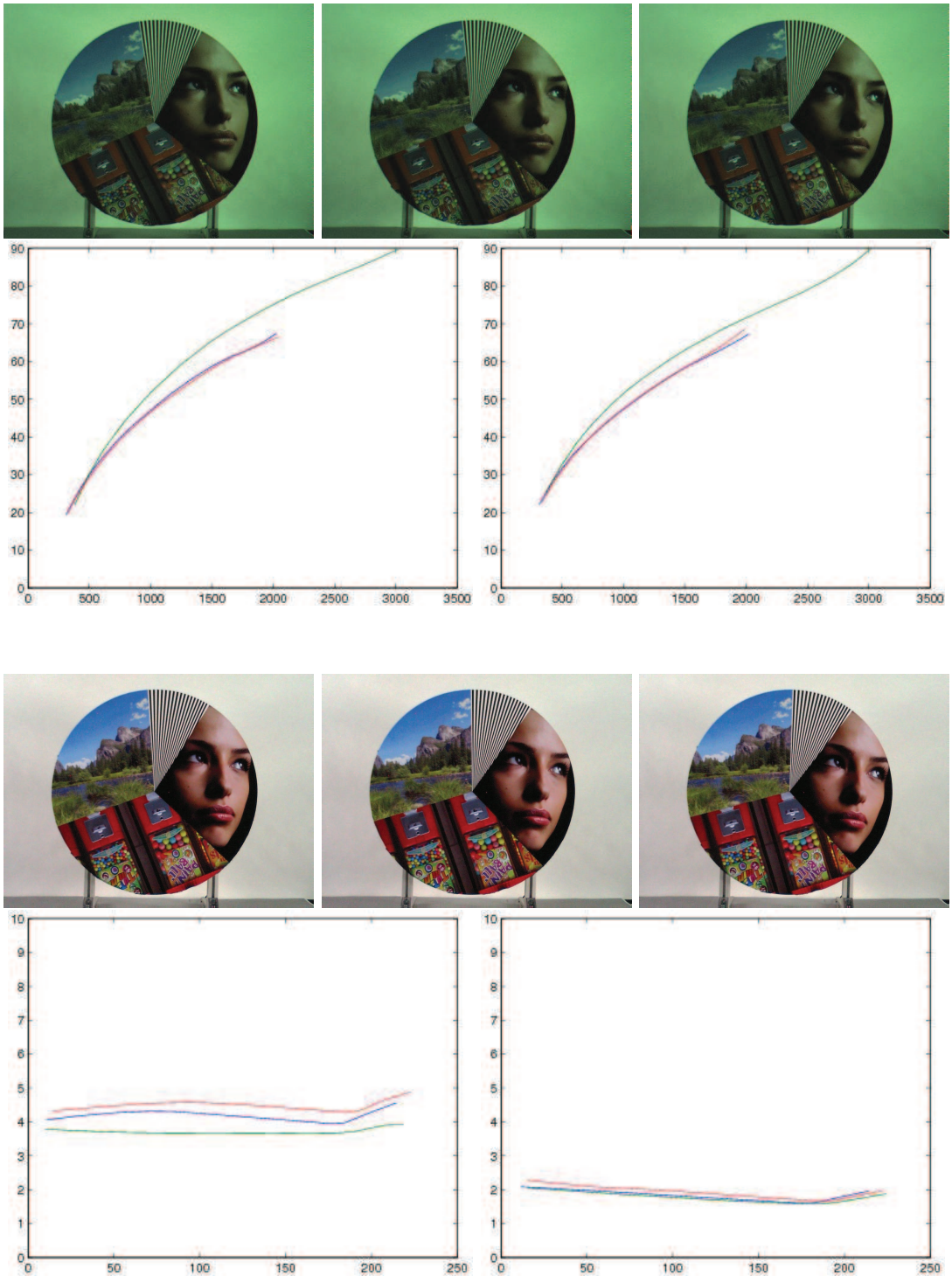


Figure 9.4: Three frames from an image sequence with a rotating pattern and a fixed pedestal both in RAW (top) and JPG (bottom). The estimated curves in the raw image coincide if either of both strategies is applied. However, as previously observed these are quite different when we take into account the JPEG image

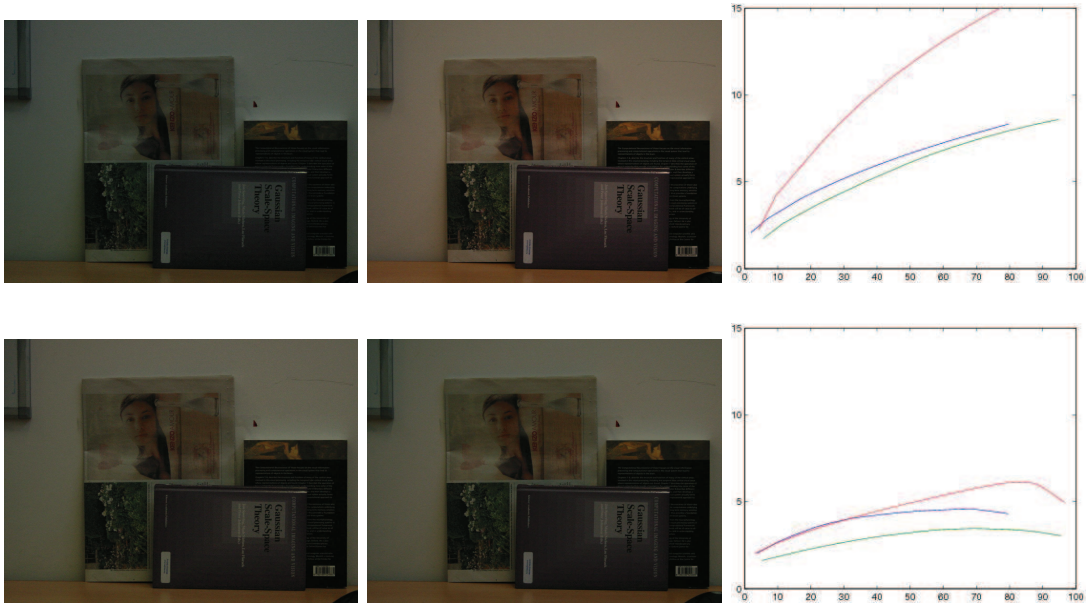


Figure 9.5: Top: two frames of an image sequence with variations of brightness. Noise curve estimated by temporal average and standard deviation after registration. Bottom: the same two frames of the sequence after a joint histogram equalization [47] and estimated noise curves. The second estimation is correct. The first was not, because of the almost imperceptible lighting conditions.

