3D Vision Geometry for Rolling Shutter Cameras

Yizhen Lao

DOCTORAL THESIS

3D Vision Geometry for Rolling Shutter Cameras

Author:
Yizhen LAO

Thesis supervisors:
Helder ARAUJO
Omar AIT-AIDER

Defended on
16 MAY 2019

Jury members:

Rapporteur	Nicolas ANDREFF, Professor, Université Franche Comté
Rapporteur	Tomas PAJDLA, Associate Professor and Distinguished Researcher, Czech Technical University in Prague
Examinateur	Sylvie CHAMBON, MCF, ENSEEIHT Toulouse
Examinateur	Adrien BARTOLI, Professor, Université Clermont Auvergne

A thesis submitted in fulfillment of the requirements for the degree of Docteur de l'Université Clermont Auvergne

at

(ISPR group, ComSee team)
Institut Pascal

UNIVERSITÉ CLERMONT AUVERGNE
Ecole Doctorale des Sciences Pour l'Ingénieur
Institut Pascal

Abstract

**3D Vision Geometry
for Rolling Shutter Cameras**

by Yizhen LAO

Many modern CMOS cameras are equipped with Rolling Shutter (RS) sensors which are considered as low cost, low consumption and fast cameras. In this acquisition mode, the pixel rows are exposed sequentially from the top to the bottom of the image. Therefore, images captured by moving RS cameras produce distortions (e.g. wobble and skew) which make the classic algorithms at best less precise, at worst unusable due to singularities or degeneracies. The goal of this thesis is to propose a general framework for modelling and solving structure from motion (SfM) with RS cameras. Our approach consists in addressing each sub-task of the SfM pipe-line (namely image correction, absolute and relative pose estimation and bundle adjustment) and proposing improvements.

The first part of this manuscript presents a novel RS correction method which uses line features. Unlike existing methods, which uses iterative solutions and make Manhattan World (MW) assumption, our method R4C computes linearly the camera instantaneous-motion using few image features. Besides, the method was integrated into a RANSAC-like framework which enables us to detect curves that correspond to actual 3D straight lines and reject outlier curves making image correction more robust and fully automated.

The second part revisits Bundle Adjustment (BA) for RS images. It deals with a limitation of existing RS bundle adjustment methods in case of close read-out directions among RS views which is a common configuration in many real-life applications. In contrast, we propose a novel camera-based RS projection algorithm and incorporate it into RSBA to calculate reprojection errors. We found out that this new algorithm makes SfM survive the degenerate configuration mentioned above.

The third part proposes a new RS Homography matrix based on point correspondences from an RS pair. Linear solvers for the computation of this matrix are also presented. Specifically, a practical solver with 13 point correspondences is proposed. In addition, we present two essential applications in computer vision that use RS homography: plane-based RS relative pose estimation and RS image stitching.

The last part of this thesis studies absolute camera pose problem (PnP) and SfM which handle RS effects by drawing analogies with non-rigid vision, namely Shape-from-Template (SfT) and Non-rigid SfM (NRSfM) respectively. Unlike all existing methods which perform 3D-2D registration after augmenting the Global Shutter (GS) projection model with the velocity parameters under various kinematic models, we propose to use local differential constraints. The proposed methods outperform stat-of-the-art and handles configurations that are critical for existing methods.

KEYWORDS: Rolling shutter; Image correction; Pose estimation; Relative pose estimation; Homography; Structure from Motion; Bundle Adjustment.

Résumé

De nombreuses caméras CMOS modernes sont équipées de capteurs Rolling Shutter (RS). Ces caméras à bas coût et basse consommation permettent d'atteindre de très hautes fréquences d'acquisition. Dans ce mode d'acquisition, les lignes de pixels sont exposées séquentiellement du haut vers le bas de l'image. Par conséquent, les images capturées alors que la caméra et/ou la scène est en mouvement présentent des distorsions qui rendent les algorithmes classiques au mieux moins précis, au pire inutilisables en raison de singularités ou de configurations dégénérées. Le but de cette thèse est de revisiter la géométrie de la vision 3D avec des caméras RS en proposant des solutions pour chaque sous-tâche du pipe-line de Structure-from-Motion (SfM).

Le chapitre II présente une nouvelle méthode de correction du RS en utilisant les droites. Contrairement aux méthodes existantes, qui sont itératives et font l'hypothèse dite Manhattan World (MW), notre solution est linéaire et n'impose aucune contrainte sur l'orientation des droites 3D. De plus, la méthode est intégrée dans un processus de type RANSAC permettant de distinguer les courbes qui sont des projections de segments droits de celles qui correspondent à de vraies courbes 3D. La méthode de correction est ainsi plus robuste et entièrement automatisée.

Le chapitre III revient sur l'ajustement faisceaux ou bundle adjustment (BA). Nous proposons un nouvel algorithme basé sur une erreur de projection dans laquelle l'index de ligne des points projetés varie pendant l'optimisation afin de garder une cohérence géométrique contrairement aux méthodes existantes qui considère un index fixe (celui mesurés dans l'image). Nous montrons que cela permet de lever la dégénérescence dans le cas où les directions de scan des images sont trop proches (cas très communs avec des caméras embraquées sur un véhicule par exemple).

Dans le chapitre VI nous étendons le concept d'homographie aux cas d'images RS en démontrant que la relation point-à-point entre deux images d'un nuage de points coplanaires pouvait s'exprimer sous la forme de 3 à 7 matrices de taille 3X3 en fonction du modèle de mouvement utilisé. Nous proposons une méthode linéaire pour le calcul de ces matrices. Ces dernières sont ensuite utilisées pour résoudre deux problèmes classiques en vision par ordinateur à savoir le calcul du mouvement relatif et le « mosaïcing » dans le cas RS.

Dans le chapitre V nous traitons le problème de calcul de pose et de reconstruction multi-vues en établissant une analogie avec les méthodes utilisées pour les surfaces déformables telles que SfT (Structure-from-Template) et NRSfM (Non Rigid Structure-from-Motion). Nous montrons qu'une image RS d'une scène rigide en mouvement peut être interprétée comme une image Global Shutter (GS) d'une surface virtuellement déformée (par l'effet RS). La solution proposée pour estimer la pose et la structure 3D de la scène est ainsi composée de deux étapes. D'abord les déformations virtuelles sont d'abord calculées grâce à SfT ou NRSfM en assumant un modèle GS classique (relaxation du modèle RS). Ensuite, ces déformations sont réinterprétées comme étant le résultat du mouvement durant l'acquisition (réintroduction du modèle RS). L'approche proposée présente ainsi de meilleures propriétés de convergence que les approches existantes.

Mot clés Rolling shutter, Pose absolue et relative, Homographie, S-f-M, Ajustement de faisceaux.

Dedicated to my family

This work has been sponsored by the French government research program "Investissements d'Avenir" through the IDEX-I-SITE initiative 16-IDEX-0001 (CAP 20-25), the IMobS3 Laboratory of Excellence (ANR-10-LABX-16-01). This research was also financed by the European Union through the Regional Competitiveness and Employment program -2014-2020- (ERDF – AURA region) and by the AURA region.

Acknowledgements

The past three years at Clermont-FD have been an unforgettable and invaluable experience to me. I would not have been able to make this journey without the help and support of many, many people and I feel deeply indebted to them.

First of all, I want to thank my supervisor Helder Araujo for his guidance during my study. I also want thank Adrien Bartoli for being on my thesis jury and also for a lot of advices and valuable help throughout our collaboration. But specially, I want to express my deepest appreciation to my advisor Omar Ait-Aider. Omar is an extremely kind, caring and supportive advisor that I could not have asked for more. He continually instilled the requirements of being a good scientist into my mind, which will have a lifelong influence on me. And I have already started to miss him.

I would also like to thank all the jury members, Tomas Pajdla, Nicolas Andreff and Sylvie Chambon for reading my thesis. Their insightful remarks are absolutely helpful for this work.

Secondly, I want to thank my fellow PhD students at the ComSee group for many enlightening discussions and the time we working together. Besides, my thanks also go to my friends at Clermont-Ferrand, Chao Zhang, Jinpeng Wang, Yongzhe Yan, Miao Wang with all of whom I shared hours of discussion, study, work, play, food, and travel. In addition, I am thankful to some friends outside, Fang Li, Mingye Bao and Hanjun Ling for making my spare time enriched and memorable.

I thank my parents: Liling Jiang and Ningjun Lao. Like most Chinese students in my generation, I am the only child of my family and I have a very close relationship with them. They give me their very long emotional and financial support, true love, wisdom. In short, my parents made me who I am today and I never know how to pay them back. I hope that they are at least a little proud of me for what I have been through so far.

In the end, I want to thank my wife Yumeng for her love and support. Thank you for making our married life all that I dreamed it would be. Thank you for filling our home with love and happiness. Thank you for giving me the most beautiful children Da-He. I love you with all my heart.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Context

The main objective of 3D computer vision is to recover 3D information from a collection of 2-D images. This task is widely termed as 3D reconstruction. Although active image sensors which can provide the scene depth directly such as laser sensor cameras, time-of-flight and structured light RGB-D cameras are now available, 3D reconstruction from classical 2D images basing on multiple-view geometry remains an intensively studied topic for industrial and academic communities since it is considered as a low-cost and flexible solution. It had been applied to building reconstruction, hyperscale landform model generation, street view mobile mapping and even VR/AR gaming etc. When 3D reconstruction is needed, multiple view geometry based methods using consumer cameras (e.g. cellphone cameras) is much more cost and time efficient than classical methods such as aerial photogrammetry or laser scanning. However, these low-cost filming devices could easily suffer from some low-level imaging problems [Yang et al., 2018], namely photometric calibration, image blur and rolling shutter (RS) effect, which can greatly affect the quality of 3D reconstruction from 2D images. In this thesis, we address the problem of 3D reconstruction from multiple images under RS effect.

1.1.1 Global Shutter VS. Rolling Shutter:

The camera sensor consists of a 2D array of photon-sensitive elements which record the intensity of light that falls on them. The global shutter (GS) mechanism employed in CCD sensors exposes all sensor elements to light at the same time. Thus, the 2D image is a projection of a 3D scene at a particular time, similar to the projection of light on the human retina. On the other hand, the rolling shutter (RS) mechanism in a camera sensor captures scenes in a different way compared to that of humans and global shutter. This mechanism is employed primarily in CMOS sensors as against CCD sensors as discussed by [Litwiller, 2001]. The rolling shutter sensor is designed in such a way that each of its rows starts its exposure sequentially one after the other. Through this mechanism, each of the rows ends its exposure at different times from each other, and hence, a single read-out circuit is shared by all rows. Thus, a rolling shutter camera embeds a temporal interplay between its exposure and sensor rows.

The shared read-out mechanism employed in rolling shutter sensors leveraging the sequential exposure reduces the circuitry used, resulting in a compact camera with reduced power consumption. For these reasons, RS cameras are considered as low-cost, smaller, faster cameras that use less power. They are the preferred choice for embedded consumer systems.

As shown in Fig. 1.1, almost all consumer cameras in the current market employ the RS mechanism, and hence, it is important to study its effects now.

FIGURE 1.1: CMOS cameras where rolling shutter commonly used are now wining the battle of current camera market against to CCD cameras.

1.1.2 3D Vision with RS Cameras

A static RS camera filming a rigid scene can be regarded as a GS camera. However, when either the scene or the camera moves during the acquisition period, the row delay introduces distortions and image artefacts called RS effect. Even with a shorter exposure, each row of sensors experiences different camera motion during its own exposure interval producing geometric distortions. Fig. 1.2(a) shows an image of a rotating propeller captured by a static rolling shutter camera. Fig. 1.2(b), shows a scene captured from a moving rolling shutter camera. Obviously, RS effect can not be corrected in a straightforward manner since it depends on both the scene geometry and the camera motion which are precisely the desired parameters in 3D vision applications.

Nowadays the multi-view geometry with GS is well understood and mature and stable solutions have been proposed for all 3D vision problems. Many solutions have been commercialized such as [Wu, 2011]. However, almost all these classical 3D reconstruction methods were developped for GS camera models. The use of these methods with RS cameras and dynamic scenes produces at best, degraded results, at worst errors due to ill-defined problems or singularities [Hedborg et al., 2012].

Although RS, as a space-time sweeping camera, is related to non-central camera models such as push-broom (PB) cameras [Hartley and Zisserman, 2003], oblique cameras [Pajdla, 2002] and The Crossed-Slits (X-Slits) cameras [Feldman et al., 2003]. However, RS camera model had never been specifically addressed (even not been mentioned in the famous textbook in 3D vision [Hartley and Zisserman, 2003]) before it was defined geometrically for the first time in 2005 [Meingast et al., 2005].

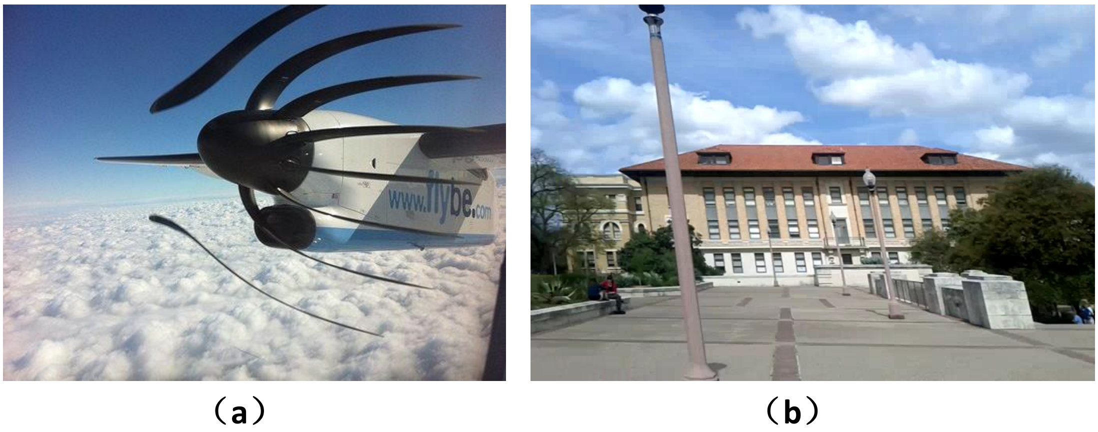


FIGURE 1.2: Images captured with a rolling shutter camera. (a) Rotating propeller captured with a static rolling shutter camera. (b) Moving rolling shutter camera, capturing a static scene. Due to the camera motion, distortion presents in the image.

1.2 Motivation and Challenges

1.2.1 Applications

Developing 3D vision solutions for RS cameras is of a very broad interest in many applications where the images are taken with moving RS cameras such as hand-held mobile phone photography and robot platform. Good solutions have many applications in a range of domains including (some examples are shown in Fig. 1.3):

- Entertainment

  - Augmented Reality (AR)/Virtual Reality (VR) [Bapat and Frahm, 2016, Bapat and Frahm, 2018]

  - Mobile phone image correction and denoising [Forssén and Ringaby, 2010, Rengarajan et al., 2017]

- Object pose and kinematic estimation [Ait-Aider et al., 2006, Magerand et al., 2012]

- 3D scene reconstruction

  - Mobile vehicle platform [Klingner et al., 2013, Sau, 2013]

  - UAV platform [Vautherin et al., 2016]

- Robot navigation

  - Absolute pose estimation [Albl et al., 2015, Saurer et al., 2015]

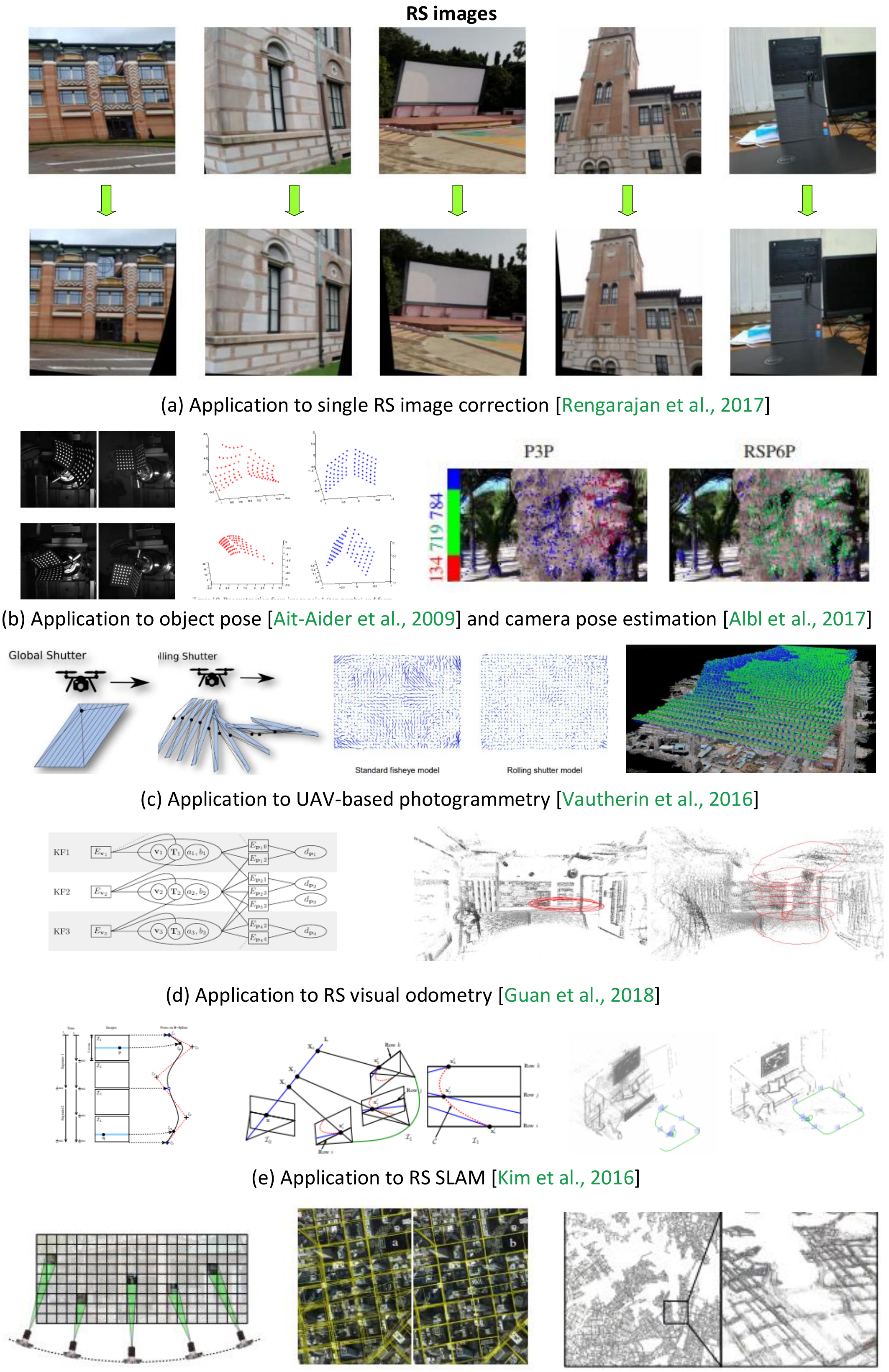
  - Visual odometry (VO) [Guan et al., 2018]

  - Visual SLAM [Kim et al., 2016]

  - RS compensation for self-driving car [Purkait and Zach, 2017]

1.2.2 Current Approaches and Their Limitations

As mentioned earlier, the 3D reconstruction by using SfM with GS cameras cannot be directly extended to RS case. As these RS images may suffer from deformations, the inter-image visual motion is now dependent on both camera pose and instantaneous-motion during the acquisition period.