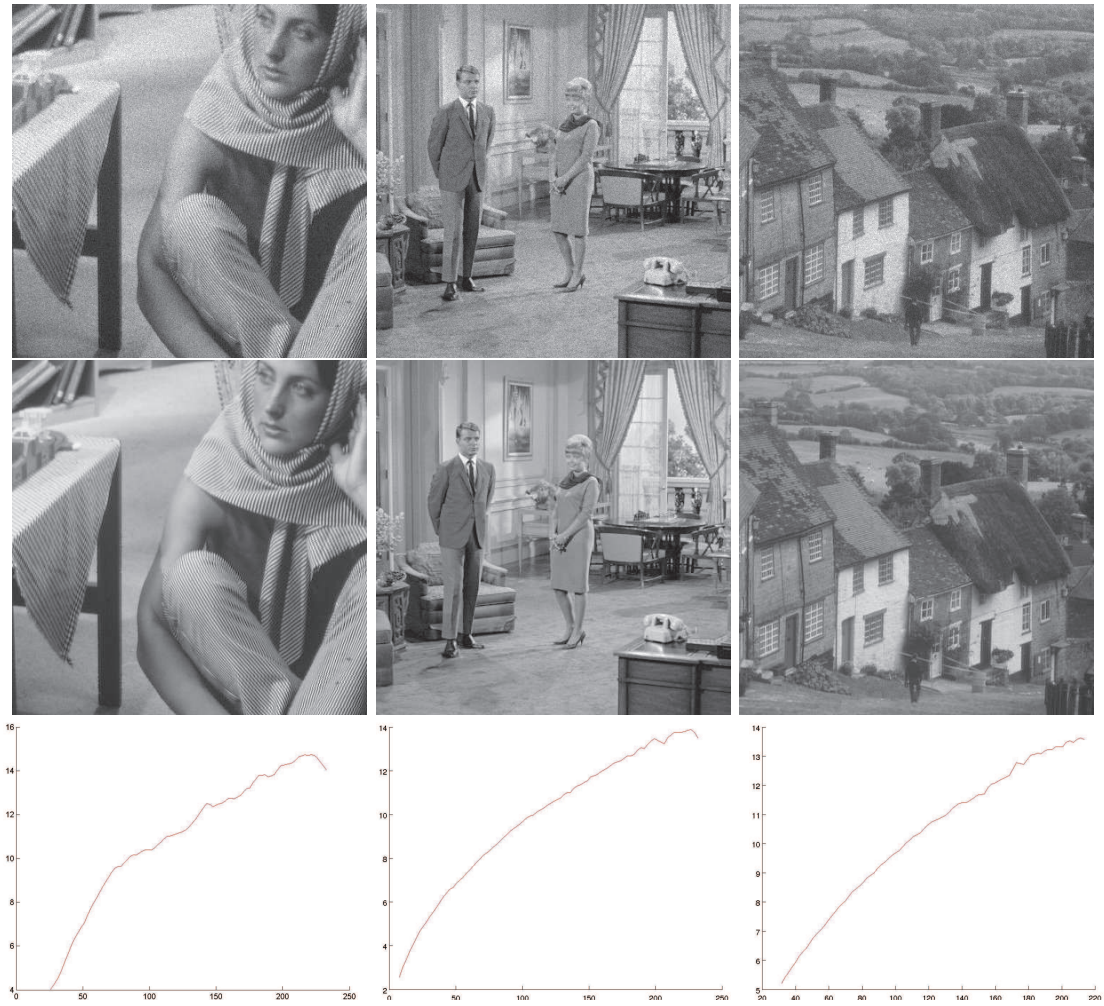


Figure 9.6: Noise curve. From top to bottom: one of the simulated images by moving the image and adding Poisson noise, denoised by accumulation after registration and the noise curve obtained by the accumulation process using the sixteen images. The standard deviation of the noise (Y-axis) fits to the square root of the intensity (X-axis).

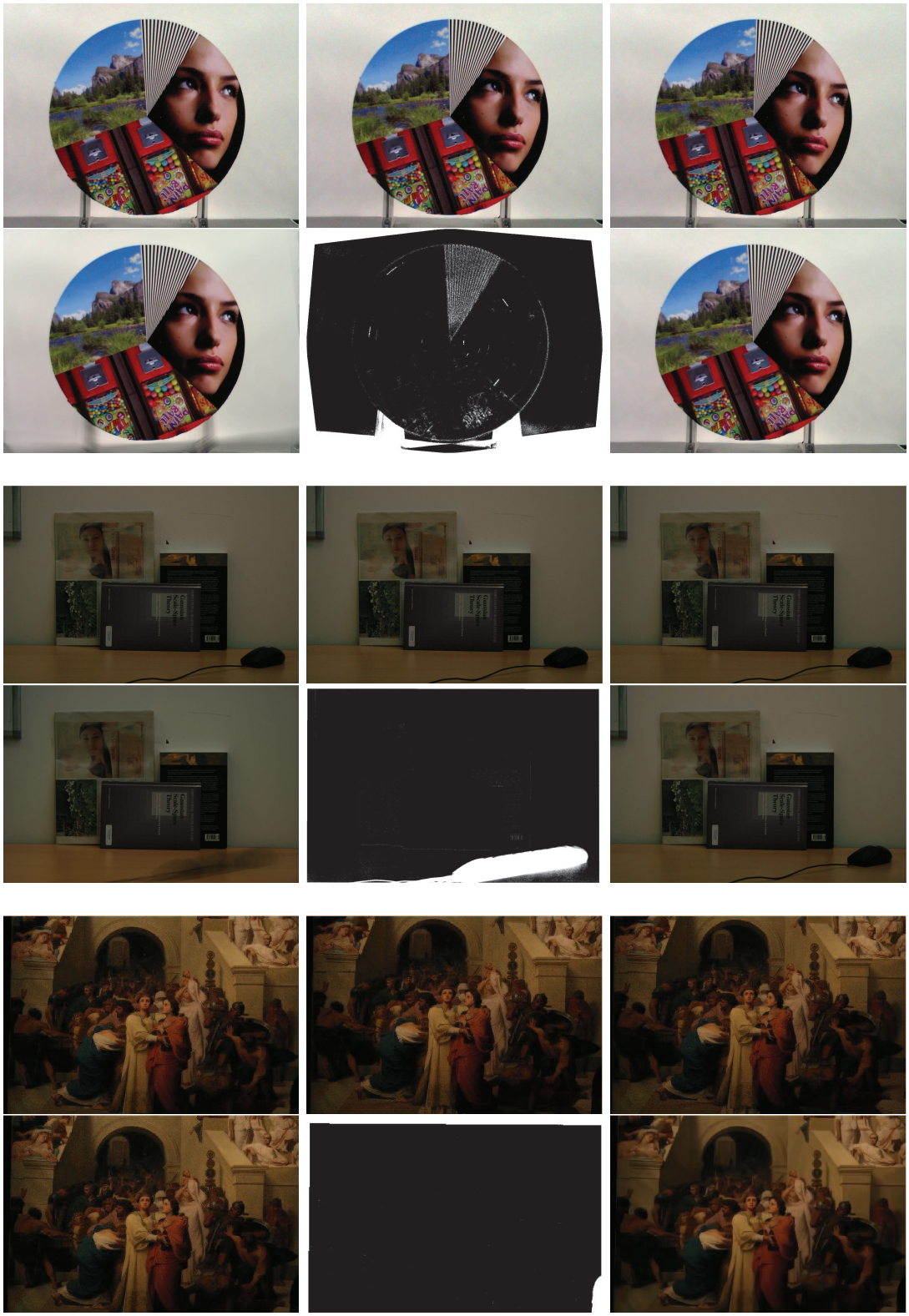


Figure 9.7: In each double row: three images of a sequence in the first row. In the second row on the left the average after registration, in the middle the mask of points with a too large temporal standard deviation, and on the right the restored image by hybrid method. These experiments illustrate how the hybrid method detects and corrects the potential wrong registrations due to local errors in the global homography.

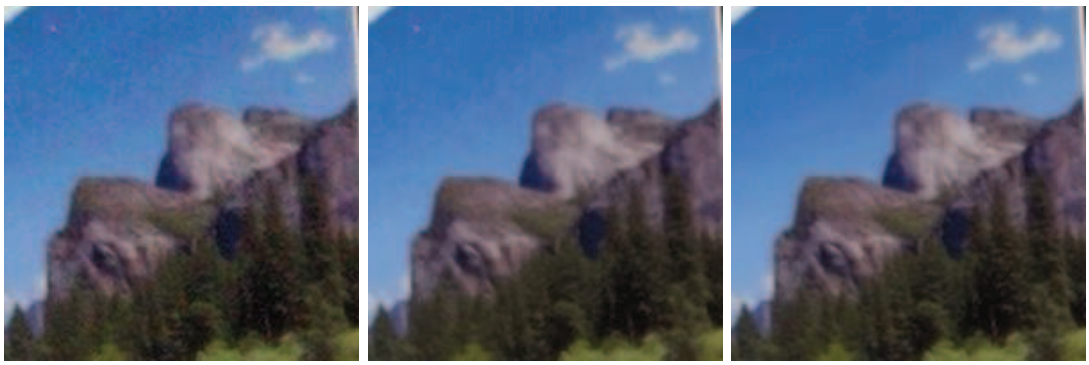


Figure 9.8: Detail from image in Fig. 9.7. From left to right: original image, NL-means (BM3D gives a similar result) and hybrid AAR. The images may need to be zoomed in on a screen to compare details and textures. Compare the fine texture details in the trees and the noise in the sky.

Figure 9.9: Detail from image in Fig. 9.7. From left to right: original image, BM3D (considered the best state of the art video denoiser) and AAR. The images are displayed after their color values have been stretched to $[0, 255]$. The images may need to be zoomed in on a screen to compare details and textures. Notice how large color spots due to the demosaicking and to JPEG have been corrected in the final result.

Figure 9.10: Detail from image in Fig. 9.7. From left to right: original image, BM3D (considered the best state of the art video denoiser) and AAR. Images are displayed after their color values have been stretched to $[0, 255]$. The images may need to be zoomed in on a screen to compare details and textures. Compare details on the face and on the wall texture.

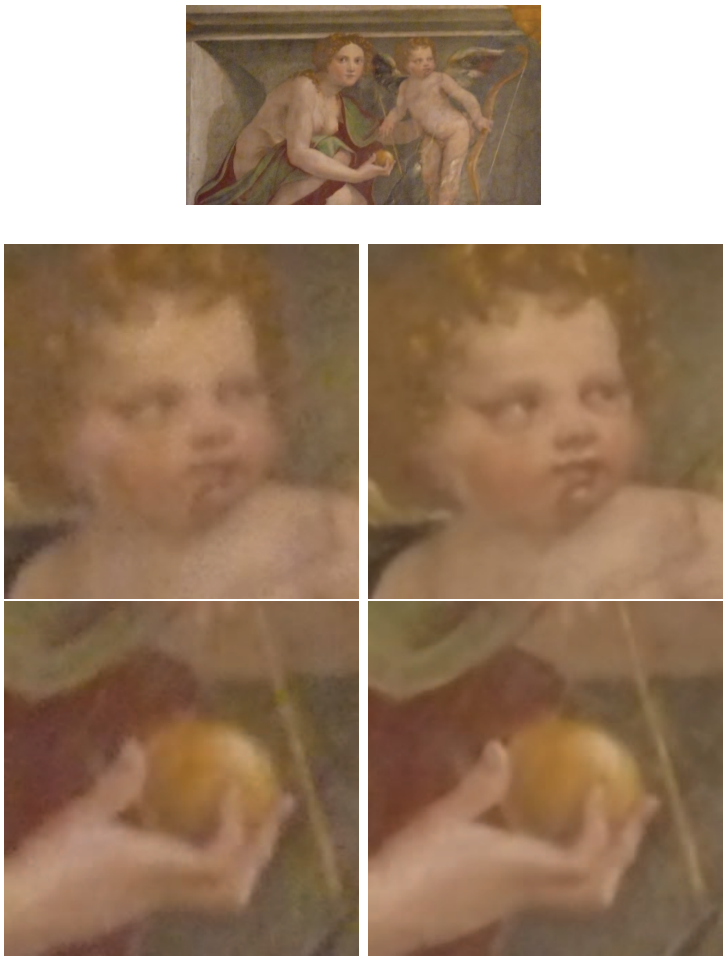


Figure 9.11: Top: initial image of the burst containing six images. Bottom: details on the initial and hybrid AAR images.

Chapter 10

Conclusion and Perspectives

Conclusion

This thesis has studied precision issues for two problems in stereo vision: lens distortion correction and correspondences between two images. Even though both problems are very fundamental, it seems that the precision aspect had not been extensively studied before.

The distortion correction is considered to have been solved since a long time. The literature shows a trend towards the automatic distortion correction from the information extracted from images without human intervention. But the precision of correction has always been considered enough for practical applications and has not been carefully measured. Traditionally, the distortion model is a part of the global camera calibration and is estimated together with the other camera internal/external parameters in a single minimization. As we have shown and explained, this kind of global camera calibration method suffers from error compensation. The extent of compensation depends on the distortion model and on the number of distortion parameters used in the global calibration. This drawback makes it difficult to use in practice. For 3D reconstruction, the residual distortion error will make the reconstructed scene also distorted even if the other camera parameters are well calibrated.

Our conclusion, somewhat contrary to the current practice, is that the distortion correction must be separated from the camera calibration to avoid error compensations. The non-parametric pattern-based method proposed in Chapter 4 is the first attempt to correct distortion. The advantage of this method is that it can accept any distortion between the pattern and the photo without demanding an adapted distortion model. An ideal correction is possible only if the pattern is completely flat. But this is never true in practice. This non-flatness error of the pattern introduces a systematic error in the estimated distortion field. Since neither the estimation of the shape of the pattern nor the fabrication of a completely flat pattern is easy in practice, we recurred to the plumb-line based method. Compared to the fabrication of a completely flat pattern, we discovered that it was much easier to ensure the straightness of the strings. Traditionally, the plumb-line method must be used together with a distortion model. The correction performance depends on whether the distortion model is adapted to the underlying distortion. So it is better to use a model which can approximate and correct different types of distortion. This is in fact the concept of self-consistency and universality discussed in Chapter 5. The polynomial model shows more universality than the other models and is chosen to be integrated in the plumb-line method in Chapter 6 to correct

the distortion. We consider the polynomial model as a non-parametric model because it is just a general approximation to the distortion without any prior structure. With a "calibration harp" carefully built by the fishing strings of good quality in Chapter 6, the plumb-line method can achieve a correction precision of about 0.02 pixels by the polynomial model of order 11.

The matching precision between two images is perhaps a more basic problem in image processing and computer vision. It is of particular interest for stereo vision when two images are taken in a wide baseline configuration. In contrast to the current research trend which tries to develop feature detectors invariant to geometric distortion, we were primarily interested in the matching precision. We took SIFT as the example to analyze and improve the matching precision. The extension to the other scale-invariant feature detectors can be envisaged. The burst denoising in Chapter 9 is one of the applications based on the precise image registration.

Perspectives

The story of precision never comes to an end and there is still a lot to exploit in the future. Several research directions are possible:

Complete Camera Calibration The distortion correction is just one part of camera calibration. The final aim is to calibrate the whole camera system and reconstruct the 3D scene in high precision. By separating the distortion correction from the global camera calibration, we envisage that the calibration of the other camera parameters will be more reliable after correcting the distortion. Due to the lack of ground truth, we will use the cross-validation to verify the stability and precision of the estimated parameters. For the intrinsic parameters of the camera, we can study their variance in several tests which do not share the same images. For the position of the camera, a similar cross-validation can be performed by studying the positional variance of a common camera shared in several tests. Finally, the reconstructed 3D scene can also be cross-validated by registering several 3D point clouds reconstructed from images taken from different views. This precision specification is a preliminary step to perform 3D points merging. Then the 2D image-based stereo technique can be compared to the result produced by 3D laser scanner.

Fully Automatic Camera Self-Calibration Traditional camera calibration requires much manual work in fabricating patterns and taking images. In addition, the camera must be re-calibrated if its parameters (like zoom or focus) are changed. This is a big limitation for many real-time stereo applications, like automatic vehicle navigation or real-time stereo from video. Camera self-calibration is a technique which can calibrate the camera from several images automatically without any pattern. Even though this technique is very useful, it suffers from numerical instabilities and geometric degeneracy. Its precision is not yet at the same level as the calibration based on a pattern. The theory of self-calibration is all about pinhole cameras. So it is not easy to integrate non-linear lens distortion in it. Some iterative approach can be designed to first do self-calibration by ignoring the distortion, then to refine the calibration result by integrating the distortion model. But it is doubtful that it can work well in the presence of a large distortion. A more direct approach is to integrate the distortion into the self-calibration. Some efficient minimization algorithm is needed to solve

this non-linear problem. It is a completely open problem, but very promising because using a camera on video streams or on many photographs should give information redundant enough to attain a highly accurate calibration.

Non-Rigid Image Registration Classic multi-view geometry is established under the assumption that the scene is static. If the scene is deformable, the whole framework should be rewritten. For 3D deformable scene reconstruction, the non-rigid image registration is the cornerstone. The methods for non-rigid image registration can be classified into two categories: non-parametric optic flow based method and parametric feature-based methods. The general opinion is that the non-parametric optical flow method with an appropriate regularization is the best choice to do non-rigid registration. Feature-based parametric methods suffer from few matchings in some homogeneous regions of image. Nonetheless, it is still worth trying Lowe's SIFT algorithm [117] to register non-rigid images even if its performance is limited. The sparse stable matchings found by SIFT can be used to approximate a rough deformation field, either by non-parametric method just like in [80], or parametric model fitting by using polynomial model [81], radial basis function [142] or thin-plate spline [19]. By taking one image as reference, the other image can be transformed by the estimated deformation field. Then SIFT can be applied again to find more matchings to estimate a more precise deformation field. This process can be iterated until that the performance becomes stable.

Another way to do non-rigid registration is to invent a new version of SIFT algorithm which is more robust against the non-rigid deformation. Recently ASIFT [131] was invented to compare two images in a full affine invariant manner. ASIFT synthesizes new images by sampling in the affine transformation space by simulating six parameters of camera position. These images are again compared by the SIFT method to make the algorithm affine invariant. In case of non-rigid transformation, the question is how to sample the space of non-rigid transformations with an acceptable complexity. To answer this question, a study about the generic deformation model is required.

It is also possible to construct a more robust descriptor which is invariant under non-rigid deformation. The SIFT descriptor is one of the most successful descriptors based on the histogram of gradient information. It is an ordered 128-vector which requires an accurate estimate of principal gradient orientations. But for non-rigid deformation, it does not help much to construct the descriptor along the principal gradient orientation, because the local transformation can be much more complicated than a rotation. A descriptor could be more efficiently constructed by the principal component analysis (PCA) of the local image patch. MSER (Maximally Stable Extremal Regions) [96] gives an idea about how to determine the size or the shape of the patch.