(a) Application to single RS image correction [Rengarajan et al., 2017]

(b) Application to object pose [Ait-Aider et al., 2009] and camera pose estimation [Albl et al., 2017]

(c) Application to UAV-based photogrammetry [Vautherin et al., 2016]

(d) Application to RS visual odometry [Guan et al., 2018]

(e) Application to RS SLAM [Kim et al., 2016]

(f) Application to large-scale RS SfM [Klingner et al., 2013]

FIGURE 1.3: Applications of 3D vision with RS cameras.

Existing work try to solve 3D reconstruction with RS cameras problem from different view angles:

- **By correcting RS effect:** [Purkait et al., 2017, Purkait and Zach, 2017] attempted to compensate the RS effect in each image before performing classical GS SfM based on the corrected images.
- **By using additional sensors:** [Klingner et al., 2013, Saurer et al., 2016] try to solve the SfM with RS cameras by using GPS and IMU, which offer more information and constraints on camera motion.
- **By using video sequence:** [Hedborg et al., 2011, Zhuang et al., 2017, Im et al., 2018] propose to use RS videos, which provide ordered and successive frames with short base-lines, to recover the 3D scene.
- **By using RS absolute pose estimation:** Absolute pose estimation can be used to add a new frame and expand the reconstruction incrementally in SfM pipeline. Therefore, an effective RS absolute pose estimation method can serve for RS SfM (e.g. [Albl et al., 2016b] uses RS pose estimation method [Albl et al., 2015] to add the views.)

The extension of GS multiple view geometry with GS to the RS case is not straightforward. Most of existing solutions are with strong assumptions that limit the field of application. These hypotheses are made either on the geometry of the scene (planars scenes, Manhattan world, presence of vanishing directions) or on the kinematic model describing the movement during the acquisition (pure translation, pure rotation, smooth motion) or on the camera poses (short baselines, varying scanning direction). With the brief discussion above (details in section. 2.4), we want to emphasize the following limitations of existing SfM with RS cameras methods:

1. Strong assumptions that do not always hold in real applications such as pure rotation [Ito and Okatani,], pure translation [Saurer et al., 2016] or smooth motion [Hedborg et al., 2011].
2. Using additional sensors which makes the implementation not straightforward [Hee Park and Levoy, 2014, Jia and Evans, 2012, Patron-Perez et al., 2015].
3. Video-based methods [Liang et al., 2008, Forssén and Ringaby, 2010, Kim et al., 2011, Grundmann et al., 2012, Zhuang et al., 2017] commonly impose a high acquisition framerate which results in high computational efficiency requirements. Unordered images with large baseline are not handled.
4. SfM with RS cameras may easily fail into the degeneracies pointed out in [Albl et al., 2016b]. The solutions proposed in [Albl et al., 2016b, Ito and Okatani,] impose a filming style (different readout directions) that can not be achieved in practical applications.

1.2.3 Objectives

In summary, although recently, many methods have been designed to fit RS camera applications, 3D vision with RS currently lacks strong theoretical understanding. 3D reconstruction with RS cameras undergoing general motion and observing general scenes remains an open and challenging problem. A new robust and stable solution to solve RS SfM with unordered images and without overly restrictive assumptions on camera motion, readout direction or projection model is still absent from the literature. Thus, the goal of this thesis is to propose a general framework for modelling and solving 3D reconstruction with RS cameras.

Our approach consists in addressing each sub-task of the SfM pipe-line (namely feature selection and matching, monocular pose and relative pose) and proposing improvements. It is now clear that the general RS SfM is not sufficiently constrained, and thus can not be solved efficiently without any prior. Nevertheless, all the proposed methods are based on constraints that are feasible and that are usually used in classical computer vision applications. Beside the theoretical contribution, this work aims to be a step in the potential widespread deployment of 3D vision with RS imaging systems.

1.3 Contribution

As shown in Fig. 1.4, this thesis has six main contributions:

1. ***A Robust Method for Strong Rolling Shutter Effects Correction Using Lines with Automatic Feature Selection.*** We present a novel RS correction method which uses line features. Unlike existing methods, which uses iterative solutions and make Manhattan World (MW) assumption, our method R4C computes linearly the camera instantaneous-motion using few image features. Besides, the method was integrated in a RANSAC-like framework which enables us to detect curves that correspond to actual 3D straight lines and reject outlier curves making image correction more robust and fully automated.

2. ***RS Bundle Adjustment (RSBA) Revisited.*** [Albl et al., 2016b] investigated mechanism of planar degeneracy which often raised during RSBA using measurements-based projection. In contrast, we propose a novel camera-based RS projection algorithm, and incorporate it into RSBA to calculate reprojection errors. We found out that this camera-based RS BA (C-RSBA) makes SfM survive degenerate solutions that occur with common and natural capture style compared to existing works. Thus, without constrains on camera motions e.g. perpendicular read-out directions among RS views [Albl et al., 2016b], C-RSBA can also successfully achieve accurate structure and motion computation.

3. ***Robustified SfM with Rolling-Shutter Camera Using Straightness Constraint.*** We propose a 3-steps approach for RSSfM problem using the proposed single RS image correction method and the camera-based RSBA. We show that the combination of these two methods enables us to achieve high accuracy 3D reconstruction basing on large dataset experiments with both synthetic and real scenes.

4. ***Rolling Shutter Homography and its Applications.*** We investigate the computation of the homography matrix based on correspondences from an RS pair. We show that at least 36 correspondences are needed in theory to compute the homography matrix linearly, and then we derive a practical method which works with 13 correspondences. In addition, we present two essential applications in computer vision that use RS homography: *1)* Plane-based RS relative pose estimation and *2)* RS image stitching.

5. ***Rolling Shutter Pose and Instantaneous-motion Estimation using Shape-from-Template.*** The main idea consists in considering that RS distortions due to camera instantaneous-motion during image acquisition can be interpreted as virtual deformations of a template captured by a GS camera. First, the virtual deformations are recovered

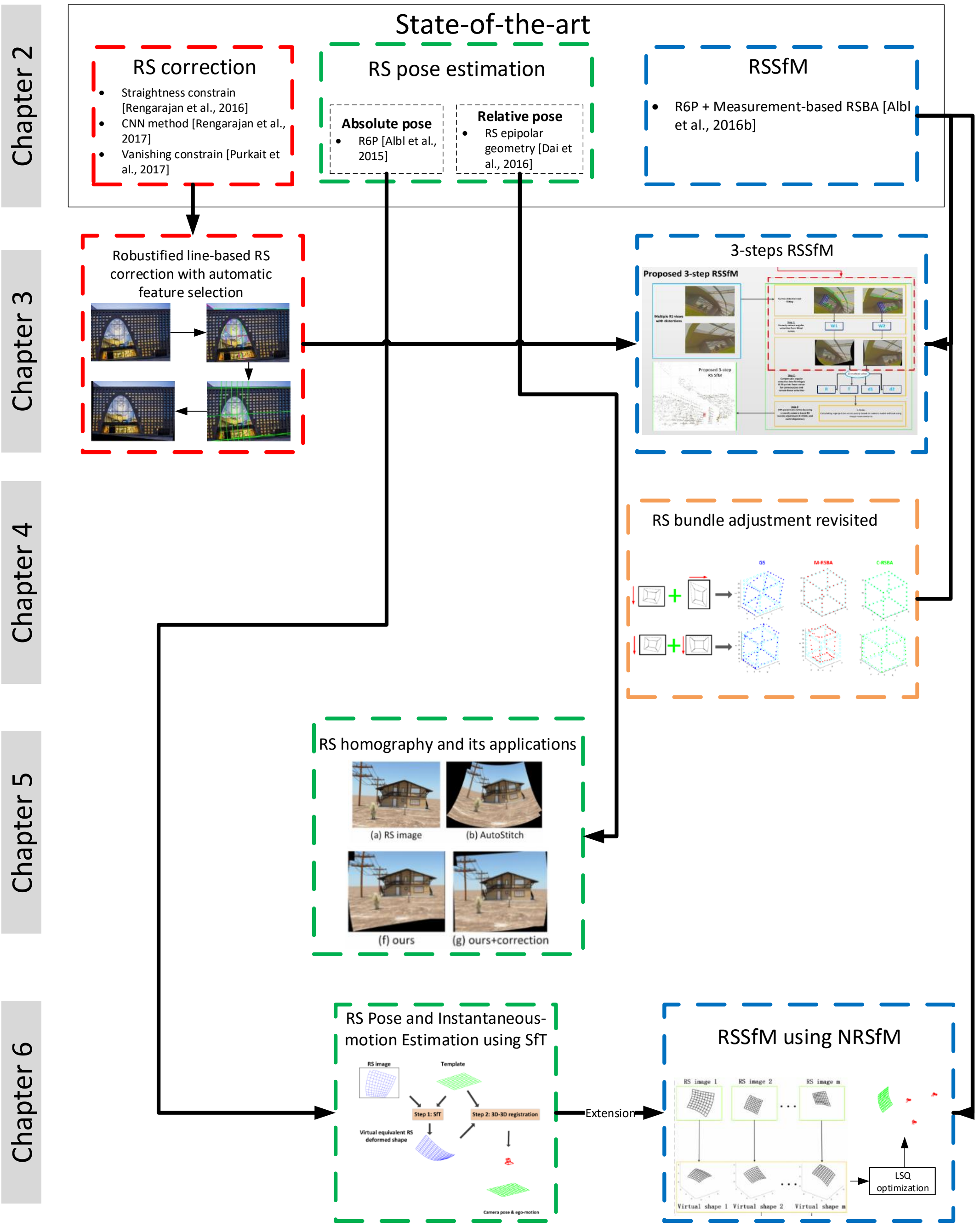

Chapter 2
Chapter 3
Chapter 4
Chapter 5
Chapter 6
State-of-the-art
RS correction
Straightness constrain [Rengarajan et al., 2016]
CNN method [Rengarajan et al., 2017]
Vanishing constrain [Purkait et al., 2017]
RS pose estimation
Absolute pose
R6P [Albl et al., 2015]
Relative pose
RS epipolar geometry [Dai et al., 2016]
RSSfM
R6P + Measurement-based RSBA [Albl et al., 2016b]
Robustified line-based RS correction with automatic feature selection
3-steps RSSfM
Proposed 3-step RSSfM
RS bundle adjustment revisited
RS homography and its applications
(a) RS image
(b) AutoStitch
(f) ours
(g) ours+correction
RS Pose and Instantaneous-motion Estimation using SfT
RS image
Template
Step 1: SfT
Step 2: 3D-3D registration
Virtual equivalent RS deformed shape
Camera pose & ego-motion
Extension
RSSfM using NRSfM
RS image 1
RS image 2
RS image n
Virtual shape 1
Virtual shape 2
Virtual shape n
LSQ optimization

FIGURE 1.4: Overview of the main contributions of this thesis. We discuss the background theory, mathematical preliminaries and state-of-the-art in chapter. 2. Chapter. 3 presents our contributions to RS correction and the 3-steps RSSfM method. Chapter. 6 give our contributions to RS pose estimation and RSSfM using NRSfM. While our contributions to RS homography and bundle adjustment are shown in Chapter. 5 and Chapter. 4 respectively. Chapter. 7 presents our conclusions and perspectives for future work.

basing on local differential constraints thanks to Shape-from-Template (SfT) technique. Then, the camera pose and instantaneous-motion are computed by registering the deformed scene on the original template. This 3D-3D registration involves a 3D cost function based on the Euclidean point distance, more physically meaningful than the re-projection error or the algebraic distance based cost functions used in previous work.

6. ***Solving Rolling Shutter SfM Using Non-Rigid SfM.*** We propose a solution to the SfM problem for RS images (RSSfM) using an analogy with Non-Rigid SfM (NRSfM). We first show that, to a certain extent, images of a rigid surface acquired by a moving RS camera can be interpreted as images of a virtually deformed surface taken by a GS camera. We then propose the following two-step method for at least three RS images of an unknown scene. The first step reconstructs virtual deformation by means of NRSfM by relaxing the RS constraint. The second step retrieves the actual structure, camera pose and instantaneous-motion by reintroducing the RS constraint. The proposed method handles most of the common degenerate configurations of RSSfM and outperforms existing methods in accuracy and stability.

1.4 Thesis Organization

We have divided this thesis into 7 chapters:

- ***Chapter.* 2:** Background theory and mathematical preliminaries.
- ***Chapter.* 3:** Contributions to RS correction and the 3-steps RSSfM method.
- ***Chapter.* 4:** Contributions to RS bundle adjustment.
- ***Chapter.* 5:** Contributions to RS homography.
- ***Chapter.* 6:** Contributions to RS pose estimation and RSSfM using NRSfM
- ***Chapter.* 7:** Conclusions and perspectives for future work.

Chapter 2

Background and Previous Work

In this chapter, we first introduce the classical Global Shutter (GS) model and three famous 3D computer vision problems which are addressed in this thesis. We will then discuss Rolling Shutter (RS) model and give a brief overview of exiting methods that addressed RS 3D vision problems.

2.1 Image Formation

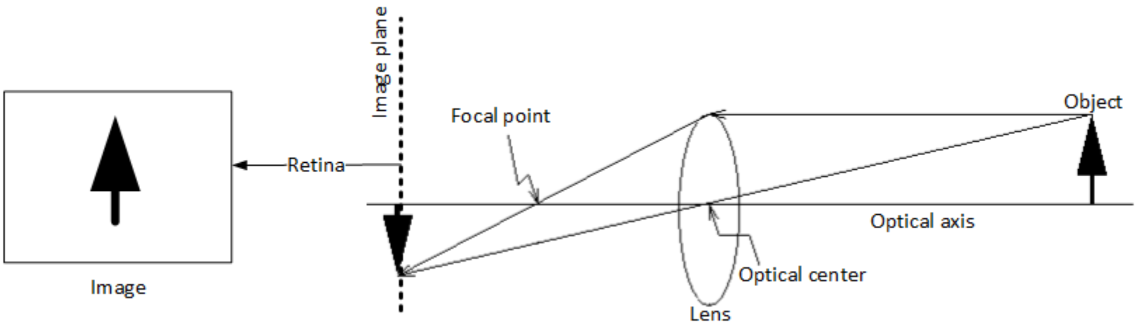


FIGURE 2.1: Camera imaging model.

A digital camera captures light rays from a 3D scene and provides a digital image under the form of a pixel array. It is constituted of two parts: an optical lens that focuses the light rays and a photosensitive retina which converts the light into electrical charges, then into digital values that represent RGB or gray level data (Fig. 2.1).

2.1.1 A Little Electronics

2.1.1.1 The Retina

The retina, is the part of the camera that converts the light (namely the number of photons that cross lens and fall in the same point) into an electrical charge which is digitized and stored.

Historically, image data was created by a chemical reaction on a thin film behind the lens. The time during which the sensor is exposed to light determines the amount of light that is stored, and therefore the value of the pixels. A mechanical shutter was then used to allow exposure for only a definite time.