Appendix A

Appendix

A.1 Cross Products

The 3×3 skew-symmetric (anti-symmetric) matrix are very useful in multi-view geometry. If $\mathbf{a} = (a_1, a_2, a_3)^T$ is a 3-vector, then the corresponding skew-symmetric matrix is defined as follows:

$$[\mathbf{a}]_\times = \begin{pmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{pmatrix} \tag{A.1}$$

The matrix $[\mathbf{a}]_\times$ is singular, and $\mathbf{a}$ is its null-vector(right and left). The cross product is related to skew-symmetric matrix by:

$$\mathbf{a} \times \mathbf{b} = [\mathbf{a}]_\times \mathbf{b} = \left(\mathbf{a}^T [\mathbf{b}]_\times\right)^T . \tag{A.2}$$

If $\mathbf{M}$ is any 3×3 matrix (invertible or not), and $\mathbf{x}$ and $\mathbf{y}$ are column vectors, then

$$(\mathbf{M}\mathbf{x}) \times (\mathbf{M}\mathbf{y}) = \mathbf{M}^* (\mathbf{x} \times \mathbf{y}) \tag{A.3}$$

with $\mathbf{M}^*$ the matrix of cofactors of $\mathbf{M}$, satisfying $\mathbf{M}^*\mathbf{M} = \mathtt{det}(\mathbf{M})$. This equation car be written as:

$$[\mathbf{M}\mathbf{x}]_\times \mathbf{M} = \mathbf{M}^* [\mathbf{x}]_\times . \tag{A.4}$$

Furthermore, for any vector $\mathbf{t}$ and non-singular matrix $\mathbf{M}$, one has:

$$[\mathbf{t}]_\times \mathbf{M} = \mathbf{M}^* \left[\mathbf{M}^{-1}\mathbf{t}\right]_\times = \mathbf{M}^{-T} \left[\mathbf{M}^{-1}\mathbf{t}\right]_\times . \tag{A.5}$$

The cross product has the important property:

$$(\mathbf{a} \times \mathbf{b})^T \mathbf{c} = |\mathbf{a} \quad \mathbf{b} \quad \mathbf{c}|, \tag{A.6}$$

(determinant of the matrix composed of the vectors as columns). This is actually the definition of the cross product as the function

$$\mathbf{c} \to \varphi(\mathbf{c}) = |\mathbf{a} \quad \mathbf{b} \quad \mathbf{c}| \tag{A.7}$$

is a linear form, which can be expressed as a scalar product with a fixed vector, namely $\mathbf{a} \times \mathbf{b}$.

A.2 Singular Value Decomposition (SVD)

The SVD is a very useful decomposition of a matrix generalizing the diagonalization of a symmetric matrix in an orthonormal frame, but valid for any matrix, even rectangular ones.

First, consider a symmetric matrix $\mathbf{A}$. The quadratic form

$$\mathbf{x} \to \frac{1}{2}\mathbf{x}^T\mathbf{A}\mathbf{x} \tag{A.8}$$

is continuous and reaches its maximum when restricted on the sphere $\mathbf{x}^T\mathbf{x} = 1$. The Lagrangian of this optimization problem can be written:

$$L(\mathbf{x}, \lambda) = \frac{1}{2}\mathbf{x}^T\mathbf{A}\mathbf{x} - \lambda(\mathbf{x}^T\mathbf{x} - 1) \tag{A.9}$$

and has all partial derivatives 0 when reaching the maximum at unit vector $\mathbf{x}_1$. Thus we get

$$\mathbf{A}\mathbf{x}_1 = \lambda\mathbf{x}_1 \tag{A.10}$$

meaning $\mathbf{x}_1$ is an eigenvector of A.

Now if $\mathbf{x}_1^T\mathbf{y} = 0$, we have

$$\mathbf{x}_1^T(\mathbf{A}\mathbf{y}) = (\mathbf{A}\mathbf{x}_1)^T\mathbf{y} = \lambda\mathbf{x}_1^T\mathbf{y} = 0, \tag{A.11}$$

the first equality using the symmetry of $\mathbf{A}$. This shows that the subspace orthogonal to $\mathbf{x}_1$ is globally preserved by $\mathbf{A}$ and an easy recursion argument shows that we can get an orthonormal basis of eigenvectors of $\mathbf{A}$.

If $\mathbf{B}$ is an $m \times n$ matrix with $m \geq n$, $\mathbf{A} = \mathbf{B}^T\mathbf{B}$ is symmetric and can be written

$$\mathbf{A} = \mathbf{V}\begin{pmatrix} s_1 & & \\ & \ddots & \\ & & s_n \end{pmatrix}\mathbf{V}^T \tag{A.12}$$

with $\mathbf{V}$ an $n \times n$ orthonormal matrix and all diagonal terms are positive since $\mathbf{A}$ is positive. Define $\mathbf{U}_i = \mathbf{B}\mathbf{V}_i/\|\mathbf{B}\mathbf{V}_i\|$ for all i such that $\mathbf{B}\mathbf{V}_i \neq 0$. For two indices i, j, we have

$$(\mathbf{B}\mathbf{V}_i)^T(\mathbf{B}\mathbf{V}_j) = \mathbf{V}_i^T\mathbf{A}\mathbf{V}_j = s_j\mathbf{V}_i^T\mathbf{V}_j = s_j\delta_{ij}. \tag{A.13}$$

So the $\mathbf{U_i}$ form an orthonormal family, which can be completed to get n orthonormal vectors of $\mathbb{R}^m$ if necessary, as $m \geq n$. Then we have

$$\mathbf{B} = \mathbf{U}\begin{pmatrix} \sqrt{s_1} & & \\ & \ddots & \\ & & \sqrt{s_n} \end{pmatrix}\mathbf{V}^T \tag{A.14}$$

since the two sides have the same images when applied to the basis $\mathbf{V}_i$. Indeed, calling $\mathbf{C}$ the right hand side of (A.14), on the one hand we have

$$\mathbf{C}\mathbf{V}_i = \sqrt{s_i}\mathbf{U}_i. \tag{A.15}$$

On the other hand,

$$\|\mathbf{B}\mathbf{V}_i\|^2 = \mathbf{V}_i^T \mathbf{A}\mathbf{V}_i = s_i. \tag{A.16}$$

Therefore, if $s_i = 0$, we have $\mathbf{C}\mathbf{V}_i = 0 = \mathbf{B}\mathbf{V}_i$. If $s_i \neq 0$, we have

$$\mathbf{C}\mathbf{V}_i = \sqrt{s_i}\frac{\mathbf{B}\mathbf{V}_i}{\|\mathbf{B}\mathbf{V}_i\|} = \mathbf{B}\mathbf{V}_i. \tag{A.17}$$

This proves that $\mathbf{B} = \mathbf{C}$.

Equation (A.14) is called the SVD of $\mathbf{B}$. U is an $m \times n$ matrix with orthonormal columns, V is an $n \times n$ rotation matrix and the $\sqrt{s_i}$ are called the singular values of B.

Exercice 1 *Show that the dimension of the kernel of* $\mathbf{B}$ *is the number of* s_i *that are* 0 *and that a basis of the kernel is formed by the corresponding* $\mathbf{V}_i$*.*

A.3 Levenberg-Marquardt Algorithm

Suppose a function $\mathbf{X} = \mathbf{f}(\mathbf{P})$ where $\mathbf{X}$ is a measurement vector and $\mathbf{P}$ is a parameter vector in $\mathbb{R}^N$ and $\mathbb{R}^M$ respectively. We want to find the vector $\hat{\mathbf{P}}$ satisfying $\mathbf{X} = \mathbf{f}(\hat{\mathbf{P}}) - \epsilon$ for which ϵ is minimized. If $\mathbf{f}$ is a linear function, this problem is a linear least-square problem. If $\mathbf{f}$ is not linear, we can start with an initial estimated value $\mathbf{P}_0$ and proceed to refine the estimate under the assumption that the function $\mathbf{f}$ is locally linear. Let $\epsilon_0 = \mathbf{f}(\mathbf{P}_0) - \mathbf{X}$. We assume that the function $\mathbf{f}$ is approximated by the Jacobian matrix $\mathbf{J} = \frac{\partial \mathbf{f}}{\partial \mathbf{P}}$. We want to find an update step $\mathbf{\Delta}$ to $\mathbf{P}_0$ such that with the updated parameter vector $\mathbf{P}_1 = \mathbf{P}_0 + \mathbf{\Delta}$, $\mathbf{f}(\mathbf{P}_1) - \mathbf{X} = \mathbf{f}(\mathbf{P}_0) + \mathbf{J}\mathbf{\Delta} - \mathbf{X} = \epsilon_0 + \mathbf{J}\mathbf{\Delta}$. This problem is in fact to minimize $\|\epsilon_0 + \mathbf{J}\mathbf{\Delta}\|$ over $\mathbf{\Delta}$, which is a linear minimization problem. The vector $\mathbf{\Delta}$ is solution to the normal equation:

$$\mathbf{J}^T\mathbf{J}\mathbf{\Delta} = -\mathbf{J}^T\epsilon_0 \tag{A.18}$$

This normal equation is in fact used in the Gauss-Newton algorithm. In the Levenberg-Marquardt algorithm, this normal equation is replaced by the augmented normal equations:

$$\left(\mathbf{J}^T\mathbf{J} + \lambda\,\mathrm{diag}(\mathbf{J}^T\mathbf{J})\right)\mathbf{\Delta} = -\mathbf{J}^T\epsilon \tag{A.19}$$

for some value of λ that varies from iteration to iteration and diag is an operator replacing non-diagonal elements of its argument by 0. A typical initial value of λ is 10^{-3}.

If the values of $\mathbf{\Delta}$ obtained by solving the augmented normal equations leads to a reduction in the error, then the increment is accepted and λ is divided by a factor (typically 10) before the next iteration. On the other hand if the value leads to an increased error, then λ is multiplied by the same factor and the augmented normal equations are solved again, this process continuing until a value of $\mathbf{\Delta}$ is found that gives rise to a decreased error. This process of repeatedly solving the augmented normal equations for different values of λ until an acceptable $\mathbf{\Delta}$ is found constitutes one iteration of the LM algorithm. When λ is small, the method is essentially the same as a Gauss-Newton iteration; on the other hand when λ is large, $\mathbf{\Delta}$ approaches the value given by the gradient descent. Thus the LM algorithm moves seamlessly between a Gauss-Newton iteration, which will cause rapid convergence in the neighborhood of the solution, and a gradient descent approach, which will guarantee a

decrease in the cost function when the progress is difficult. Indeed, when λ becomes larger and larger, the length of the increment step $\boldsymbol{\Delta}$ decreases and eventually leads to a decrease of the cost function.

In practice, the LM algorithm is completely specified by setting the initial value of λ to 10^{-3} and by setting the division or multiplication factor of λ in each iteration to be 10.

Bibliography

[1] Y.I. Abdel-Aziz and H.M. Karara. Direct linear transformation into object space coordinates in close-range photogrammetry. *Symposium on Close-Range Photogrammetry*, 37(1):1–18, 1971.

[2] Sergey K. Abramov, Vladimir V. Lukin, Benoit Vozel, Kacem Chehdi, and Jaakko T. Astola. Segmentation-based method for blind evaluation of noise variance in images. *Journal of Applied Remote Sensing*, 2(1), 2008.

[3] Luis Alvarez, Luis Gomez, and J. Rafael Sendra. Algebraic lens distortion model estimation. *Image Processing On Line*, 2010.

[4] Luis Alvarez, Luis Gomez, and Rafael Sendra. An algebraic approach to lens distortion by line rectification. *Journal of Mathematical Imaging and Vision*, 35(1):36–50, 2009.

[5] F. J. Anscomb. The transformation of poisson, binomial and negative-binomial data. *Biometrika*, 35(3):246–254, 1948.

[6] N. Ayache and C. Hansen. Rectification of images for binocular and trinocular stereovision. *International Conference on Pattern Recognition*, 1:11–16, 1988.

[7] L. Baboulaz and P. L. Dragotti. Exact feature extraction using finite rate of innovation principles with an application to image super-resolution. *IEEE Transactions on Image Processing*, 18(2):281–298, 2009.

[8] S. Baker and S.K. Nayar. Global measures of coherence for edge detector evaluation. *Conference on Computer Vision and Pattern Recognition*, 2:373–379, 1999.

[9] Simon Baker and Takeo Kanade. Limits on super-resolution and how to break them. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(9):1167–1183, 2002.

[10] J.P. Barreto and K. Daniilidis. Fundamental matrix for cameras with radial distortion. *International Conference on Computer Vision*, 1:625–632, 2005.

[11] B. Bascle, Andrew Blake, and Andrew Zisserman. Motion deblurring and super-resolution from an image sequence. In *European Conference on Computer Vision*, pages 573–582, 1996.

[12] Bénédicte Bascle and Rachid Deriche. Region tracking through image sequences. In *International Conference on Computer Vision*, pages 302–307, 1995.

[13] S. Battiato, G. Gallo, G. Puglisi, and S. Scellato. Sift features tracking for video stabilization. In *International Conference on Image Analysis and Processing*, pages 825–830, 2007.

[14] R. Baumann and W. B. Seales. Robust registration of manuscript images. In *ACM/IEEE-CS joint conference on Digital libraries*, pages 263–266, New York, NY, USA, 2009. ACM.

[15] James R. Bergen, P. Anandan, Keith J. Hanna, and Rajesh Hingorani. Hierarchical model-based motion estimation. In *European Conference on Computer Vision*, pages 237–252, 1992.

[16] M. Bertalmio and S. Levine. Fusion of bracketing pictures. In *Conference for Visual Media Productio*, 2009.

[17] Christopher M. Bishop, Andrew Blake, and Bhaskara Marthi. Super-resolution enhancement of video. In *International Conference on Artificial Intelligence and Statistics*, 2003.

[18] M.J. Black and G. Sapiro. Edges as outliers: Anisotropic smoothing using local image statistics. *Lecture notes in computer science*, pages 259–270, 1999.

[19] F.L. Bookstein. Principal warps: Thin-plate splines and the decomposition of deformations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(6):567–585, 1989.

[20] Jean-Yves Bouguet. Camera calibration toolbox for matlab, http://www.vision.caltech.edu/bouguetj/calib_doc/index.html.

[21] Jean-Yves Bouguet. Camera calibration from points and lines in dual-space geometry. Technical report, California Institute of Technology, 1998.

[22] Jean-Yves Bouguet. Closed-form camera calibration in dual-space geometry. Technical report, California Institute of Technology, 1998.

[23] F. Bretar, M. Chesnier, M. Roux, and M. Pierrot-Deseilligny. Analyse quantitative de données laser 3d : Classification et modélisation du terrain. *Revue Française de Photogrammétrie et de Télédétection*, 176:21–29, 2004.

[24] M.J. Brooks, W. Chojnacki, D. Gawley, and A. van den Henge. What value covariance information in estimating vision parameters? *International Conference on Computer Vision*, 1, 2001.

[25] D. C. Brown. Close-range camera calibration. *Photogrammetric Engineering*, 37(8):855–866, 1971.

[26] D.C. Brown. Decentering distortion of lenses. *Photogrammetric Engineering*, 32(3):444–462, 1966.

[27] M. Brown, R. Szeliski, and S. Winder. Multi-image matching using multi-scale oriented patches. *Conference on Computer Vision and Pattern Recognition*, pages 510–517, 2005.

[28] Matthew Brown and David G. Lowe. Automatic panoramic image stitching using invariant features. *International Journal of Computer Vision*, pages 59–73, 2007.

[29] A. Buades, B. Coll, and J. M. Morel. A review of image denoising algorithms, with a new one. *SIAM Multiscale Modeling Simulation*, 4(2):490–530, 2005.

[30] A. Buades, B. Coll, and J.M. Morel. Nonlocal image and movie denoising. *International Journal of Computer Vision*, 76(2):123–139, 2008.

[31] T. Buades, Y. Lou, J.-M. Morel, and Z. Tang. A note on multi-image denoising. *International workshop on Local and Non-Local Approximation in Image Processing*, pages 1–15, August 2009.

[32] M. Byrod, Z. Kukelova, K. Josephson, T. Pajdla, and K. Astrom. Fast and robust numerical solutions to minimal problems for cameras with radial distortion. *Computer Vision and Image Understanding*, 114(2):1–8, 2008.

[33] J. F. Canny. Finding edges and lines in images. *Technical Report AI-TR-720, Massachusets Institute of Technology, Artificial Intelligence Laboratory*, 1983.

[34] D. Capel and A. Zisserman. Super-resolution from multiple views using learnt image models. In *Conference on Computer Vision and Pattern Recognition*, volume 2, pages II–627–II–634, 2001.

[35] David Capel and Andrew Zisserman. Automated mosaicing with super-resolution zoom. In *Conference on Computer Vision and Pattern Recognition*, pages 885–891, 1998.

[36] C. Chevalier, G. Roman, and J.N. Niepce. *Guide du photographe.* C. Chevalier, 1854.

[37] T. A. Clarke and J. G. Fryer. The development of camera calibration methods and models. *The Photogrammetric Record*, 16:51–66, 1998.

[38] D. Claus and A. W. Fitzgibbon. A plumbline constraint for the rational function lens distortion model. *British Machine Vision Conference*, pages 99–108, 2005.

[39] D. Claus and A.W. Fitzgibbon. A rational function lens distortion model for general cameras. *IEEE Computer Vision and Pattern Recognition*, 1:213–219, 2005.

[40] Christopher Coelho, Aaron Heller, Joseph L. Mundy, David A. Forsyth, and Andrew Zisserman. An experimental evaluation of projective invariants. *Mit Press Series Of Artificial Intelligence Series*, pages 87–104, 1992.

[41] K. Dabov, A. Foi, and K. Egiazarian. Video denoising by sparse 3d transform-domain collaborative filtering. In *European Signal Processing Conference*, 2007.

[42] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian. Color image denoising via sparse 3d collaborative filtering with grouping constraint in luminance-chrominance space. In *International Conference on Image Processing*, 2007.