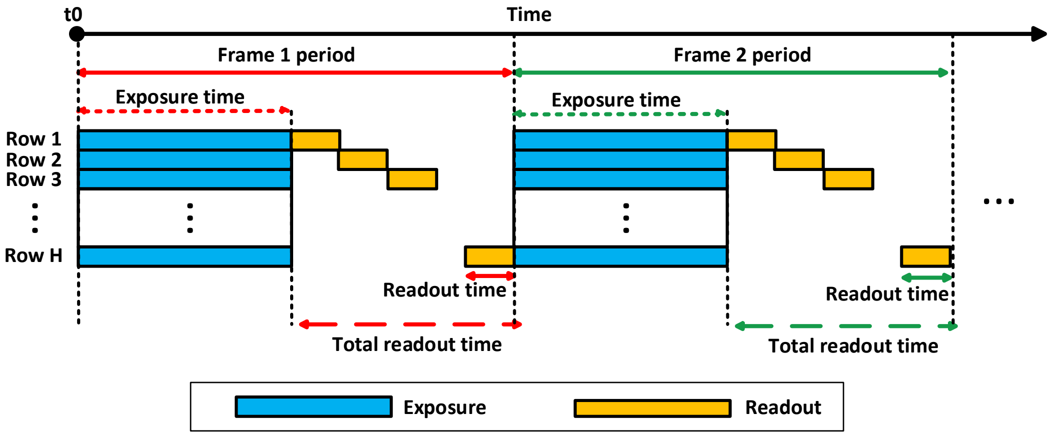


FIGURE 2.2: Exposure mechanism of GS cameras.

2.1.1.2 CCD vs CMOS

In digital cameras, the film has been replaced by a panel of sensors consisting of electronic components.

CCD (charge coupled device) and CMOS (complementary metal oxide semiconductor) image sensors are two different technologies for capturing digital images. Each has unique strengths and weaknesses giving advantages in different applications. In a CCD technology, charge is transferred using few output nodes. All of the retina surface serves to light capture providing high image uniformity. The advantage of CCDs are lower readout noise, no fixed pattern noise, and low on-chip power dissipation. But on the other side, they require high charge transfer efficiency, complex and large controlling units which increases the device size and are high power consuming.

In a CMOS sensor, each pixel has its own conversion circuit so that the chip outputs digital bits. The design complexity reduces the area devoted to light capture and image uniformity is lower. Inversely, image data transfer is massively parallel, allowing high speed imaging and on-chip image processing.

After CCD domination in the 1970s, due to superior images, there was a renewed interest on CMOS based on lowered power consumption, camera-on-a-chip integration, lowered fabrication costs and high image quality since the 1990s.

2.1.1.3 Global Shutter vs Rolling Shutter

Exposure time is no longer controlled by a mechanical obturator but rather by an electronic shutter. This led to two shutter modes.

Global Shutter: With a GS, all existing information is first removed from pixels. The pixels are then electronically allowed together to receive the light (exposure phase). At the end of the exposure time, the charges are recorded simultaneously into an area which is insensitive to light. This information is then converted in gray levels by CMOS sensors and transferred. The current CMOS sensors have become so fast that pixel information is transferred simultaneously in series on up to 24 cables, which is an extreme challenge for the following circuits (FPGA, ASIC, and USB or Ethernet chips)[1].

[1] https://fr.ids-imaging.com//technical-article//fr_tech-article-rolling-shutter-sensors.html

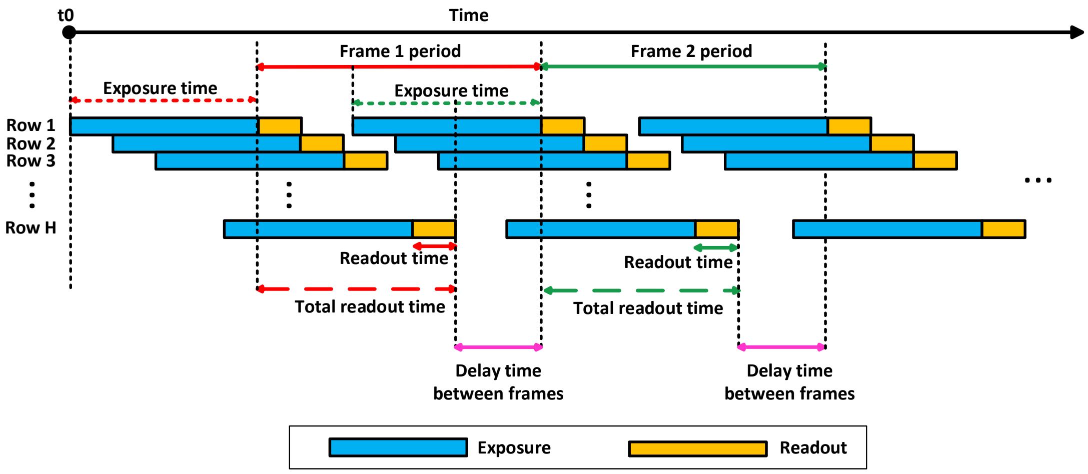


FIGURE 2.3: Exposure mechanism of RS cameras.

As shown in Fig. 2.2, all pixels are simultaneously exposed to light in a GS camera. This exposure mechanism is installed in the majority of CCD sensors. The ability of simultaneously controlling all pixels allows to capture a true snapshot of the scene regardless of the camera instantaneous-motion.

Rolling Shutter: The current trend is to get more and more pixels on a constantly smaller surface. This is an extreme challenge and requires a pixel to house many components. To achieve even smaller pixels (for example for smart-phones with mega-pixel resolution), the pixel must fit in the 1 µm range. This forces you to drop components, such as the buffer in the pixel. A global shooting at a specific moment is no longer possible with a simple chip. The solution is to determine the end of exposure by reading the information directly. Since the lines are transmitted one after the other, this is a rolling recording, hence the rolling shutter designation[2]. As shown in Fig. 2.3, the RS slit slides over the sensor exposing each scanline sequentially (commonly from the top to the bottom of the image).

Global Shutter vs Rolling Shutter: Images taken with a GS camera are free from any artifacts generated by the motion, because they are snapshots. A static RS camera filming a static scene equals to a GS one since all the scanlines exposure are with the same pose. However, images captured by moving RS cameras produce distortions (e.g. wobble, skew).

By dropping the buffer image, quality is improved. In the GS case, at the end of the exposure time, the value relative to the brightness is "saved by elimination" in a memory cell. Because the last line expects the total duration of a frame for the final extraction this information can be altered by temperature variation over time. The black level and the digital noise are increased. On the other hand, a RS sensor directly converts the brightness, without this intermediate step.

A GS sensor can produce annoying ghosting images when shooting in bright sunlight with a lot of light and a very short exposure time. Buffer information is also not exposed directly to light after shooting. However ghost images appear due to the displacement of the object once the exposure time ended. A RS sensor does not have this feature.

[2] https://fr.ids-imaging.com//technical-article//fr_tech-article-rolling-shutter-sensors.html

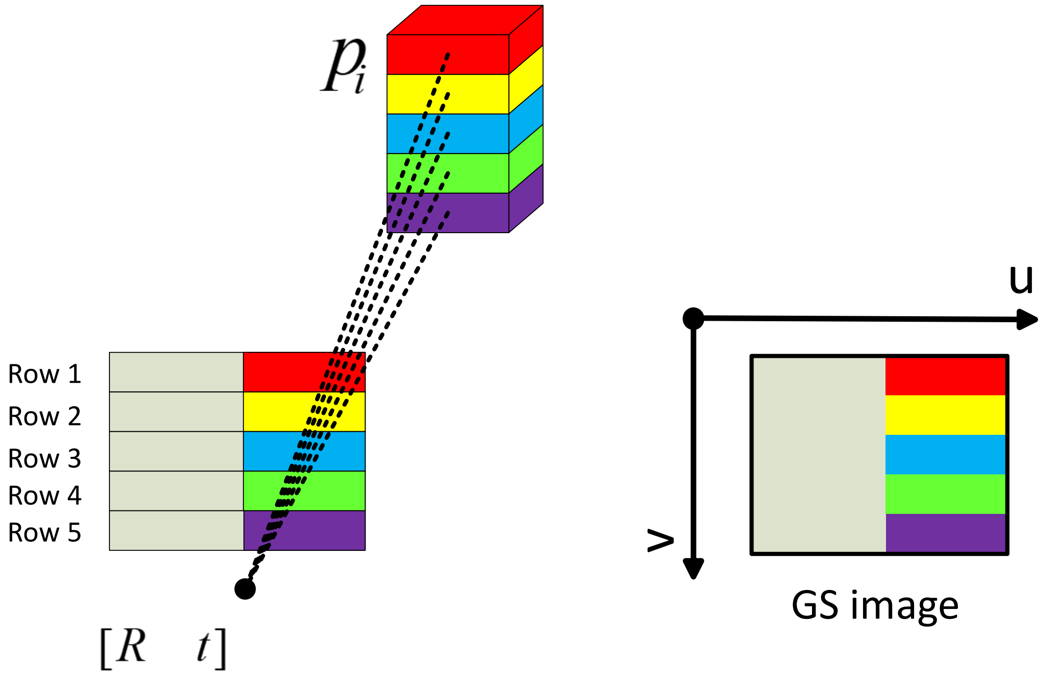


FIGURE 2.4: GS projection model.

2.2 Camera Geometrical Modelling

2.2.1 Pinhole Model

The optical part of this image formation process can be modelled geometrically using the very well-known pinhole model. It describes how a 3D point $\mathbf{Q}_i = [X_i, Y_i, Z_i]^\top$ defined in a camera coordinate systems first projects onto the image plane as image point $\mathbf{q}_i$, then transforms into the image space as an image measurement $\mathbf{m}_i = [u_i, v_i]^\top$ [Hartley and Zisserman, 2003]:

$$s_i \begin{bmatrix} \mathbf{m}_i \\ 1 \end{bmatrix} = s_i \mathbf{K} \mathbf{q_i} = \mathbf{K}\mathbf{Q}_i \tag{2.1}$$

where $s_i \in \mathbb{R}$ is a scale factor and $\mathbf{K}$ is a 3×3 intrinsic parameter matrix which is usually denoted as:

$$\mathbf{K} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \tag{2.2}$$

where c_x and c_y define the centre of image. f_x and f_y are camera's focal length. If the aspect ratio of the sensor's pixel is squared, then $f_x = f_y$.

2.2.2 Camera Pose

As shown in Fig. 2.4, assuming that a GS camera is with a pose $[\mathbf{R}|\mathbf{t}]$, the rotation $\mathbf{R} \in \mathbb{SO}^3$ and the translation $\mathbf{t} \in \mathbb{R}^3$ bring a 3D point $\mathbf{P}_i$ from the world coordinate system to the camera coordinate system as $\mathbf{Q}_i = \mathbf{R}\mathbf{P}_i + \mathbf{t}$. Thus, Eq. (2.1) becomes:

$$s_i \begin{bmatrix} \mathbf{m}_i \\ 1 \end{bmatrix} = \mathbf{K} \begin{bmatrix} \mathbf{R} & \mathbf{t}(\end{bmatrix} \begin{bmatrix} \mathbf{P}_i \\ 1 \end{bmatrix} \tag{2.3}$$

Note that scale s_i is related to the depth of $\mathbf{P}_i$. Thus, Eq. (2.3) can be rewritten as:

$$\begin{gathered} \mathbf{m}_i = \Pi^{GS}(\mathbf{K}\mathbf{Q}_i) = \Pi^{GS}(\mathbf{K} \begin{bmatrix} \mathbf{R} & \mathbf{t} \end{bmatrix} \begin{bmatrix} \mathbf{P}_i \\ 1 \end{bmatrix}) \\ \text{with} \quad \Pi^{GS}([X\ Y\ Z]^\top) = \frac{1}{Z}[X\ Y]^\top \end{gathered} \tag{2.4}$$

Π^{GS} is the GS projection operator.

Camera pose estimation (PnP) is the problem of calculating the pose $[\mathbf{R}|\mathbf{t}]$ of a calibrated camera from n 3D-2D correspondences. Camera pose estimation is important and extensively used in Simultaneous Localization And Mapping (SLAM) for robotics, object or camera localization and Augmented Reality (AR).

2.2.3 Relative Pose

The aim of relative pose estimation is to recover the rotation and translation between two cameras. It is of eminent importance in many computer vision applications such as structure-from-motion (SfM), visual odometry (VO) and SLAM. Without losing in generality, we assume two cameras are with poses $[\mathbf{I}|\mathbf{0}]$ and $[\mathbf{R}|\mathbf{t}]$ respectively. In existing works, two methods are commonly used to solve this problem (estimating $\mathbf{R}$ and $\mathbf{t}$) by using a set of correspondences between two views, namely epipolar geometry and homography.

2.2.3.1 Epipolar geometry.

Each point correspondence $\mathbf{m}_i \leftrightarrow \mathbf{m'}_i$ holds [Hartley and Zisserman, 2003]:

$$\begin{bmatrix} \mathbf{m'}_i^\top & 1 \end{bmatrix} \mathbf{F} \begin{bmatrix} \mathbf{m}_i \\ 1 \end{bmatrix} = 0 \tag{2.5}$$

where $\mathbf{F}$ is the fundamental matrix. 8pt method [Hartley, 1995] and 5pt method [Li and Hartley, 2006] can be used to estimate $\mathbf{F}$. If both the cameras are calibrated, one can calculate the 3×3 essential matrix which involves only motion parameters by using:

$$\mathbf{E} = \mathbf{K'}^\top \mathbf{F}\mathbf{K} = [\mathbf{t}]_\times \mathbf{R} \tag{2.6}$$

Eq. (2.6) can be written::

$$\begin{gathered} \begin{bmatrix} \mathbf{q'}_i^\top \end{bmatrix} \mathbf{E} \begin{bmatrix} \mathbf{q}_i \end{bmatrix} = 0 \\ \text{with} \quad \mathbf{K}^{-1} \begin{bmatrix} \mathbf{m}_i \\ 1 \end{bmatrix} = \mathbf{q}_i \end{gathered} \tag{2.7}$$

Once $\mathbf{E}$ is obtained, we can decompose it into camera relative pose $\mathbf{R}$ and $\mathbf{t}$ by using singular value decomposition (SVD) [Hartley and Zisserman, 2003].

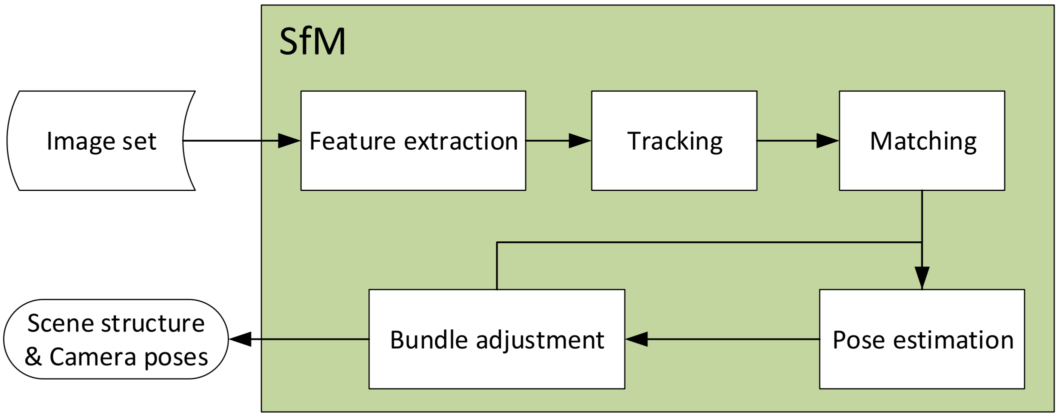


FIGURE 2.5: A simplified overview of classical SfM pipeline.

2.2.3.2 Homography

If all the 3D points $\{\mathbf{P}_i\}$ locate in a plane, each point correspondence holds:

$$\begin{aligned} s_i\mathbf{m}'_i &= \mathbf{H}_{GS}\mathbf{m}_i \\ \text{with} \quad \mathbf{H}_{GS} &= \mathbf{K}'(\mathbf{R}_0 - \frac{\mathbf{t}_0\mathbf{n}^\top}{d})\mathbf{K}^{-1} \end{aligned} \tag{2.8}$$

where $\mathbf{H}_{GS}$ is the 3×3 GS homography matrix, α_i is a scale factor that depends on the depth of $\mathbf{P}_i$ in each camera. $\mathbf{n}$ is the normal vector of the observed plane and d is the distance from the first camera to the plane under the constraint $\mathbf{n}^\top\mathbf{P}_i + d = 0$. $\mathbf{H}$ is usually computed using the Direct Linear Transform algorithm (DLT) with at least four point correspondences. If the two cameras are calibrated, $\mathbf{H}_{GS}$ can be decomposed into $\mathbf{R}$ and $\mathbf{t}$ using SVD [Ma et al., 2012, Malis and Vargas, 2007].

2.2.4 Structure from Motion

Given a set of images, the task of structure-from-motion (SfM) is to compute the camera pose and the scene's 3D structure simultaneously. We review the procedure a classical SfM pipeline in the following:

Problem setting: A group of 3D points $\{\mathbf{P}_i\}$ with $i = 1 \ldots n$ viewed by m cameras with pose $[\mathbf{R}^j|\mathbf{t}^j]$. Let $\left\{\mathbf{m}_i^j\right\}$ be the coordinates of the projection of the i^{th} 3D point onto the j^{th} camera. The SfM problem described as follows: given the group of pixel coordinates $\{\mathbf{m}_i\}$, find the corresponding set of camera pose $[\mathbf{R}^j|\mathbf{t}^j]$ and the scene structure $\{\mathbf{P}_i\}$ such that

$$\begin{bmatrix} \mathbf{m}_i^j \\ 1 \end{bmatrix} \simeq \mathbf{K}_i \begin{bmatrix} \mathbf{R}^j & \mathbf{t}^j \end{bmatrix} \begin{bmatrix} \mathbf{P}_i \\ 1 \end{bmatrix} \tag{2.9}$$

where a collection $\left\{\mathbf{P}_i, [\mathbf{R}^j|\mathbf{t}^j], \mathbf{m}_i^j\right\}$ holds Eq. (2.9) is called a ***configuration***. Thus, we say that configuration $\left\{\mathbf{P}_i, [\mathbf{R}^j|\mathbf{t}^j], \mathbf{m}_i^j\right\}$ explains images $\mathbf{m}_i^j$.