[43] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian. Image denoising by sparse 3d transform-domain collaborative filtering. *IEEE Transations on Image Processing*, 16(8), 2007.

[44] A. Danielyan, A. Foi, V. Katkovnik, and K. Egiazarian. Image and video super-resolution via spatially adaptive block-matching filtering. In *International Workshop on Local and Non-Local Approximation in Image Processing (LNLA)*, 2008.

[45] Aram Danielyan and A. Foi. Noise variance estimation in nonlocal transform domain. In *International Workshop on Local and Non-Local Approximation in Image Processing (LNLA)*, 2009.

[46] Dmitry Datsenko and Michael Elad. Example-based single document image super-resolution: a global map approach with outlier rejection. In *Multidimensional Systems and Signal Processing*, number 18, pages 103–121, 2007.

[47] J. Delon. Midway image equalization. *Journal of Mathematical Imaging and Vision*, 21(2):119–134, 2004.

[48] J. Delon. Movie and video scale-time equalization application to flicker reduction. *IEEE Transactions on Image Processing*, 15(1):241–248, Jan. 2006.

[49] R. Deriche. Using canny's criteria to derive a recursively implemented optimal edge detector. *International Journal of Computer Vision*, 1(2):167–187, 1987.

[50] A. Desolneux, L. Moisan, and J.-M. Morel. Meaningful alignments. *International Journal of Computer Vision*, 40(1):7–23, 2000.

[51] A. Desolneux, L. Moisan, and J.-M. Morel. *From Gestalt Theory to Image Analysis, a Probabilistic Approach*. Springer, 2008.

[52] F. Devernay. A non-maxima suppression method for edge detection with sub-pixel accuracy. Technical Report 2724, INRIA rapport de recherche, 1995.

[53] F. Devernay and O. Faugeras. Straight lines have to be straight. *Machine Vision and Applications*, 13:14–24, 2001.

[54] D. Donoho and J. Johnstone. Ideal spatial adaption via wavelet shrinkage. *Biometrika*, 81(3):425–455, 1994.

[55] David Donoho and Iain M. Johnstone. Adapting to unknown smoothness via wavelet shrinkage. *Journal of the American Statistical Association*, 90:1200–1224, 1995.

[56] M. Ebrahimi and E.R. Vrscay. Multi-frame super-resolution with no explicit motion estimation. In *International Conference on Image Processing, Computer Vision, and Pattern Recognition*, 2008.

[57] Mehran Ebrahimi and Edward Vrscay. Solving the inverse problem of image zooming using "self-examples". *Image Analysis and Recognition*, pages 117–130, 2007.

[58] E. Eisemann and F. Durand. Flash photography enhancement via intrinsic relighting. *ACM Transactions on Graphics*, 23(3):673–678, 2004.

[59] Michael Elad and Arie Feuer. Super-resolution reconstruction of continuous image sequences. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21:459–463, 1999.

[60] W. Faig. Calibration of close-range photogrammetric systems: Mathematical formulation. *Photogrammetric Engieering Remote Sensing*, 41(12):1479–1486, 1975.

[61] Hany Farid and Alin C. Popescu. Blind removal of lens distortion. *Journal of the Optical Society of America*, 2001.

[62] S. Farsiu, M. Elad, and P. Milanfar. Multiframe demosaicing and super-resolution of color images. *IEEE Transactions on Image Processing*, 15(1):141–159, Jan. 2006.

[63] Sina Farsiu, Dirk Robinson, Michael Elad, and Peyman Milanfar. Fast and robust multi-frame super-resolution. *IEEE Transactions on Image ProcessinG*, 13:1327–1344, 2003.

[64] Raanan Fattal, Maneesh Agrawala, and Szymon Rusinkiewicz. Multiscale shape and detail enhancement from multi-light image collections. In *ACM SIGGRAPH*, page 51, 2007.

[65] O. Faugeras. *Three-Dimensional Computer Vision: A Geometric Viewpoint*. The MIT Press, 1993.

[66] O.D. Faugeras and G. Toscani. The calibration problem for stereo. *Conference on Computer Vision and Pattern Recognition*, pages 15–20, 1986.

[67] M.A. Fischler and R.C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24:381–395, 1981.

[68] A. Fitzgibbon. Simultaneous linear estimation of multiple view geometry and lens distortion. *Conference on Computer Vision and Pattern Recognition*, 1:125–132, 2001.

[69] A. Foi, S. Alenius, V. Katkovnik, and K. Egiazarian. Noise measurement for raw-data of digital imaging sensors by automatic segmentation of non-uniform targets. *IEEE Sensors Journal*, 7(10):1456–1461, 2007.

[70] A. Foi, M. Trimeche, V. Katkovnik, and K. Egiazarian. Practical poissonian-gaussian noise modeling and fitting for single image raw-data. *IEEE Transaction on Image Processing*, 17(10):1737–1754, 2008.

[71] David A. Forsyth and Jean Ponce. *Computer Vision: A Modern Approach*. Prentice Hall, 2002.

[72] Jan-Michael Frahm and Marc Pollefeys. Ransac for (quasi-)degenerate data (qdegsac). *Conference on Computer Vision and Pattern Recognition*, 1, 2006.

[73] William T. Freeman, Thouis R. Jones, and Egon C Pasztor. Example-based super-resolution. *IEEE Computer Graphics and Applications*, 22:56–65, 2002.

[74] F. Fuchs, H. Jibrini, G. Maillet, N. Paparoditis, M. Pierrot-Deseilligny, and F. Taillandier. Trois approches pour la construction automatique de modèles 3d de bâtiments en imagerie aérienne haute résolution. *Revue Française de Photogrammétrie et de Télédétection*, 166:10–18, 2002.

[75] A. Fusiello and L. Irsara. Quasi-euclidean uncalibrated epipolar rectification. *International Conference on Pattern Recognition*, pages 1–4, 2008.

[76] A. Fusiello, E. Trucco, and A. Verri. A compact algorithm for rectification of stereo pairs. *Machine Vision and Applications*, 12(1):16–22, 2000.

[77] S. Ganapathy. Decomposition of transformation matrices for robot vision. *IEEE International Conference on Robotics and Automation*, pages 130–139, 1984.

[78] J. Gluckman and S.K. Nayar. Rectifying transformations that minimize resampling effects. *Conference on Computer Vision and Pattern Recognition*, 2001.

[79] A. Goldenshluger and A. Nemirovski. On spatial adaptive estimation of nonparametric regression. *Mathematical Methods of Statistics*, 6:1737–1754, 1997.

[80] R. Grompone von Gioi, P. Monasse, J.-M. Morel, and Z. Tang. Towards high-precision lens distortion correction. *International Conference on Image Processing*, pages 4237–4240, 2010.

[81] R. Grompone von Gioi, P. Monasse, J.-M. Morel, and Z. Tang. Lens distortion correction with a calibration harp. *Preprint*, 2011.

[82] Jiang Guang and Quan Long. Detection of concentric circles for camera calibration. *International Conference on Computer Vision*, pages 333–340, 2005.

[83] Pierre Gurdjos and Peter Sturm. Methods and geometry for plane-based self-calibration. *Conference on Computer Vision and Pattern Recognition*, pages 491–496, 2003.

[84] E.L. Hall, J.B.K. Tio, C.A. McPherson, and F.A. Sadjadi. Measuring curved surfaces for robot vision. *Computer Journal*, 15(12):42–54, 1982.

[85] R. Hartley and S. B. Kang. Parameter-free radial distortion correction with center of distortion estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1309-1321, 2007.

[86] R. I. Hartley and T. Saxena. The cubic rational polynomial camera model. *DARPA Image Understanding Workshop*, pages 649–653, 1997.

[87] R.I. Hartley. In defense of the eight-point algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(6):580–593, 1997.

[88] R.I. Hartley. Theory and practice of projective rectification. *International Journal of Computer Vision*, 35(2):115–127, 1999.

[89] R.I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004.

[90] G.E. Healey and R. Kondepudy. Radiometric ccd camera calibration and noise estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(3):267–276, 1994.

[91] Janne Heikkilä and Olli Silvén. A four-step camera calibration procedure with implicit image correction. *Conference on Computer Vision and Pattern Recognition*, 1997.

[92] Marko Heikkilä and Matti Pietikäinen. An image mosaicing module for wide-area surveillance. In *ACM international workshop on Video surveillance & sensor networks*, pages 11–18, New York, NY, USA, 2005. ACM.

[93] A. Heyden and K. Rohr. Evaluation of corner extraction schemes using invariance methods. *International Conference on Pattern Recognition*, 13:895–899, 1996.

[94] Michal Irani and Shmuel Peleg. Motion analysis for image enhancement: Resolution, occlusion, and transparency. *Journal of Visual Communication and Image Representation*, 4:324–335, 1993.

[95] Lavest J., Viala M., and Dhome M. Do we really need accurate calibration pattern to achieve a reliable camera calibration? *European Conference on Computer Vision*, 1:158–174, 1998.

[96] M. Urba J. Matas, O. Chum and T. Pajdla. Robust wide baseline stereo from maximally stable extremal regions. *British Machine Vision Conference*, pages 384–396, 2002.

[97] K. Josephson and M. Byrod. Pose estimation with radial distortion and unknown focal length. *Conference on Computer Vision and Pattern Recognition*, pages 2419–2426, 2009.

[98] Y. Kanazawa and K. Kanatani. Do we really have to consider covariance matrices for image features? *International Conference on Computer Vision*, 2, 2001.

[99] J. Kannala and S. S. Brandt. A generic camera model and calibration method for conventional, wide-angle, and fish-eye lenses. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28:1335–1340, 2006.

[100] C. Kervrann and J. Boulanger. Local adaptivity to variable smoothness for exemplar-based image regularization and representation. *International Journal of Computer Vision*, 79(1):45–69, 2008.

[101] E. Kilpelä. Compensation of systematic errors of image and model coordinates. *Photogrammetria*, 37(1):15–44, 1980.

[102] E. D. Kolaczyk. Wavelet shrinkage estimation of certain poisson intensity signals using corrected thresholds. *Statistica Sinica*, 9:119–135, 1999.

[103] Z. Kukelova and T. Pajdla. A minimal solution to the autocalibration of radial distortion. *Conference on Computer Vision and Pattern Recognition*, pages 1–7, 2007.

[104] Z. Kukelova and T. Pajdla. Two minimal problems for cameras with radial distortion. *IEEE International Conference on Computer Vision*, pages 1–8, 2007.

[105] F. Lafarge, X. Descombes, J. Zerubia, and M. Pierrot-Deseilligny. Modèle paramétrique pour la reconstruction automatique en 3d de zones urbaines denses à partir d'images satellitaires haute résolution. *Revue Française de Photogrammétrie et de Télédétection (SFPT)*, 180:4–12, 2005.

[106] Stamatios Lefkimmiatis, Petros Maragos, and George Papandreou. Bayesian inference on multiscale models for poisson intensity estimation: Application to photo-limited image denoising. *IEEE Transactions on Image Processing*, 18(8):1724–1741, 2009.

[107] K. Levenberg. A method for the solution of certain non-linear problems in least squares. *The Quarterly of Applied Mathematics*, 2:164–168, 1944.

[108] Hongdong Li and Richard Hartley. A non-iterative method for correcting lens distortion from nine point correspondences. *OmniVision*, 2005.

[109] Yanfang Li, Yaming Wang, Wenqing Huang, and Zuoli Zhang. Automatic image stitching using sift. In *International Conference on Audio, Language and Image Processing (ICALIP)*, pages 568–571, July 2008.

[110] C. Liu, R. Szeliski, S.B. Kang, C.L. Zitnick, and W.T. Freeman. Automatic estimation and removal of noise from a single image. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(2):299–314, 2008.

[111] Ce Liu, William T. Freeman, Richard Szeliski, and Sing Bing Kang. Noise estimation from a single image. *Conference on Computer Vision and Pattern Recognition*, 1:901–908, 2006.

[112] Ce Liu, Jenny Yuen, and Antonio Torralba. Sift flow: Dense correspondence across scenes and its applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2010.

[113] Y. Liu, A. Al-Obaidi, A. Jakas, and L. Li. Accurate camera calibration and correction using rigidity and radial alignment constraints. *International Symposium on 3D Data Processing, Visualization and Transmission*, pages 145–152, 2008.

[114] Patricio Loncomilla and Javier Ruiz del solar. Improving sift-based object recognition for robot applications. *Lecture Notes in Computer Science*, pages 1084–1092, 2005.

[115] Quan Long. *Image-Based Modeling.* Springer, 2010.

[116] C. Loop and Z. Zhang. Computing rectifying homographies for stereo vision. *Conference on Computer Vision and Pattern Recognition*, 1:125–131, 1999.

[117] David G Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[118] H. Lu, Y. Kim, and J.M.M. Anderson. Improved poisson intensity estimation: denoising application using poisson data. *IEEE Transations on Image Processing*, 13(8):1128–1135, Aug. 2004.

[119] L. Ma, Y. Chen, and K. L. Moore. Rational radial distortion models of camera lenses with analytical solution for distortion correction. *International Journal Information Acquisition*, 1(2):135–147, 2004.

[120] Y. Ma, S. Soatto, J. Kosecka, and S. Sastry. *An Invitation to 3D Vision*. Springer, 2003.

[121] M. Makitalo and A. Foi. On the inversion of the anscombe transformation in low-count poisson image denoising. In *International Workshop on Local and Non-Local Approximation in Image Processing (LNLA)*, 2009.

[122] J. Mallon and Paul F. Whelan. Projective rectification from the fundamental matrix. *Image and Vision Computing*, 23:643–650, 2005.

[123] Donald Marquardt. An algorithm for least-squares estimation of nonlinear parameters. *SIAM Journal on Applied Mathematics*, 11:431–441, 1963.

[124] Antonio Marquina and S. Osher. Image super-resolution by tv-regularization and bregman iteration. *Journal of Scientific Computing*, 37(3):367–382, 2008.

[125] T. Melen. *Geometrical modelling and calibration of video cameras for underwater navigation. Dr. Ing Thesis*. PhD thesis, Norge tekniske høgskole, Institutt for teknisk kybernetikk.

[126] B. Micusik and T. Pajdla. Estimation of omnidirectional camera model from epipolar geometry. *Conference on Computer Vision and Pattern Recognition*, 1:485, 2003.

[127] Krystian Mikolajczyk and Cordelia Schmid. An affine invariant interest point detector. *European Conference of Computer Vision*, pages 128–142, 2002.

[128] Krystian Mikolajczyk and Cordelia Schmid. Scale and affine invariant interest point detectors. *International Journal of Computer Vision*, 60(1):63–86, 2004.

[129] L. Moisan and B. Stival. A probabilistic criterion to detect rigid point matches between two images and estimate the fundamental matrix. *International Journal of Computer Vision*, 57:201–218, 2004.

[130] M. Moradi, P. Abolmaesumi, and P. Mousavi. Deformable registration using scale space keypoints. In *SPIE Medical Imaging 2006: Image Processing*, volume 6144, pages 61442G1– 61442G8, 2006.

[131] J.M. Morel and G.Yu. Asift: A new framework for fully affine invariant image comparison. *SIAM Journal on Imaging Sciences*, 2(2):438–469, 2009.

[132] J.M. Morel and G.Yu. On the consistency of the sift method. *Preprint, CMLA, ENS-Cachan*, 2009.

[133] P. Musé, F. Sur, F. Cao, Y. Gousseau, and J.-M. Morel. An *A Contraio* decision method for shape element recognition. *International Journal of Computer Vision*, 69(3):295–315, 2006.

[134] James Nagy and Zdenek Strakos. Enforcing nonnegativity in image reconstruction algorithms. In *SPIE Mathematical Modeling Estimation and Imaging*, pages 182–190, 2000.

[135] D. Nistér. Preemptive ransac for live structure and motion estimation. *Machine Vision and Applications*, 16(5):321–329, 2005.

[136] Robert D. Nowak and Richard G. Baraniuk. Wavelet-domain filtering for photon imaging systems. *IEEE Transactions on Image Processing*, 8(5):666–678, 1997.

[137] Chum O., Urban M., Matas J., and Pajdla T. Robust wide baseline stereo from maximally stable extremal regions. *British Machine Vision Conference*, pages 384–396, 2002.

[138] American Society of Photogrammetry. *Manual of Photogrammetry*. Asprs Pubns, 1980.

[139] S. I. Olsen. Estimation of noise in images: an evaluation. *CVGIP: Graph. Models Image Process*, 55(4):319–323, 1993.

[140] Daniel Oram. Rectification for any epipolar geometry. *British Machine Vision Conference*, 2001.

[141] U. Orguner and F. Gustafsson. Statistical characteristics of harris corner detector. *IEEE Workshop on Statistical Signal Processing (SSP)*, page 571–575, 2007.

[142] M. J. L. Orr. Introduction to radial basis function networks. *Technical report, University of Edinburgh*, 1996.

[143] Brand P. and Mohr R. Accuracy in image measure. *SPIE Conference on Videometrics III*, 2350:218–228, 1994.

[144] T. Pajdla, Z. Kukelova, and M. Bujnak. Automatic generator of minimal problem solvers. *European Conference on Computer Vision*, pages 302–315, 2008.

[145] Tomáš Pajdla, Tomáš Werner, and Václav Hlaváč. Correcting radial lens distortion without knowledge of 3-d structure. *Research Report, Czech Technical University*, 1997.