SfM pipeline: As shown in Fig.2.5, a basic SfM pipeline has five steps:

- **Step 1: Keypoint Extraction.** Extracting keypoints from the images. Different features detectors and descriptors exit. SURF points [Bay et al., 2006] were used in the experiments described in this manuscript.
- **Step 2: Matching.** Image point pairs are obtained by matching the feature descriptors. Then the outliers of matching point can be rejected by a geometric verification process such as Random sample consensus (RANSAC) with epipolar constraint [Hartley and Zisserman, 2003].
- **Step 3: Tracking.** We use 2D trackers to record the visibilities of each 3D point over multiple images and its correspondence projection.
- **Step 4: Pose estimation.** Commonly, two or three views are used to initialize 3D structure and relative pose between the camera using epipolar geometry [Hartley and Zisserman, 2003]. Then additional views are added incrementally to expand the existing reconstruction using relative pose estimation or absolute pose estimation [Gao et al., 2003].
- **Step 5: Bundle adjustment.** Typically, every time a new image is added, we will perform a non-linear optimization of the model called bundle adjustment, to refine all unknown parameters $\mathbf{P}_i$ and $[\mathbf{R}^j|\mathbf{t}^j]$.

Actually, this description is a very summarized representation. In practice many additional tasks such as loop closing and scale factor drift handling have to be implemented to make the SfM pipeline work in realistic applications (for example for large scale environments). The input image set of SfM can categorized in two types: *1) Ordered image set.* This commonly can be extracted from a video. However, processing a video sequence frame by frame is with a heavy computational load and time-consuming. *2) Unordered image set.* This is a more general but also more challenging case for SfM since building the correct corresponding matches is difficult. Recently, large scale SfM with unordered image set have been reported in [Agarwal et al., 2009, Heinly et al., 2015]. Unordered image sets can also come from a network composed of multiple cameras.

2.2.4.1 Bundle Adjustment

We commonly perform a GS Bundle Adjustment (BA) as the final step in a SfM pipeline. Given an initial set of 3D points $\mathbf{P}_i$ and camera poses $[\mathbf{R}^j|\mathbf{t}^j]$ and 2D corresponding image features measurements $\mathbf{m}_i^j$, BA aims to find the best 3D structure and camera poses that minimize the sum of squares of reprojection errors as follow:

$$\begin{aligned} \left\{\{\mathbf{P}_i^*\}, [\mathbf{R}^{j*}|\mathbf{t}^{j*}]\right\} &= \arg\min \sum_{j=1}^{m}\sum_{i=1}^{n} V_i^j \left\|\mathbf{e}_i^j\right\|^2 \\ \text{with} \quad \mathbf{e}_i^j &= \tilde{\mathbf{m}}_i^j - \Pi^{GS}(\mathbf{K}^j \begin{bmatrix}\mathbf{R}^j & \mathbf{t}^j\end{bmatrix} \begin{bmatrix}\mathbf{P}_i \\ 1\end{bmatrix}) \end{aligned} \tag{2.10}$$

where $\mathbf{e}_i^j$ is reprojection error, which is the distance between measured image point $\tilde{\mathbf{m}}_i^j$ and the points predicted by reprojection.

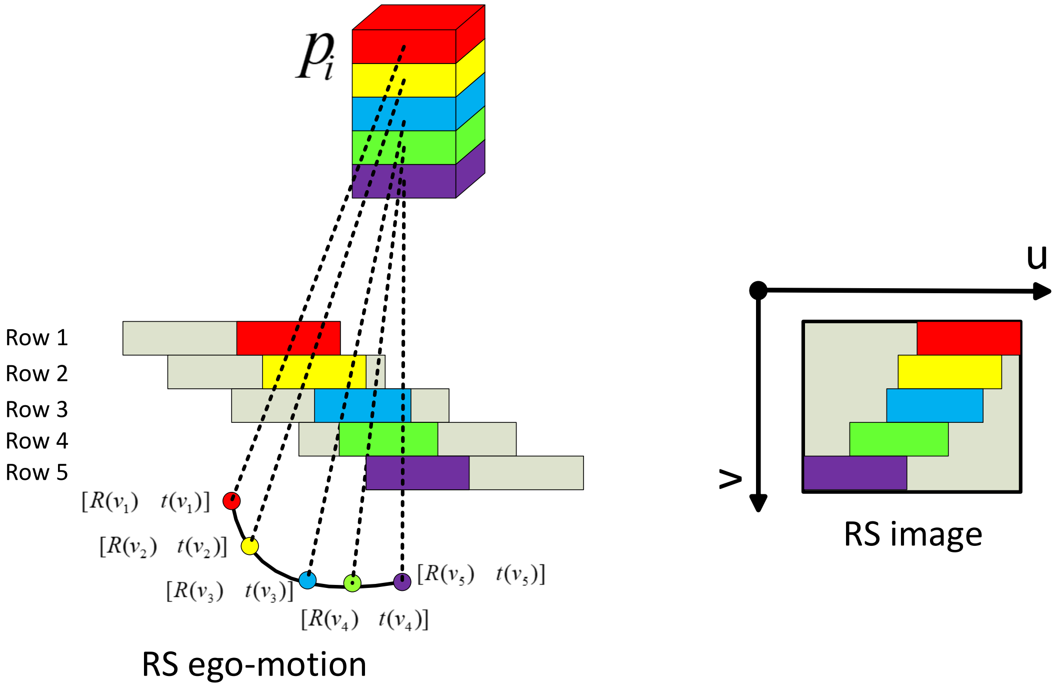


FIGURE 2.6: RS projection model.

2.2.4.2 Degeneracies

Once images $\{\mathbf{m}_i\}$ can be explained by at least two configurations that are not equivalent, we say that these images are ***critical***. Thus, such cases which leads to ambiguities in 3D reconstruction, are termed degenerate.

Degeneracies in SfM with pinhole cameras have been studied for calibrated two views [Maybank, 2012], multiple views [Hartley and Kahl, 2007] and also for uncalibrated sequences [Torr et al., 1999].

2.3 Computer Vision with RS Cameras

2.3.1 Rolling Shutter Definition

The majority of CMOS sensors are inherently with RS mechanism. Therefore, RS is an important topic to study in computer vision, wherein the default camera model is GS.

As shown in Fig. 2.3, the RS slit slides over the sensor exposing each scanline sequentially (commonly from the top to the bottom of the image). Images captured by moving RS cameras produce distortions (e.g. wobble, skew), which defeat the classical GS geometric models in 3D computer vision. Thus, new methods adapted to RS cameras are strongly desired.

2.3.2 RS Projection Model

As shown in Fig. 2.6, with a moving RS camera, each row will be captured in turn and thus with a different pose during frame exposure, yielding a new projection operator Π^{RS}. Thus, Eq. (2.4) becomes:

$$\mathbf{m}_i = \Pi^{RS}(\mathbf{K}\mathbf{Q}_i) = \Pi^{GS}(\mathbf{K}\mathbf{Q}_i^{RS}) = \Pi^{GS}(\mathbf{K}[\mathbf{R}(v_i) \quad \mathbf{t}(v_i)]\mathbf{P}_i) \tag{2.11}$$

where $\mathbf{R}(v_i)$ and $\mathbf{t}(v_i)$ define the camera pose when the image row of index v_i is acquired. Therefore, a static 3D point $\mathbf{P}_i$ in world coordinates is transformed into $\mathbf{Q}_i^{RS}$, instead of $\mathbf{Q}_i$, in camera coordinates.

2.3.3 RS Instantaneous-motion Models

Instantaneous camera motion: During the acquisition of a single shot, the rotation $\mathbf{R}(v_i)$ and the translation $\mathbf{t}(v_i)$ of an RS camera are functions of row index v_i. Note that this motion during the image acquisition is different from the term 'motion' in "structure from motion (SfM)", which refers to the relative pose between two images (views). Many naming have been used in previous works such as kinematic model [Ait-Aider et al., 2006], RS model [Albl et al., 2016b], RS camera model [Dai et al., 2016] or camera motion [Rengarajan et al., 2016, Purkait et al., 2017]. For the consistency and readability, we name the camera motion during the image acquisition as ***instantaneous-motion*** in this manuscript, and keep denote the camera relative pose as ***motion***. Although the term "instantaneous" is not entirely appropriate, it quite correctly reflects the fact that the exposure time is of course very short compared to the unpredictable changes in camera trajectory.

Using the RS projection model with a new pose parameters for each feature as described by section 2.1.1 would lead to over parametrized systems whilst overdetermined systems are required to solve most of computer vision problems. Independently estimating of the pose of each row is an ill-posed problem. Therefore, it seems clear that there is a need for a more minimal parametrization that describes the relationship between the poses of image features.

In this section, we introduce various RS instantaneous-motion models. Since the image acquisition time is short (usually a fraction of a second), it is common to assume that the instantaneous-motion of an RS camera is smooth. Note that this assumption is on the camera instantaneous-motion only, without constraint on the motion (relative pose) between views.

We will present the different models of instantaneous movement used in RS methods. We will then discuss the relevance and applicability of each model depending on the application context.

2.3.3.1 Constant Velocity Motion

Constant velocity motion assumes that the camera is with constant direction and magnitude of translational and rotational velocity during the acquisition. Now, we introduce different parametrizations that can then be used for the constant translational and rotational velocity.

Constant translational velocity: The constant translational motion assumes a linearized translational motion with constant velocity as:

$$\mathbf{t}(v_i) = \mathbf{t}_0 + \mathbf{d}v_i \tag{2.12}$$

where $\mathbf{t}_0$ is the translation of first row while $\mathbf{d} = [d_x, d_y, d_z]^\top$ is the translational velocity.

Constant rotational velocity: Three main parametrizations have been used for the constant rotational velocity in the existing works.

1. **Rodrigues' Formula.** Rodrigues formula is to define a rotation matrix. In RS case, we assume that the rotation is with instantaneous rotational speed ω around an instantaneous axis of unit vector $\mathbf{a} = [a_x, a_y, a_z]^\top$. Then the rotations during the acquisition are obtained as follows [Ait-Aider et al., 2006, Ait-Aider et al., 2007]:

$$\mathbf{R}(v_i) = (\mathbf{a}\mathbf{a}^\top(1 - \cos(\omega v_i)) + \mathbf{I}\cos(\omega v_i) + [\mathbf{a}]_\times \sin(\omega v_i))\mathbf{R}_0 \tag{2.13}$$

   where $\mathbf{t}_0$ is the translation of first row, $[\mathbf{a}]_\times$ is the antisymetric matrix of $\mathbf{a}$ and $\mathbf{I}$ is a 3×3 identity matrix.

2. **Linearized model.** Most of rotation parametrizations leads to highly nonlinear equations which can be solved by nonlinear optimization techniques for the last refinement steps. However, in order to reduce the complexity of the model, we can use a linearization of the rotation matrices. Assuming that the rotation during the acquisition is small, the first order Taylor expansion of Eq. (2.13) gives a polynomial approximation of the rotations as:

$$\begin{aligned} &\mathbf{R}(v_i) = (\mathbf{I} + [\boldsymbol{\omega}]_\times v_i)\mathbf{R}_0 \\ \text{with} \quad &\boldsymbol{\omega} = \omega\mathbf{a} \qquad \omega = \|\boldsymbol{\omega}\| \\ &[\boldsymbol{\omega}]_\times = \begin{bmatrix} 0 & -\omega z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} \end{aligned} \tag{2.14}$$

   where $\boldsymbol{\omega} = [\omega_x, \omega_y, \omega_z]^\top$ is the rotational velocity vector. Such instantaneous-motion model with the RS rotation linearized, was used in [Magerand et al., 2012, Dai et al., 2016, Albl et al., 2016b, Magerand et al., 2012, Lao et al., 2018a, Lao et al., 2018b].

3. **Double linearized model.** Eq. (2.14) has been further linearized in [Albl et al., 2015, Albl et al., 2016a] using the first order Taylor expansion as:

$$\mathbf{R}(v_i) = (\mathbf{I} + [\boldsymbol{\omega}]_\times v_i)(\mathbf{I} + [\mathbf{r}]_\times) \tag{2.15}$$

   where $\mathbf{r} = [r_x, r_y, r_z]^\top$ is the Euler angles component about axes x, y, z. This approximation requires the initial global rotation $\mathbf{R}_0$ is small.

2.3.3.2 Constant Acceleration Motion

Constant acceleration motion relaxes the constant velocity assumption. Specifically, it assumes constant direction of translational and rotational velocity, but allows its magnitude to change gradually. Different parametrizations of the constant translational and rotational acceleration can be used.

Constant translational acceleration: The constant translational acceleration motion has been considered first time in [Zhuang et al., 2017] to enhance the generality of the RS instantaneous-motion model as:

$$\mathbf{t}(v_i) = \mathbf{t}_0 + \mathbf{d}v_i + \frac{1}{2}\mathbf{a}{v_i}^2 \tag{2.16}$$

where $\mathbf{a}$ is the acceleration.

Cayley transform: Cayley transform [Golub and Van Loan, 2012] was used to express the accelerated rotations [Rengarajan et al., 2016, Purkait et al., 2017].

We denote the Rodrigues parameterization [Morawiec, 2003] of each row as $r_A = [r_x, r_y, r_z]^\top$ which is defined by:

$$\begin{cases} r_x = \alpha + a_1 t + \cdots + a_{\dot{n}} t^{\dot{n}} \\ r_y = \beta + b_1 t + \cdots + b_{\dot{n}} t^{\dot{n}} \\ r_z = \gamma + c_1 t + \cdots + c_{\dot{n}} t^{\dot{n}} \end{cases} \tag{2.17}$$
$$\text{with} \quad a_i,\ b_i \text{ beeing real coefficients and } t = \frac{v_i - 1}{H}$$

Then the rotation matrix can be calculated by using Cayley transform as follows:

$$\mathbf{R}(r_A) = \frac{1}{Z}\begin{bmatrix} 1 + {r_x}^2 - {r_y}^2 - {r_z}^2 & 2r_x r_y - 2r_z & 2r_y + 2r_x r_z \\ 2r_z + 2r_x r_y & 1 - {r_x}^2 + {r_y}^2 - {r_z}^2 & 2r_y r_z - 2r_x \\ 2r_x r_z - 2r_y & 2r_y r_z + 2r_x & 1 - {r_x}^2 - {r_y}^2 + {r_z}^2 \end{bmatrix} \tag{2.18}$$
$$\text{with} \quad Z = 1 + {r_x}^2 + {r_y}^2 + {r_z}^2$$

Therefore, estimation of the instantaneous rotation is equivalent to the estimation of $3(\dot{n}+1)$ motion parameters $[\alpha, a_1, \ldots, a_{\dot{n}}; \beta, b_1, \ldots, b_{\dot{n}}; \gamma, c_1, \ldots, c_{\dot{n}};]$. For instance, $\dot{n} = 2$ are used in [Rengarajan et al., 2016, Purkait et al., 2017].

2.3.3.3 Interpolated Poses

Instead of constraining the velocity during the acquisition period, one can also constrain the instantaneous-motion by interpolating the poses between two successive frames. A very popular interpolation method for rotations is SLERP [Shoemake, 1985] which has been used for RS images in [Hedborg et al., 2012]. Assuming two successive frames with rotations $\mathbf{R}^1$ and $\mathbf{R}^2$ respectively, the rotations of rows v_i in 1st image can be estimated as follows:

$$\begin{aligned} \mathbf{R}(v_i) &= \mathbf{R}^1 \frac{\sin(\boldsymbol{\Omega} - t\boldsymbol{\Omega})}{\sin(\boldsymbol{\Omega})} - \mathbf{R}^2 \frac{\sin(t\boldsymbol{\Omega})}{\sin(\boldsymbol{\Omega})} \\ \text{with} \quad \boldsymbol{\Omega} &= \arccos({\mathbf{R}^1}^\top \mathbf{R}^2) \\ t &= \frac{v_i - 1}{H} \end{aligned} \tag{2.19}$$

where H is the number of rows in the first image.

2.3.3.4 Non-general-motion Models

Models in this category restrict the instantaneous-motion (e.g. ignore either translational or rotational velocity) to obtain a simplified model for specific applications. We introduce three such models that were used in existing works.

(1) Pure translation model. Authors in [Sau, 2013, Saurer et al., 2016] consider pure translation with constant translational velocity defined in Eq. (2.12). However, this assumption approximately holds only for limited scenarios such as car driving straightly.

(2) Pure rotation model. This model is based on the assumption that the rotations play a major role for RS effect in images captured by hand-held cameras [Ringaby and Forssén, 2012] and even by vehicle cameras [Duchamp et al., 2015]. Thus, many previous works use pure rotation instantaneous-motion model and do not consider the translational velocity [Rengarajan et al., 2016, Ito and Okatani, , Purkait et al., 2017, Lao and Ait-Aider, 2018].