[146] T. Papadopoulo and M.I.A. Lourakis. Estimating the jacobian of the singular value decomposition:theory and application. *European Conference on Computer Vision*, 1:554–570, 2000.

[147] M. Pierrot-Deseilligny and I. Cléry. Apero, an open source bundle adjusment software for automatic calibration and orientation of a set of images. *SPRS Commission V Symposium, Image Engineering and Vision Metrology*, 2011.

[148] M. Pollefeys, R. Koch, and L. Van Gool. A simple and efficient rectification method for general motion. *International Conference on Computer Vision*, 1:496–501, 1999.

[149] N. N. Ponomarenko, V. V. Lukin, S. K. Abramov, K. O. Egiazarian, and J. T. Astola. Blind evaluation of additive noise variance in textured images by nonlinear processing of block dct coefficients. In *Image Processing: Algorithms and Systems II*, volume 5014 of SPIE Proceedings, pages 178–189, 2003.

[150] N. N. Ponomarenko, V. V. Lukin, M. S. Zriakhov, A. Kaarna, and J. T. Astola. An automatic approach to lossy compression of aviris images. *IEEE International Geoscience and Remote Sensing Symposium*, 2007.

[151] B. Prescott and G. F. Mclean. Line-based correction of radial lens distortion. *Graphical Models and Image Processing*, 59:39–47, 1997.

[152] M. Protter, M. Elad, H. Takeda, and P. Milanfar. Generalizing the non-local-means to super-resolution reconstruction. *IEEE Transactions on Image Processing*, 18(1):36–51, 2009.

[153] K. Rank, M. Lendl, and R. Unbehauen. Estimation of image noise variance. In *IEEE Proceedings- Vision, Image and Signal Processing*, volume 146, pages 80–84, 1999.

[154] N. Sabater. *Reliability and Accuracy in Stereovision. Application to Aerial and Satellite High Resolution Images.* PhD thesis, École Normale Supérieur de Cachan, 2009.

[155] C. Schmid, R. Mohr, and C. Bauckhage. Evaluation of interest point detectors. *International Journal of Computer Vision*, 37(2):151–172, 2000.

[156] Yeol-Min Seong and HyunWook Park. Superresolution technique for planar objects based on an isoplane transformation. *Optical Engineering*, 47, 2008.

[157] E. Shechtman, Yaron Caspi, and Michal Irani. Increasing space-time resolution in video. In *European Conference on Computer Vision*, pages 753–768, 2002.

[158] Eli Shechtman, Yaron Caspi, and Michal Irani. Space-time super-resolution. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(4):531–545, 2005.

[159] C. C Slama. *Manual of Photogrammetry, 4th edition.* Falls Church, American Society of Photogrammetry, Virginia, 1980.

[160] R.M. Steele and C. Jaynes. Feature uncertainty arising from covariant image noise. *Conference on Computer Vision and Pattern Recognition*, 1, 2005.

[161] Gideon P. Stein. Lens distortion calibration using point correspondences. *Conference on Computer Vision and Pattern Recognition*, 602-608, 1997.

[162] Daniel Stevenson and Margaret M. Fleck. Nonparametric correction of distortion. *IEEE Workshop on Applications of Computer Vision*, 1995.

[163] Peter Sturm and Steve Maybank. On plane-based camera calibration: A general algorithm, singularities, applications. *Conference on Computer Vision and Pattern Recognition*, pages 432–437, 1999.

[164] F. Sur, F. Cao, P. Musé, and Y. Gousseau. Unsupervised thresholds for shape matchings. *International Conference on Image Precessing*, 2, 2003.

[165] H. Takeda, P. Milanfar, M. Protter, and M. Elad. Super-resolution without explicit subpixel motion estimation. *IEEE Transactions on Image Processing*, 18(9):1958–1975, 2009.

[166] Chi-Keung Tang, Gerard G. Medioni, and Mi-Suen Lee. N-dimensional tensor voting and application to epipolar geometry estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(8):829–844, 2001.

[167] Brodatz Textures. www.ux.uis.no/ tranden/brodatz.html.

[168] S. Thirthala and M. Pollefeys. Multi-view geometry of 1d radial cameras and its application to omnidirectional camera calibration. *International Conference on Computer Vision*, 2:1539–1546, 2005.

[169] Klaus E. Timmermann and Robert D. Nowak. Multiscale modeling and estimation of poisson processes with application to photon-limited imaging. *IEEE Transactions on Information Theory*, 45(3):846–862, 1999.

[170] P. Torr and A. Zisserman. Mlesac: A new robust estimator with application to estimating image geometry. *Computer Vision and Image Understanding*, 78:138–156, 2000.

[171] B. Triggs, P. McLauchlan, R. Hartley, and A. Fitzgibbon. Bundle adjustment-a modern synthesis. *The International Workshop on Vision Algorithms*, page 298–372, 1999.

[172] Roger Y. Tsai. A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses. *IEEE Journal of Robotics and Automation*, Vol. RA-3, 1987.

[173] Patrick Vandewalle, Sabine Süsstrunk, and Martin Vetterli. A frequency domain approach to registration of aliased images with application to super-resolution. *EURASIP Journal on Applied Signal Processing*, 2006:1–14, March 2006.

[174] R. Grompone von Gioi, J. Jakubowicz, J.-M. Morel, and G. Randall. Lsd: A fast line segment detector with a false detection control. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 99, 2008.

[175] R. Grompone von Gioi, P. Monasse, J.-M. Morel, and Z. Tang. Self-consistency and universality of camera lens distortion models. *CMLA Preprint, ENS-Cachan*, 2010.

[176] Guo-Qing Wei and Song De Ma. Implicit and explicit camera calibration: theory and experiments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16:469–480, 1994.

[177] J. Weng, P. Cohen, and M. Herniou. Camera calibration with distortion models and accuracy evaluation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14:965–980, 1992.

[178] Paul R. Wolf. *Elements of photogrammetry.* McGraw-Hill Companies, 1983.

[179] K.W. Wong. Mathematical formulation and digital analysis in close-range photogrammetry. *Photogrammetric Engieering Remote Sensing*, 41(11):1355–1373, 1975.

[180] Jianchao Yang, J. Wright, T. Huang, and Yi Ma. Image super-resolution as sparse representation of raw image patches. In *Conference on Computer Vision and Pattern Recognition*, pages 1–8, June 2008.

[181] Zhan-Long Yang and Bao-Long Guo. Image mosaic based on sift. *International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, pages 1422–1425, 2008.

[182] G. Yu and S. Mallat. Sparse super-resolution with space matching pursuit. In *Proc. of Signal Processing with Adaptive Sparse Structured Representation (SPARS)*, 2009.

[183] G. Yu and S. Mallat. Super-resolution with sparse mixing estimators. Technical report, CMAP, Ecole Polytechnique, 2009.

[184] G. Yu, G. Sapiro, and S. Mallat. Solving inverse problems with piecewise linear estimators: From gaussian mixture models to structured sparsity. *Arxiv preprint arXiv:1006.3056*, 2010.

[185] Lu Yuan, Jian Sun, Long Quan, and Heung-Yeung SHum. Image deblurring with blurred/noisy image pairs. In *SIGGRAPH*, 2007.

[186] Z. Yuan, P. Yan, and S. Li. Super resolution based on scale invariant feature transform. In *International Conference on Audio, Language and Image Processing*, pages 1550–1554, 2008.

[187] B. Zeisl, P.F. Georgel, F. Schweiger, E. Steinbach, and N. Navab. Estimation of location uncertainty for scale invariant feature points. *British Machine Vision Conference*, 2009.

[188] B. Zhang, M. Fadili, and J. L. Starck. Wavelet, ridgelets and curvelets for poisson noise removal. *IEEE Transactions on Image Processing*, 17(7):1093–1108, 2008.

[189] Li Zhang, Sundeep Vaddadi, Hailin Jin, and Shree Nayar. Multiple view image denoising. In *Conference on Computer Vision and Pattern Recognition*, 2009.

[190] Z. Zhang. On the epipolar geometry between two images with lens distortion. *International Conference on Pattern Recognition*, 1996.

[191] Z. Zhang. A flexible new technique for camera calibration. *International Conference on Computer Vision*, pages 663–673, September 1999.

[192] Z. Zhang, R. Deriche, O. D. Faugeras, and Q.T. Luong. A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry. *Artificial Intelligence*, 78(1-2):87–119, 1995.

[193] Gangqiang Zhao, Ling Chen, Jie Song, and Gencai Chen. Large head movement tracking using sift-based registration. In *International conference on Multimedia*, pages 807–810, New York, NY, USA, 2007. ACM.

[194] W. Zhao and Harpreet S. Sawhney. Is super-resolution with optical flow feasible? In *European Conference on Computer Vision*, pages 599–613, 2002.