(3) Ackermann motion. [Purkait and Zach, 2017] use the conventional Ackermann steering principle, which models a four wheels vehicle rolling around a common point during a turn, to constraint the RS instantaneous-motion. This principle holds for any vehicle which ensures all the wheels exhibit a rolling motion.

2.3.3.5 Discussion

It is common that the higher order of the instantaneous-motion model leads to more realistic description of the camera motion during image acquisition. However, at the same time, despite the advent of modern new polynomial equation solvers [Henrion and Lasserre, 2003, Kukelova et al., 2008], solving such model is also time consuming and unstable (local minimas when performing a non-linear optimization and numerically unstable when solving a polynomial system). In this manuscript, we use the linearized model described by Eq. (2.12) and (2.14) as follows:

$$\mathbf{R}(v_i) = (\mathbf{I} + [\boldsymbol{\omega}]_\times v_i)\mathbf{R}_0 \qquad \mathbf{t}_{v_i} = \mathbf{t}_0 + \mathbf{d}v_i \tag{2.20}$$

because we consider that it is a good compromise between accuracy and complexity as it has been justified in [Magerand et al., 2012].

Note that both translational and rotational velocities express the displacements per row instead of displacements per time unit (e.g. second or minute) [Ait-Aider et al., 2006]. The transformation between these two expressions of velocities requires the exposure time per row, which can be calibrated by imaging a flashing light source with known frequency [Meingast et al., 2005].

2.4 Problem Statements and Previous Works

After an analysis of the state of the art in 3D vision with RS cameras, we identified several themes for which improvements could be made both in terms of theoretical formalization and in terms of proposing pragmatic and practical solutions. We address four important but challenging problems in RS 3D vision:

2.4.1 RS Correction

Definition: By given single or multiple distorted RS images, the aim of RS correction is to remove the distortions and recover the corresponding **undistorted GS images**.

Previous works: The state-of-the-art works for RS image rectification can be divided into three main classes:

1. **Video-based methods:** Methods of [Liang et al., 2008, Forssén and Ringaby, 2010, Kim et al., 2011,Grundmann et al., 2012,Zhuang et al., 2017] try to recover the geometry between RS frames first, then to compensate RS effects by scanline realignment or RS-aware warping.

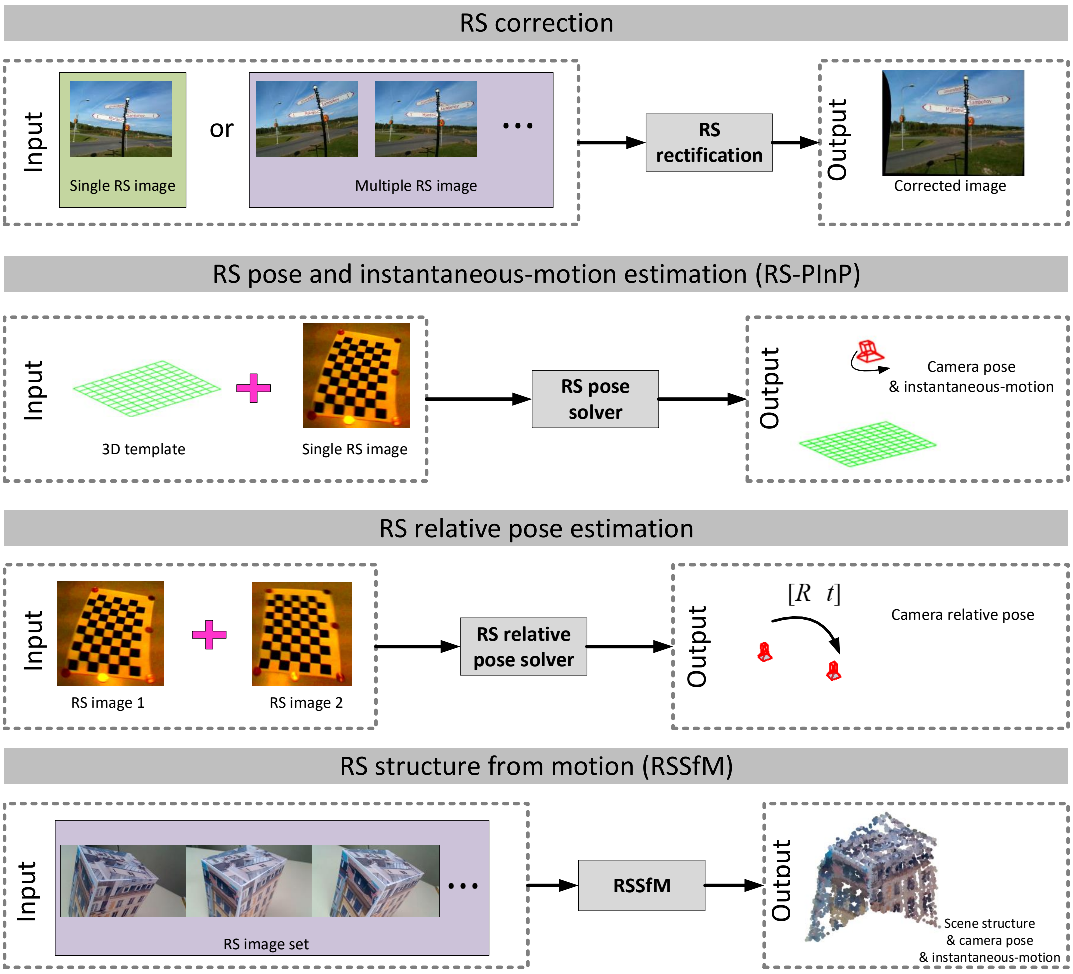


FIGURE 2.7: The four RS vision problems we address in this manuscript.

2. **Gyroscopes-based methods:** Methods of [Hee Park and Levoy, 2014,Jia and Evans, 2012,Patron-Perez et al., 2015] utilize gyroscopes to measure camera instantaneous-motion during acquisition and compensate RS effects directly.

3. **Straightness constraint-based methods:** After line features had been explored in object motion estimation [Ait-Aider et al., 2007], 3D straight lines have been also used for solving single RS image correction problem based on straightness [Rengarajan et al., 2016] or vanishing direction [Purkait et al., 2017] constraints.

4. **Machine learning methods:** Rengarajan et al. first developed a learning-based single RS correction method using Convolutional Neural Network (CNN) in [Rengarajan et al., 2017].

2.4.2 RS Pose and Instantaneous-motion Estimation

Definition: The aim of **RS pose** and **instantaneous-motion** estimation (**RS-PInP**) is to calculate the pose of a calibrated RS camera and its instantaneous-motion from n 3D-2D correspondences.

Previous works: The existing works for RS-PEnP problem can be divided into two main classes:

1. **Non-linear optimization methods:** In [Ait-Aider et al., 2006] authors first solved the RS-PEnP problem using a non-linear optimization with the instantaneous-motion assumption modelling by Rodrigues' Formula and constant translation velocity. This method had been further extended to use lines instead of point features in [Ait-Aider et al., 2007]. Magerand et al. [Magerand et al., 2012] present a polynomial projection model for RS cameras and propose the constrained global optimization of its parameters by means of a semidefinite programming problem obtained from the generalized problem of moments method.

2. **Minimal, non-iterative methods:** In [Ait-Aider et al., 2006], a linear method using 8.5pt had been presented by assuming a planar scene. Saurer et al. [Saurer et al., 2015] propose a minimal solver to estimate RS camera pose based on the translation-only model with at least 5 3D-2D correspondences. Albl et al. propose a minimal and non-iterative solution to the RS-PEnP problem called R6P [Albl et al., 2015]. Albl et al. [Albl et al., 2016a] also propose another minimal solver, which requires at least 5 3D-2D matches but requires to the assistance of inertial measurement units (IMUs).

2.4.3 RS Relative Pose Estimation

Definition: The goal of RS relative pose estimation is to estimate the **relative pose** (rotation and translation) between two RS images and also to calculate the **instantaneous-motions** of both cameras by using feature correspondences.

Previous works: [Feldman et al., 2003] is the first work develop the epipolar geometry of the translation-only crossed-slits cameras which related to RS projection model. [Dai et al., 2016] is the only work that investigate the epipolar geometry of RS cameras and provide solutions to the RS relative pose problem. The authors derived the 5×5 and 7×7 essential matrices for pure translation and uniform linearized model respectively. They also introduce 20pt and 44pt linear solvers to estimate the two types of RS essential matrices, which then are needed to be refined by nonlinear solvers by using Sampson error. Linear algorithms for the extraction of the relative pose and instantaneous-motion from these matrices are also presented.

2.4.4 RS structure from Motion

Definition: The goal of RS structure from motion (**RSSfM**) is to reconstruct the ***3D structure*** and to estimate the **camera poses** as well as the **camera instantaneous-motions** during each of the image readouts.

Previous works:

1. **RSSfM using video sequence:** The methods in [Hedborg et al., 2011] use an RS video sequence to solve RSSfM by assuming smooth camera motion between every consecutive frames. [Im et al., 2018] proposes a dense 3D reconstruction method from narrow-baseline RS image sequences.

2. **Optical flow based RSSfM:** In [Zhuang et al., 2017], 8pt and 9pt linear solvers were developed to recover the relative pose of a RS camera that undergoes constant velocity and acceleration motion respectively.

3. **RSSfM with additional sensors:** [Klingner et al., 2013] proposes a vehicle-based RSSfM method for large scale scene by exploiting a good relative pose prior using additional sensors such as GPS and IMU.

4. **Stereo RSSfM:** [Ait-Aider and Berry, 2009] first studied the calibrated RS stereo system, and presented a method for object structure and kinematics estimation by using iterative optimization. [Sau, 2013] addressed the stereo RSSfM under pure translation instantaneous-motion model.

5. **Unordered RSSfM:** The methods in [Ito and Okatani,] attempt to solve RSSfM by establishing an equivalence with self-calibrated SfM based on the pure rotation instantaneous-motion model and affine camera assumption. [Albl et al., 2016b] also addressed the unordered RSSM problem and pointed out common degeneracies of RSSfM.

Chapter 3

RS Correction and SfM using Straightness Constraint

3.1 Introduction

Straight lines frequently appear in man-made environments such as urban or indoor scenes. Furthermore, using straight lines as features offers several advantages such as detection accuracy and the possibility to handle partial occlusions. Thus, line features have been used for various GS vision application such as pose estimation, SfM, SLAM, visual odometry (VO). However, when using RS cameras, straight segments may be rendered as curves under different kinematic models. Thus, classical GS based vision methods using lines suffers from RS effect. To the best of our knowledge, except in [Ait-Aider et al., 2007, Rengarajan et al., 2016, Purkait et al., 2017], line features have never been used for RS vision applications.

In this section, we present two methods both using a set of image curves, basing on the knowledge that they correspond to 3D straight lines. Namely, a robust method which compensates RS distortions in a single image, and a robustified 3-step RSSfM method (Fig. 3.1). Unlike in existing work, no a priori knowledge about the line directions (e.g. Manhattan World assumption) is required.

Chapter outline. The rest of this chapter organized as:

- First, we show that the parameterization of the projection of a 3D straight line leads to a first (section 3.3.2), second (section 3.3.4.1 and 3.3.4.2) or third degree (section 3.3.3) polynomial depending on the kinematic model considered during image acquisition (table. 3.1).
- Second we propose a theoretical linear rotational velocity extraction algorithm using at least 16 image curves (section 3.4.1), then we derive a more practical version using 4 curves (section 3.4.3). We also analyse the degenerate cases in the rotational velocity extraction method (section 3.4.4).
- Moreover, we propose a RANSAC-like strategy to select image curves which really correspond to 3D straight lines and reject those corresponding to actual curves in 3D world (section 3.5.0.2). This automatic feature selection is crucial because it makes the method robust to noise and also fully automated. Then the RS image can be finally corrected by compensating the estimated camera instantaneous-motion.
- Finally we propose a 3-step RSSfM method (section 3.6). In step one, each image is corrected using lines while rotational speed is computed. In step two, the translational speed as well as the motion between cameras is recovered. At the end,

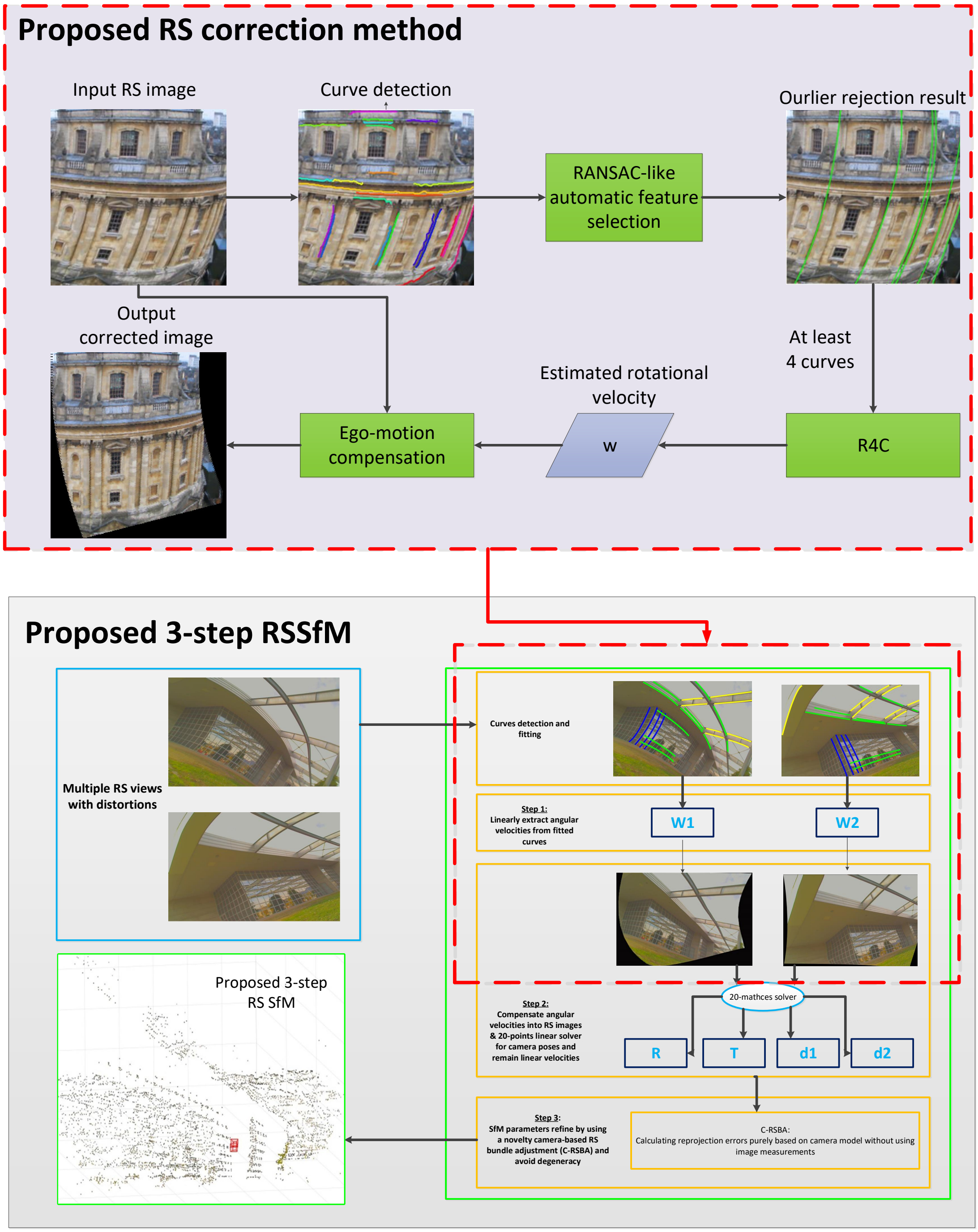
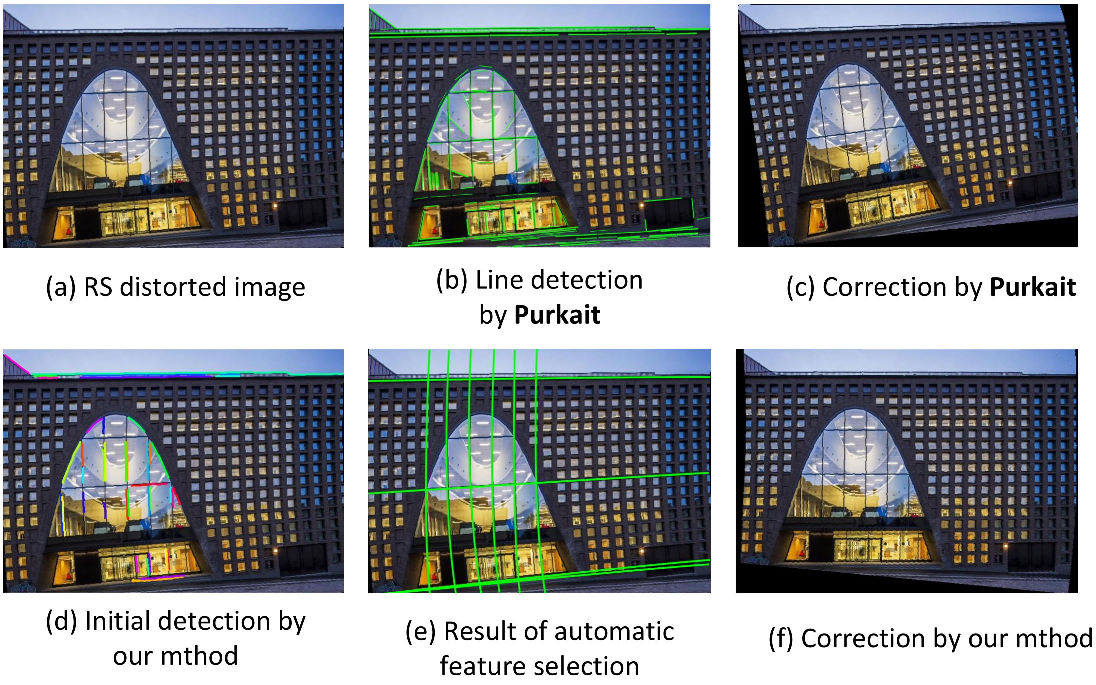

Proposed RS correction method
Input RS image
Curve detection
RANSAC-like automatic feature selection
Ourlier rejection result
Output corrected image
At least 4 curves
Estimated rotational velocity
Ego-motion compensation
w
R4C
Proposed 3-step RSSfM
Multiple RS views with distortions
Curves detection and fitting
Step 1:
Linearly extract angular velocities from fitted curves
W1
W2
20-mathces solver
Step 2:
Compensate angular velocities into RS images & 20-points linear solver for camera poses and remain linear velocities
R
T
d1
d2
Step 3:
SfM parameters refine by using a novelty camera-based RS bundle adjustment (C-RSBA) and avoid degeneracy
C-RSBA:
Calculating reprojection errors purely based on camera model without using image measurements
Proposed 3-step RS SfM

FIGURE 3.1: Overview of the proposed RS correction and RSSfM method using straightness constraint.



(a) RS distorted image

(b) Line detection by **Purkait**

(c) Correction by **Purkait**

(d) Initial detection by our mthod

(e) Result of automatic feature selection

(f) Correction by our mthod

FIGURE 3.2: (a) An example of a distorted RS image. (b)Arc segments detected by **Purkait** [Purkait et al., 2017] using LSD detector [Von Gioi et al., 2010] where outliers are also considered in correction. In contrast, the automatic feature selection in our method successfully filters outliers among detected candidate curves (d) and obtains correctly fitted curves (e). Final corrections by **Purkait** and our method are shown on (c) and (f).

all the parameters are refined using a new BA technique which enables to avoid degeneracy reported in the state-of-the-art (section 3.6.3).

- A comparative experimental study with both synthetic and real data from famous benchmarks shows that the proposed method outperforms all the existing techniques from the state-of-the-art for both RS correction and RSSfM problem (section 3.7).

3.2 Related Work and Motivation

3.2.1 Related Work for RS correction

The related works for RS correction are discussed in section 2.4.1 suffer from the following disadvantages:

(1) Simplified instantaneous-motion model: Methods of [Rengarajan et al., 2016, Rengarajan et al., 2017] correct RS by making the output image visually pleasant for human without considering correctness of projection geometry. Thus, camera x and y axis instantaneous-rotations are neglected which may lead to geometrical inconsistencies.

(2) Manhattan world (MW) assumption: Methods of [Rengarajan et al., 2016, Purkait et al., 2017] require that the MW assumption is valid. This requires that the images feature at least two orthogonal vanishing directions. Beside the difficulty to find such image features, nonorthogonal 3D lines and 3D curves are also common in urban area. Since both methods lack of outlier filter process, strong deformations occur in the final correction (Fig. 3.2(b)(c)).

(3) Time-consuming: Nonlinear iterative solutions are used in [Rengarajan et al., 2016, Purkait et al., 2017] which may be time-consuming and suffer from the risk to fall into local minima due to the absence of a good initial guess estimation process.

In order to overcome disadvantages of the techniques from the state-of-the art, we present a method which enables us to compensate RS distortions using a set of image curves which correspond to 3D straight lines with free unknown directions. The method estimates linearly the camera instantaneous-motion and then compensates image distortions according to the computed rotation. The presence of outlier curves which do not correspond to actual 3D straight lines is also addressed (Fig. 3.2(e)(f)).

3.2.2 Related Work for RSSfM

Due to the complexity and the high non-linearity of RS perspective projection model, strong assumptions which usually do not hold in practice, have been made in existing literatures in order to solve the SfM problem with RS cameras.

Smooth motion assumption. Some approaches require continuity and "smoothing" of the movement during the shooting but also between the views thus imposing an acquisition at very high frequency [Hedborg et al., 2012, Kim et al., 2016] which makes both data transferring and processing very time and memory consuming without mentioning the case where different cameras with wide baselines are used. Other approaches consider simplified movements such as linear motion (pure translation) [Sau, 2013], pure rotation [Ito and Okatani,] or small angular velocity [Dai et al., 2016]. We believe that a method based on a more general kinematic model and which handles wide baselines would give significant improvement not only in terms of accuracy of pose and motion estimation, but also in terms of automatic data matching performances (namely outliers rejection).

Degeneracies in RSSfM. With numerous parameters and highly non linear projection model, problems of local minima occur more frequently, for instance in bundle adjustment [Hedborg et al., 2012]. RS degeneracies were firstly reported in [Ait-Aider and Berry, 2009] showing that a linear motion (pure translation) nearly parallel to the baseline gives an infinite number of solutions due to the coupling between shape and motion parameters. [Albl et al., 2016b, Ito and Okatani,] analyzed the case of planar degeneracy which occurs most often for RS SfM and prove that images captured by cameras having parallel read-out directions is a critical motion sequence (CMS) with specific angular-velocities as degenerate solutions. They both suggested that it can be avoided by using RS images with different readout directions, which is not a convenient solution for practical applications.

RS vision applications using lines. [Rengarajan et al., 2016] used line features to estimate instantaneous-motion of cameras by extracting angular velocity from curves with pre-knowledge of corresponding directions in 3D space. The aim of this work is to rectify the

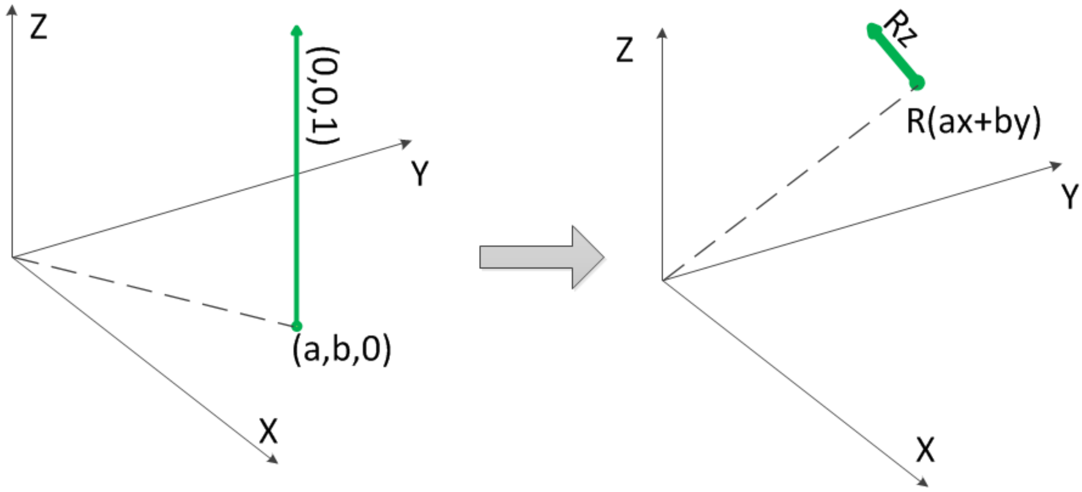


FIGURE 3.3: **3D line representation.** The line can be treated as a line parallel to Z-axis passing through point $(a, b, 0)$ within XY-Plane (green line shown in the left figure) which is then rotated by $\mathbf{R}$ to a new position (Shown on the right part). Thus, the final straight line passes through point $\mathbf{R}(a\mathbf{x} + b\mathbf{y})$, and is heading $\mathbf{Rz}$, where $\mathbf{x} = [1, 0, 0]^\top$, $\mathbf{y} = [0, 1, 0]^\top$, $\mathbf{z} = [0, 0, 1]^\top$ and $\mathbf{R} \in \mathbb{R}^3$ indices the rotation.

image for better visualization by fitting curves with larger length and specific angle on image assuming a pure rotational motion model and using an iterative optimization of a nonlinear function. Thus, the recovered angular-velocities in this work may be far away from ground-truth as long as the results are visually acceptable, which is not sufficient to be used in RSSfM.

3.3 Straight Line Projection with RS

Context. One way to handle problems of degeneracy and local minima mentioned in section. 3.2.2, consists in adding constraints on scene geometry. But the constraint should be convenient and feasible in practical situations. Straight lines can be used to partially constrain the geometry of a scene. Advantages of using line features in computer vision are known very well (vanishing point detection, uncoupling rotation and translation parameters etc.). In the case of a moving RS camera, straight lines do not project as straight lines anymore but as curves whose shape depends from the motion during image scanning. Thus, motion parameters are hiding in the deviation from those curves to a straight line. This is the basis of the 'Straight line have to be straight' principle used in [Devernay and Faugeras, 2001] to remove radial distortion effects.

3.3.1 3D Line Representation

In this paper we adopt the convenient formulation used in [Schindler et al., 2006] and which represents a 3D straight line in $\mathbb{R}^3$ as a tuple $\mathfrak{L} = <\mathbf{R}, (a, b)) > \in SO^3 \times \mathbb{R}^2$. $\mathfrak{L}$ defines an algebraic set which is a 4 dimensional manifold embedded in $SO^3 \times \mathbb{R}^2$ with 4 degrees of freedom (DoF) as illustrated in Fig. 3.3. Note that under this parametrization, $\mathbf{R} \in SO^3$ but there are just two DoF since we fix the yaw angle as 0. In other word, a general rotation matrix is defined by $\mathbf{R}^{\text{general}} = \mathbf{R}_z(\alpha)\mathbf{R}_y(\beta)\mathbf{R}_x(\gamma)$, however, the $\mathbf{R}$ in our parametrization is defined as: $\mathbf{R} = \mathbf{R}_y(\beta)\mathbf{R}_x(\gamma)$.

3.3.2 3D Line Projection with GS Camera

Assuming a calibrated camera, intrinsic matrix $\mathbf{K}$ is known. [Schindler et al., 2006] prove that the projection of a 3D line into a GS camera image can be divided into three main steps:

Transformation into camera coordinate frame: We denote a 3D line in the world coordinate system as $< \mathbf{R}_w, (a_w, b_w)) >$ and the transformation between the camera coordinate frame and the world frame as $\mathbf{R}_c^w$ and $\mathbf{t}_c^w$. The 3D straight line can be expressed in the camera coordinate system as:

$$\begin{aligned} \mathbf{R}_c &= \mathbf{R}_c^w \mathbf{R}_w \\ \mathbf{t}_c &= \begin{bmatrix} t_x & t_y & t_z \end{bmatrix}^\top = (\mathbf{R}_w)^\top \mathbf{t}_c^w \\ (a_c, b_c) &= \left(a_w - t_x, b_w - t_y\right) \end{aligned} \tag{3.1}$$

Perspective projection: The direction $\mathbf{m}_{cip} = \begin{bmatrix} m_x & m_y & m_z \end{bmatrix}^\top$ of a straight line on the image denoted as $m_x u + m_y v + m_z = 0$ within a plane at $z = 1$ in the camera frame can be calculated by the cross product of $\mathbf{R}_z$ and $\mathbf{R}_c(a_c x + b_c y)$:

$$\mathbf{m}_{cip} = a_c \mathbf{R}_{c2} - b_c \mathbf{R}_{c1} \tag{3.2}$$

Where $\mathbf{R}_{c2}$ and $\mathbf{R}_{c1}$ are the second and first columns of $\mathbf{R}_c$.

Image space projection: Image lines can be obtained as: $\mathbf{m}_{ci} = \left(\mathbf{K}^\top\right)^{-1} \mathbf{m}_{cip}$. Finally, we can write a projected 2D line in image as follows:

$$^{GS}\mathrm{F}_1 u +^{GS} \mathrm{F}_2 v +^{GS} \mathrm{F}_3 = 0 \tag{3.3}$$

3.3.3 3D Line Projection with Uniform RS Instantaneous-motion Model

Under the more realistic assumption of a uniform motion with both rotational and translational velocities, modelling by linearized instantaneous-motion described by Eq. (2.20), Eq. (3.1) becomes:

$$\begin{aligned} \mathbf{R}_c &= ((\mathbf{I} + [\boldsymbol{\omega}]_\times v])\mathbf{R}_w^c))^\top \mathbf{R}_w \\ \mathbf{t}_c &= \begin{bmatrix} t_x & t_y & t_z \end{bmatrix}^\top = (\mathbf{R}_w)^\top (\mathbf{t}_c^w + \mathbf{d}v) \\ (a_c, b_c) &= \left(a_w - t_x, b_w - t_y\right) \end{aligned} \tag{3.4}$$

from which we finally obtain a cubic curve:

$$^{Unif}F_1 v^3 +^{Unif} F_2 v^2 u +^{Unif} F_3 v^2 +^{Unif} F_4 vu +^{Unif} F_5 v +^{Unif} F_6 u +^{Unif} F_7 = 0 \tag{3.5}$$

where the seven coefficients are determined by $\mathbf{K}$, 3D line parameters, camera pose and kinematic parameters $(\mathbf{d}, \boldsymbol{\omega})$.

Derivation of Eq. (3.5). We first denote $(\mathbf{K}^\top)^{-1}$ by using the components of the intrinsic matrix $\mathbf{K}$ as follows:

$$(\mathbf{K}^\top)^{-1} = \begin{vmatrix} f_x & 0 & 0 \\ 0 & f_y & 0 \\ c_x & c_y & 1 \end{vmatrix} \tag{3.6}$$

Based on Eq. (3.4), in order to make it easier to derived curve expression, we let:

$$\begin{aligned} \mathbf{R}_c &= ((\mathbf{I} + [\boldsymbol{\omega}]_\times v)\mathbf{R}^c_w)^\top \mathbf{R}_w = \mathbf{R}^w_c \mathbf{R}_w + \mathbf{R}^w_c [\boldsymbol{\omega}]_\times \mathbf{R}_w v = \mathbf{A} + \mathbf{B}v \\ a_c &= a_w - \mathbf{R}^\top_{w1}\mathbf{t}^w_c - \mathbf{R}^\top_{w1}\mathbf{d}v = C^a - D^a \\ b_c &= b_w - \mathbf{R}^\top_{w2}\mathbf{t}^w_c - \mathbf{R}^\top_{w2}\mathbf{d}v = C^b - D^b \end{aligned} \tag{3.7}$$

where, $\mathbf{A}$ and $\mathbf{B}$ are two 3×3 matrices, C^a, C^b, D^a and D^b are scalar variables. Thus, the direction vector of straight line $\mathbf{m}_{cip}$ are now also determined by row-index v:

$$\begin{gathered} \begin{bmatrix} m^x_{cip} \\ m^y_{cip} \\ m^z_{cip} \end{bmatrix} = \mathbf{L} \begin{bmatrix} v^2 \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} L^x_2 & L^x_1 & L^x_0 \\ L^y_2 & L^y_1 & L^y_0 \\ L^z_2 & L^z_1 & L^z_0 \end{bmatrix} \begin{bmatrix} v^2 \\ v \\ 1 \end{bmatrix} \\ \text{with} \quad \mathbf{L} = \begin{bmatrix} D^bB_{11} - D^aB_{12} & C^aB_{12} - D^aA_{12} - C^bB_{11} + D^bA_{11} & C^aA_{12} - C^bA_{11} \\ D^bB_{21} - D^aB_{22} & C^aB_{22} - D^aA_{22} - C^bB_{21} + D^bA_{21} & C^aA_{22} - C^bA_{21} \\ D^bB_{31} - D^aB_{32} & C^aB_{32} - D^aA_{32} - C^bB_{31} + D^bA_{31} & C^aA_{32} - C^bA_{31} \end{bmatrix} \end{gathered} \tag{3.8}$$

where the 3×3 auxiliary matrix $\mathbf{L}$ consisted by nine variables L^x_0, L^x_1, L^x_2, L^y_0, L^y_1, L^y_2, L^z_0, L^z_1 and L^z_2 in order to do further derivations. Now we substitute $\mathbf{m_{cip}}$ in Eq. (3.3) by Eq. (3.8). We obtain the expression of a cubic curve instead of a straight line as follows:

$$\begin{bmatrix} f_yL^y_2 \\ f_xL^x_2 \\ f_yL^y_1 + c_xL^x_2 + c_yL^y_2 + L^z_2 \\ f_xL^x_1 \\ f_yL^y_0 + c_xL^x_1 + c_yL^y_1 + L^z_1 \\ f_xL^x_0 \\ c_xL^x_0 + c_yL^y_0 + L^z_0 \end{bmatrix}^\top \begin{bmatrix} v^3 \\ v^2u \\ v^2 \\ vu \\ v \\ u \\ 1 \end{bmatrix} = \begin{bmatrix} {}^{Unif}F_1 \\ {}^{Unif}F_2 \\ {}^{Unif}F_3 \\ {}^{Unif}F_4 \\ {}^{Unif}F_5 \\ {}^{Unif}F_6 \\ {}^{Unif}F_7 \end{bmatrix}^\top \begin{bmatrix} v^3 \\ v^2u \\ v^2 \\ vu \\ v \\ u \\ 1 \end{bmatrix} = 0 \tag{3.9}$$

where Eq. (3.9) is equivalent to Eq. (3.5). If we force the translational velocity $\mathbf{d}$ and rotational velocity $\boldsymbol{\omega}$ to be equal to zero, Eq. (3.9) will collapse into Eq. (3.3) as a straight line.

3.3.4 3D Line Projection with Other RS Instantaneous-motion Models

From the uniform model in Eq. (3.5), one can derive two simpler models: linear RS model and rotate-only model which assumes pure translation and pure rotation during image acquisition. By either forcing the linear velocity $\mathbf{d}$ or the angular velocity $\boldsymbol{\omega}$ to be equal to $\mathbf{0}$ respectively, one can obtain a hyperbolic curve. The parameterizations of 3D line projection with different RS models are summarized in table. 3.1 and Fig. 3.4.

3.3.4.1 3D Line Projection with linear RS model

By forcing the rotational velocity $\boldsymbol{\omega} = \mathbf{0}$, Eq. (3.5) becomes:

TABLE 3.1: Parametric representation of 3D straight line projection with different RS models

Camera model	Projection equation	Curve type	Parameters
GS Eq. (3.3)	${}^{GS}\mathrm{F}_1 u +^{GS} \mathrm{F}_2 v +^{GS} \mathrm{F}_3 = 0$	Straight line	**R**,**t**
Linear RS Eq. (3.10) Proof in section 3.3.4.1	${}^{Lin}F_1 v^2 +^{Lin} F_2 vu +^{Lin} F_3 v +$ ${}^{Lin}F_4 u +^{Lin} F_5 = 0$	Hyperbolic curve	**R**,**t**,**d**
Rotate-only RS Eq. (3.14) Proof in section 3.3.4.2	${}^{Rot}F_1 v^2 +^{Rot} F_2 vu +^{Rot} F_3 v^{Rot} +$ ${}^{Rot}F_4 u +^{Rot} F_5 = 0$	Hyperbolic curve	**R**,**t**,$\boldsymbol{\omega}$
Uniform RS Eq. (3.5) Proof in section 3.3.3	${}^{Unif}F_1 v^3 +^{Unif} F_2 v^2 u +$ ${}^{Unif}F_3 v^2 +^{Unif} F_4 vu +$ ${}^{Unif}F_5 v +^{Unif} F_6 u +^{Unif} F_7 = 0$	Cubic curve	**R**,**t**,**d**,$\boldsymbol{\omega}$

$$
{}^{Lin}F_1 v^2 +^{Lin} F_2 vu +^{Lin} F_3 v +^{Lin} F_4 u +^{Lin} F_5 = 0 \tag{3.10}
$$

where the five coefficients are determined by **K**, 3D line parameters, camera pose and kinematic parameters (**d**).

Derivation of Eq. (3.10). Distinctively, if we assume that the angular velocity $\boldsymbol{\omega}$ is equal to zero. Eq. (3.4) becomes:

$$
\begin{aligned}
\mathbf{R}_c &= \mathbf{R}_w^{c\top} \mathbf{R}_w \\
\mathbf{t}_c &= \begin{bmatrix} t_x & t_y & t_z \end{bmatrix}^\top = (\mathbf{R}_w)^\top (\mathbf{t}_c^w + \mathbf{d}v) \\
(a_c, b_c) &= \left(a_w - t_x, b_w - t_y\right)
\end{aligned} \tag{3.11}
$$

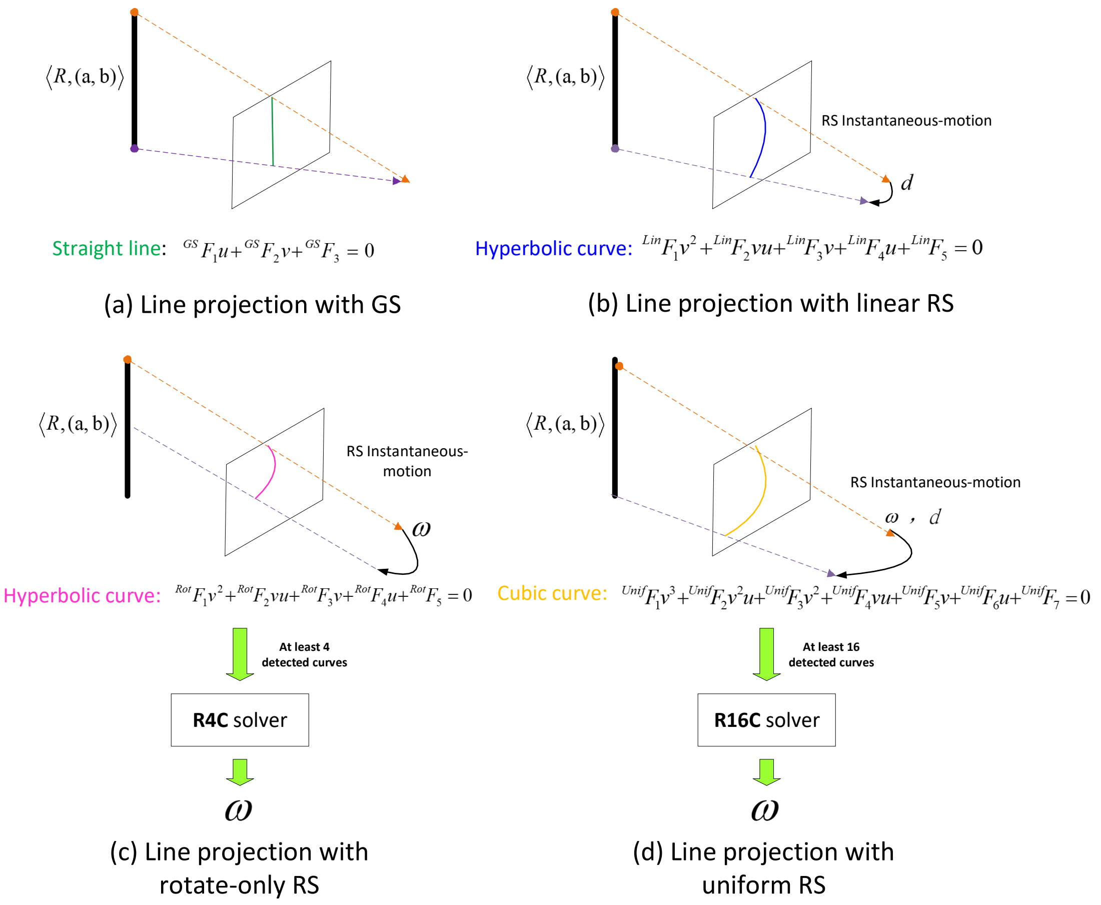
Thus, the straight line director vector $\mathbf{m}_{cip}$ can be expressed as follows:

$$
\begin{aligned}
&\begin{bmatrix} m_{cip}^x \\ m_{cip}^y \\ m_{cip}^z \end{bmatrix} = \mathbf{L} \begin{bmatrix} v \\ 1 \end{bmatrix} = \begin{bmatrix} L_1^x & L_0^x \\ L_1^y & L_0^y \\ L_1^z & L_0^z \end{bmatrix} \begin{bmatrix} v \\ 1 \end{bmatrix} \\
\text{with} \quad \mathbf{L} = &\begin{bmatrix} (\mathbf{R}_{w2}^\top R_{c22} - \mathbf{R}_{w1}^\top R_{c12})\mathbf{d} & (a_w - \mathbf{R}_{w1}^\top \mathbf{t}_c^w) R_{c12} - (b_w - \mathbf{R}_{w2}^\top \mathbf{t}_c^w) R_{c11} \\ (\mathbf{R}_{w2}^\top R_{c21} - \mathbf{R}_{w1}^\top R_{c22})\mathbf{d} & (a_w - \mathbf{R}_{w1}^\top \mathbf{t}_c^w) R_{c22} - (b_w - \mathbf{R}_{w2}^\top \mathbf{t}_c^w) R_{c21} \\ (\mathbf{R}_{w2}^\top R_{c31} - \mathbf{R}_{w1}^\top R_{c32})\mathbf{d} & (a_w - \mathbf{R}_{w1}^\top \mathbf{t}_c^w) R_{c32} - (b_w - \mathbf{R}_{w2}^\top \mathbf{t}_c^w) R_{c31} \end{bmatrix}
\end{aligned} \tag{3.12}
$$

where the 3×2 auxiliary matrix **L** is consisted by six variables L_0^x, L_1^x, L_0^y, L_1^y, L_0^z and L_1^z. Then we substitute Eq. (3.12) into Eq. (3.3), we obtain a hyperbolic curve as follows:

$$
\begin{bmatrix} f_y L_1^y \\ f_x L_1^x \\ f_y L_0^y + c_x L_1^x + c_y L_1^y + L_1^z \\ f_x L_0^x \\ f_x L_0^x + c_y L_0^y + L_0^z \end{bmatrix}^\top \begin{bmatrix} v^2 \\ vu \\ v \\ u \\ 1 \end{bmatrix} = \begin{bmatrix} {}^{Lin}F_1 \\ {}^{Lin}F_2 \\ {}^{Lin}F_3 \\ {}^{Lin}F_4 \\ {}^{Lin}F_5 \end{bmatrix}^\top \begin{bmatrix} v^2 \\ vu \\ v \\ u \\ 1 \end{bmatrix} = 0 \tag{3.13}
$$

where Eq. (3.13) is equivalent to Eq. (3.10). If we set the translational velocity $\mathbf{d} = \mathbf{0}$, Eq. (3.13) will collapse into Eq. (3.3) as a straight line.


$\langle R,(\mathrm{a},\mathrm{b})\rangle$
Straight line: ${}^{GS}F_1u+{}^{GS}F_2v+{}^{GS}F_3=0$
$\langle R,(\mathrm{a},\mathrm{b})\rangle$
RS Instantaneous-motion
d
Hyperbolic curve: ${}^{Lin}F_1v^2+{}^{Lin}F_2vu+{}^{Lin}F_3v+{}^{Lin}F_4u+{}^{Lin}F_5=0$
$\langle R,(\mathrm{a},\mathrm{b})\rangle$
RS Instantaneous-motion
ω
Hyperbolic curve: ${}^{Rot}F_1v^2+{}^{Rot}F_2vu+{}^{Rot}F_3v+{}^{Rot}F_4u+{}^{Rot}F_5=0$
At least 4 detected curves
R4C solver
ω
$\langle R,(\mathrm{a},\mathrm{b})\rangle$
RS Instantaneous-motion
ω , d
Cubic curve: ${}^{Unif}F_1v^3+{}^{Unif}F_2v^2u+{}^{Unif}F_3v^2+{}^{Unif}F_4vu+{}^{Unif}F_5v+{}^{Unif}F_6u+{}^{Unif}F_7=0$
At least 16 detected curves
R16C solver
ω

(a) Line projection with GS

(b) Line projection with linear RS

(c) Line projection with rotate-only RS

(d) Line projection with uniform RS

FIGURE 3.4: 3D line projection with different RS models.

3.3.4.2 3D Line Projection with pure rotation model

By setting the rotational velocity $\mathbf{d} = \mathbf{0}$, Eq. (3.5) becomes:

$$ {}^{Rot}F_1v^2 +^{Rot} F_2vu +^{Rot} F_3v +^{Rot} F_4u +^{Rot} F_5 = 0 \tag{3.14}$$

where the five coefficients are determined by $\mathbf{K}$, 3D line parameters, camera pose and kinematic parameters ($\mathbf{d}$).

Derivation of Eq. (3.14). Distinctively, if we assume that the angular velocity $\mathbf{d}$ is equal to zero. Eq. (3.4) becomes:

$$\begin{aligned} \mathbf{R}_c &= ((\mathbf{I} + [\boldsymbol{\omega}]_\times v])\mathbf{R}_w^c))^\top \mathbf{R}_w \\ \mathbf{t}_c &= \begin{bmatrix} t_x & t_y & t_z \end{bmatrix}^\top = (\mathbf{R}_w)^\top \mathbf{t}_c^w \\ (a_c, b_c) &= \left(a_w - t_x, b_w - t_y\right) \end{aligned} \tag{3.15}$$

Thus, the straight line director vector $\mathbf{m}_{cip}$ can be expressed as follows:

$$\begin{bmatrix} m^x_{cip} \\ m^y_{cip} \\ m^z_{cip} \end{bmatrix} = \mathbf{L}\begin{bmatrix} v \\ 1 \end{bmatrix} = \begin{bmatrix} L^x_1 & L^x_0 \\ L^y_1 & L^y_0 \\ L^z_1 & L^z_0 \end{bmatrix}\begin{bmatrix} v \\ 1 \end{bmatrix}$$
$$\text{with} \quad \mathbf{L} = \begin{bmatrix} a_cB_{12} - b_cB_{11} & a_cA_{12} - b_cA_{11} \\ a_cB_{22} - b_cB_{21} & a_cA_{22} - b_cA_{21} \\ a_cB_{22} - b_cB_{31} & a_cA_{32} - b_cA_{31} \end{bmatrix} \text{ and } \begin{cases} \mathbf{A} = \mathbf{R}^w_c\mathbf{R}_w \\ \mathbf{B} = \mathbf{R}^w_c[\boldsymbol{\omega}]_\times\mathbf{R}_w \end{cases} \tag{3.16}$$

where $\mathbf{A}$ and $\mathbf{B}$ are two 3×3 auxiliary matrices defined by $\mathbf{R}_c = \mathbf{A} + \mathbf{B}v$. While the 3×2 auxiliary matrix $\mathbf{L}$ is consisted by six variables L^x_0, L^x_1, L^y_0, L^y_1, L^z_0 and L^z_1. Then we substitute Eq. (3.16) into Eq. (3.3), we obtain a hyperbolic curve as follows:

$$\begin{bmatrix} f_yL^y_1 \\ f_xL^x_1 \\ f_yL^y_0 + c_xL^x_1 + c_yL^y_1 + L^z_1 \\ f_xL^x_0 \\ f_xL^x_0 + c_yL^y_0 + L^z_0 \end{bmatrix}^\top \begin{bmatrix} v^2 \\ vu \\ v \\ u \\ 1 \end{bmatrix} = \begin{bmatrix} {}^{Rot}F_1 \\ {}^{Rot}F_2 \\ {}^{Rot}F_3 \\ {}^{Rot}F_4 \\ {}^{Rot}F_5 \end{bmatrix}^\top \begin{bmatrix} v^2 \\ vu \\ v \\ u \\ 1 \end{bmatrix} = 0 \tag{3.17}$$

where Eq. (3.17) is equivalent to Eq. (3.14). If we force the translational velocity $\boldsymbol{\omega} = \mathbf{0}$, Eq. (3.17) will collapse into Eq. (3.3) as a straight line.

3.4 Instantaneous-motion from 2D Curves

In this section, we show how to estimate the instantaneous-motion of the camera by using the coefficients of 2D curves.

3.4.1 Linear 16-curves Solution for Uniform RS Model

We introduce a 16-curves linear solution called **R16C** to estimate instantaneous-rotation basing on the RS uniform instantaneous-motion model.

For a single RS image, if we assume the camera frame as world coordinate system, hence, $\mathbf{R}^\mathbf{c}_\mathbf{w} = \mathbf{I}$ and $\mathbf{t}^\mathbf{c}_\mathbf{w} = \mathbf{0}$. Then, based on Eq. (3.5), $\boldsymbol{\omega}$ can be expressed according to the seven coefficients of the cubic curves and intrinsic parameters in $\mathbf{K}$:

$$\begin{bmatrix} C_1 & \cdots & C_{17} \end{bmatrix} \begin{bmatrix} W_1 & \cdots & W_{17} \end{bmatrix}^T = 0 \tag{3.18}$$

where C_i are 17 known auxiliary variables determined by $\mathbf{K}$ and cubic curve coefficients ${}^{unif}\mathrm{F}_1$ to ${}^{unif}\mathrm{F}_7$. While $W_i, i = 1, \ldots, 17$ are 17 unknown variables consisted by components of $\boldsymbol{\omega}$. Finally, this equation can be solved linearly by SVD with at least 16 detected curves which correspond to 3D straight segments.

Derivation of Eq. (3.18). Basing on Eq. (3.9). We first denote 3D line structural parameters as $a_c\mathbf{R_{w2}} - b_c\mathbf{R_{w1}} = [s_1, s_2, s_3]^T$, and assume $D^a\mathbf{R_{w2}} - D^b\mathbf{R_{w1}} = [h_1, h_2, h_3]^T$. Thus, the 3×3 auxiliary matrix $\mathbf{L}$ consisted by nine variables L^x_0, L^x_1, L^x_2, L^y_0, L^y_1, L^y_2, L^z_0, L^z_1 and L^z_2 in Eq. (3.8) can also be denoted as follows:

$$\mathbf{L} = \begin{bmatrix} L^x_2 & L^x_1 & L^x_0 \\ L^y_2 & L^y_1 & L^y_0 \\ L^z_2 & L^z_1 & L^z_0 \end{bmatrix} = \begin{bmatrix} \omega_zh_2 - \omega_yh_3 & s_3\omega_y - s_2\omega_z - h_1 & s_1 \\ \omega_xh_3 - \omega_zh_1 & s_1\omega_z - s_3\omega_x - h_2 & s_2 \\ \omega_yh_1 - \omega_xh_2 & s_2\omega_x - s_1\omega_y - h_3 & s_3 \end{bmatrix} \tag{3.19}$$

For readability, we denote F^{Uni}_1, F^{Uni}_2, F^{Uni}_3, F^{Uni}_4, F^{Uni}_5, F^{Uni}_6 and F^{Uni}_7 as F_1, F_2, F_3, F_4, F_5, F_6 and F_7 respectively. Thus, for each fitted cubic curve, we obtain a polynomial group with 7 equations:

$$\begin{cases} F_1 = f_y(\omega_x h_3 - \omega_z h_1) \\ F_2 = f_x(\omega_z h_2 - \omega_y h_3) \\ F_3 = f_y(s_1\omega_z - s_3\omega_x - h_2) + c_x\frac{F_2}{f_x} + c_y\frac{F_1}{f_y} + \omega_y h_1 - \omega_x h_2 \\ F_4 = f_x(s_3\omega_y - s_2\omega_z - h_1) \\ F_5 = f_y s_2 + c_x\frac{F_4}{f_x} + c_y(s_1\omega_z - s_3\omega_x - h_2) + s_2\omega_x - s_1\omega_y - h_3 \\ F_6 = f_x s_1 \\ F_7 = c_x s_1 + c_y s_2 + s_3 \end{cases} \tag{3.20}$$

The objective is to eliminate h_1, h_2, h_3, s_1, s_2 and s_3 from the above 7 equations.

we Basing on Eq. (3.3), h_1, h_2 and h_3, which contain translational velocity $\mathbf{d}$, can be expressed by using $\boldsymbol{\omega}$ and s_1, s_2, s_3 and curve coefficients as follows:

(1) Base on the definition of $F_4 = f_x(s_3\omega_y - s_2\omega_z - h_1)$, we obtain the expression of h_1 as:

$$h_1 = s_3\omega_y - s_2\omega_z - F_4/f_x \tag{3.21}$$

(2) Base on the definition of $F_1 = f_y(\omega_x h_3 - \omega_z h_1)$ and Eq. (3.21), we the expression of h_3 as:

$$h_3 = \frac{F_1}{f_y\omega_x} + \frac{\omega_z}{\omega_x}h_1 = \frac{F_1}{\omega_x f_y} + \frac{\omega_z(s_3\omega_y - s_1\omega_z) - F_4/f_x}{\omega_x} \tag{3.22}$$

(3) Base on the definition of $F_2 = f_x(\omega_z h_2 - \omega_y h_3)$ and Eq. (3.21) and (3.22), we the expression of h_2 as:

$$\begin{aligned} h_2 &= \frac{F_2}{f_x\omega_z} + \frac{\omega_y}{\omega_z}h_3 = \frac{F_2}{f_x\omega_z} + \frac{F_1\omega_y}{f_y\omega_x\omega_z} + \frac{\omega_y}{\omega_x}h_1 \\ &= \frac{\frac{F_2}{f_x} + \omega_y\frac{F_1}{\omega_x f_y} + \frac{\omega_z(s_3\omega_y - s_1\omega_z) - F_4/f_x}{\omega_x}}{\omega_z} \end{aligned} \tag{3.23}$$

Then we introduce two new auxiliary variables a and b:

(1) Basing on the definition of $F_3 = f_y(s_1\omega_z - s_3\omega_x - h_2) + c_x\frac{F_2}{f_x} + c_y\frac{F_1}{f_y} + \omega_y h_1 - \omega_x h_2$ from Eq. (3.20), we assume:

$$a = F_3 - c_x F_2/f_x - c_y F_1/f_y = f_y(s_1\omega_z - s_3\omega_x - h_2) + (\omega_y h_1 - \omega_x h_2) \tag{3.24}$$

(2) Basing on the definition of $F_5 = f_y s_2 + c_x\frac{F_4}{f_x} + c_y(s_1\omega_z - s_3\omega_x - h_2) + s_2\omega_x - s_1\omega_y - h_3$ from Eq. (3.20), we assume:

$$b = F_5 - c_x F_4/f_x = f_y s_2 + c_y(s_1\omega_z - s_3\omega_x - h_2) + s_2\omega_x - s_1\omega_y - h_3 \tag{3.25}$$

Now, we consider F_6 and F_7 in Eq. Eq. (3.20):

(1) From the definition of $F_6 = f_x s_1$, we obtain the expression of d_1:

$$s_1 = \frac{F_6}{f_x} \tag{3.26}$$

(2) From the definition of $F_7 = c_x s_1 + c_y s_2 + s_3$, we obtain the relation between s_2 and s_3 as:

$$s_3 = F_7 - \frac{c_x F_6}{f_x} - c_y s_2 \tag{3.27}$$

Then we substitute h_1 from Eq. (3.21), h_2 from Eq. (3.23), h_3 from Eq. (3.22) and Eq. (3.27) into Eq. (3.24) to obtain the expression of s_3:

$$s_3 = \frac{\frac{f_y F_6}{f_x}\omega_z - \frac{f_y F_2}{f_x \omega_z} - \frac{f_y F_1 \omega_y}{f_x \omega_x \omega_z} + \frac{F_7 f_y \omega_y}{\omega_x{}^2} - \frac{f_y c_x F_6 \omega_y}{f_x c_y \omega_x} - \frac{f_y F_4 \omega_y}{f_x \omega_x} - \frac{F_2 \omega_x}{f_x \omega_z} - \frac{F_1 \omega_y}{f_x \omega_z} - a}{f_y \omega_x + \frac{f_y \omega_y{}^2}{\omega_x} + \frac{f_y \omega_y}{c_y \omega_x}} \tag{3.28}$$

This time, By substituting s_3, s_2 calculated by Eq. (3.27) and Eq. (3.27), h_1 from Eq. (3.21), h_2 from Eq. (3.23), h_3 from Eq. (3.22) into Eq. (3.25), we will obtain a pronominal which all the terms are consisted by $\boldsymbol{\omega}$ while all the coefficients are determined by $\mathbf{K}$ and cubic curve coefficients F_1 to F_7 as follows:

$$\begin{bmatrix} C_1 & \cdots & C_{17} \end{bmatrix} \begin{bmatrix} W_1 & \cdots & W_{17} \end{bmatrix}^T = \begin{bmatrix} J_1 \\ J_7 + G_7 \\ J_2 \\ J_3 + J_6 - G_3 \\ G_9 \\ J_1 \\ J_5 - G_4 \\ J_4 - G_6 \\ G_2 \\ G_1 \\ G_5 \\ J_3 \\ J_3 + J_7 \\ -G_8 \\ J_4 \\ J_5 \\ J_6 \end{bmatrix}^\top \begin{bmatrix} \omega_x^4 \\ \omega_x^3 \\ \omega_x^3\omega_y \\ \omega_x^3\omega_z \\ \omega_x^2\omega_y\omega_z^2 \\ \omega_x^2\omega_y^2 \\ \omega_x^2\omega_y\omega_z \\ \omega_x^2 \\ \omega_x^2\omega_z^2 \\ \omega_x^2\omega_z \\ \omega_x^2\omega_y \\ \omega_x\omega_y^3 \\ \omega_x\omega_y^2\omega_y \\ \omega_x\omega_y\omega_z^2 \\ \omega_y^2 \\ \omega_y^2\omega_z \\ \omega_y^4\omega_z \end{bmatrix} = 0 \tag{3.29}$$

$$\text{with} \begin{cases} J_1 = \frac{F_6/f_x c_x + F_7}{c_y^2} \\ J_2 = \frac{F_6/f_x}{c_y} \\ J_3 = \frac{-c_x F_7 - F_6/f_x c_x^2}{c_y^2} \\ J_4 = \frac{F_1}{f_x c_y} + \frac{F_4}{f_x c_y} \\ J_5 = \frac{-f_6 c_x}{f_x c_y} \\ J_6 = \frac{-f_6 c_x}{f_x c_y} \\ J_7 = \frac{b}{c_y} - \frac{f_y F_7 - c_x f_y F_6/f_x}{c_y^2} \end{cases} \quad \text{and} \quad \begin{cases} G_1 = -(a - F_6/f_x) \\ G_2 = \frac{c_x a}{f_y} \\ G_3 = \frac{a}{f_y} \\ G_4 = \frac{F_6}{f_x f_y} \\ G_5 = \frac{F_4}{f_x f_y} \\ G_6 = \frac{F_2}{f_x} \\ G_7 = \frac{F_2}{f_x f_y} \\ G_8 = \frac{F_6 F_4}{f_x^2} \\ G_9 = \frac{F_6 F_4}{f_x^2 f_y} \end{cases}$$

where $J_1, \ldots, J_7$ and $G_1 \ldots G_9$ are auxiliary variables which can be computed by calibration matrix $\mathbf{K}$ and cubic curve coefficients ${}^{unif}\mathrm{F}_1$ to ${}^{unif}\mathrm{F}_7$.

Basing on Eq. (3.29), we can use non-linear optimization to estimate ω_x, ω_y and ω_z. In such case, at least 3 fitted curves are needed. Alternatively, by giving 16 curves, Eq. (3.29) can also be solved linearly using SVD.

3.4.2 Comparison of the Three RS Models

Some existing works argued that only angular-velocity plays a main role for hand-held and vehicle devices [Ringaby and Forssén, 2012, Duchamp et al., 2015, Rengarajan et al.,

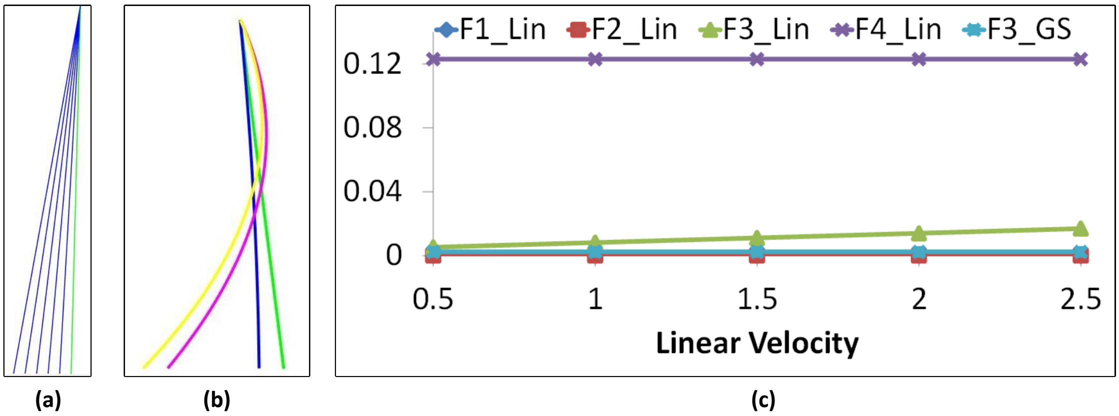


FIGURE 3.5: Projections of a 3D straight lines with different RS camera kinematics. (a) A simulated 3D straight line projected onto a 2D image as different forms of curves with no instantaneous-motion (green), linear (blue), rotate-only (pink) and uniform instantaneous-motion (yellow). Assuming the depth from the 3D straight line to camera as 1 unit length. Blue curves in (b) are the projection of a 3D line into a linear RS camera with linear velocities from 0.5 to 2.5 unit/s, while green line is for the GS case. The variations of ${}^{Lin}F_1$, ${}^{Lin}F_2$, ${}^{Lin}F_3$ and ${}^{Lin}F_4$ and constant value of ${}^{GS}F_3$ are shown in (c).

2016, Purkait et al., 2017, Ito and Okatani,]. Here, we give a further quantitative analysis of both rotational and translational instantaneous-motion effects on 3D line projection. Although the linear RS model will introduce a hyperbolic curve, however, its second order coefficients ${}^{Lin}F_1 = \mathbf{K}_{22}^{-\top}(\mathbf{R}_{w21}R_{w2}^{\top} - \mathbf{R}_{w22}R_{w1}^{\top})\mathbf{d}$, ${}^{Lin}F_2 = \mathbf{K}_{11}^{-\top}(\mathbf{R}_{w11}\mathbf{R}_{w2}^{\top} - \mathbf{R}_{w12}R_{w1}^{\top})\mathbf{d}$ are much smaller compared to ${}^{Lin}F_3 = \mathbf{K}_{22}^{-\top}(a_w\mathbf{R}_{w22} - b_w\mathbf{R}_{w21}) + \frac{\mathbf{K}_{31}^{-\top}}{\mathbf{K}_{11}^{-\top}}{}^{Lin}F_2 + \frac{\mathbf{K}_{32}^{-\top}}{\mathbf{K}_{22}^{-\top}}{}^{Lin}F_1 + (\mathbf{R}_{w31}\mathbf{R}_{w2}^{\top} - \mathbf{R}_{w32}\mathbf{R}_{w1}^{\top})\mathbf{d}$ and ${}^{Lin}F_4 = \mathbf{K}_{11}^{-\top}(a_w\mathbf{R}_{w12} - b_w\mathbf{R}_{w11})$ and can be ignored in practice. The simulated data experiment shown in Fig. 3.5 confirmed that even with a high linear speed, ${}^{Lin}F_1$, ${}^{Lin}F_2$ are relatively low, and projected curves (blue) are close to straight lines as for GS case (green).

In practice, the effect of translational speed can be compensated by an increment on the rotational speed. Therefore, we chose to extract the angular velocity basing on the rotate-only RS model instead of the uniform model. This assumption holds because the translation during frame exposure is negligible in comparison to the depth of the features to be used. Doing so, it becomes possible to compensate rolling shutter effects of the hole image independently from the depth associated to each pixel. This is the key of the single-view based RS correction.

However, the **R16C** still can be used in very specific application where the translational speed is very high in comparison to scan speed, and where curve detection can be achieved with a very high accuracy (subpixellic).

3.4.3 Practical Linear 4-Curves Algorithm

Now, we introduce a 4-curves linear solution called **R4C** to estimate instantaneous-rotation basing on the RS rotate-only model.

For a single RS image, if we assume the camera frame as world coordinate system, hence, $\mathbf{R}_{\mathbf{w}}^{\mathbf{c}} = \mathbf{I}$ and $\mathbf{t}_{\mathbf{w}}^{\mathbf{c}} = \mathbf{0}$. Then we denote the 3D line structural parameters as

$a_c\mathbf{R}_{w2} - b_c\mathbf{R}_{w1} = [s_1, s_2, s_3]^{\top}$, for five hyperbolic coefficients of each curve, we can formulate a group of equations:

$$\begin{aligned}
F_1 &= f_y(s_1\omega_z - s_3\omega_x) \\
F_2 &= f_x(s_3\omega_y - s_2\omega_z) \\
F_3 &= f_y s_2 + c_x(s_3\omega_y - s_2\omega_z) + c_y(s_1\omega_z - s_3\omega_x) + (s_2\omega_x - s_1\omega_y) \\
F_4 &= f_x s_1 \\
F_5 &= f_x s_1 + c_y s_2 + s_3
\end{aligned} \tag{3.30}$$

where s_1, s_2 and s_3 are different for each curve.

From Eq. (3.30), s_1, s_2, s_3 and ω_z can be substituted by ω_x and ω_y. We can obtain a bivariate cubic polynomial. With new coefficients C_1 to C_8 which are only determined by matrix $\mathbf{K}$ and coefficients F_1 to F_5. Now, by giving four curves, we have:

$$\begin{vmatrix} C_1^1 & \cdots & C_8^1 \\ \vdots & \ddots & \vdots \\ C_1^4 & \cdots & C_8^4 \end{vmatrix} [{\omega_x}^3, {\omega_y}^2\omega_x, {\omega_x}^2, {\omega_y}^2, \omega_x\omega_y, \omega_x, \omega_y, 1]^\top = 0 \tag{3.31}$$

By eliminating ω_x^3 and $\omega_y^2\omega_x$, Eq. (3.31) becomes a two bi-variables quadratic polynomial equations:

$$\begin{vmatrix} T_1^1 & \cdots & T_6^1 \\ T_1^2 & \cdots & T_6^2 \end{vmatrix} [{\omega_x}^2, {\omega_y}^2, \omega_x\omega_y, \omega_x, \omega_y, 1]^\top = 0 \tag{3.32}$$

Where coefficients T are calculated by coefficients C in Eq. 3.31. Again, we further substitute ω_y by ω_x and the 10 coefficients T in Eq. (3.32), then we obtain:

$$(H_1, H_2, H_3, H_4, H_5)({\omega_x}^4, {\omega_x}^3, {\omega_x}^2, \omega_x, 1)^\top = 0 \tag{3.33}$$

Thus, Eq. 3.32 turns into a bi-quadratic polynomial equation with one unknown (ω_x). The five variables $H_1, \dots, H_5$ can be computed by using $F_1, \dots, F_5$ and components of $\mathbf{K}$.

If more than 4 curves are available, parameter ω_x can be recovered by solving the non homogeneous linear system Eq. (3.33), after which ω_y will be recovered using Eq. (3.31). Then we calculate ω_z based on Eq. (3.30).

3.4.3.1 Derivation of Eq. (3.33)

Transformation from Eq. (3.30) to Eq. (3.31). In order to extract angular velocity from Eq. (3.30), we can first substitute structure unknowns s_1, s_2 and s_3 just by angular velocity parameters ω_x, ω_y and ω_z. Thus, we define four auxiliary variables a, b, c and d as follows:

$$\begin{aligned}
a &= F_5 - F_4 = c_y s_2 + s_3 \\
b &= \frac{F_2}{f_x} = s_3\omega_y - \frac{a - s_3}{c_y}\omega_z \\
c &= \frac{F_1}{f_y} = \frac{F_4}{f_x}\omega_z - s_3\omega_x \\
d &= F_3 - \frac{c_y}{f_y}F_1 - \frac{c_x}{f_x}F_2 = (f_y + \omega_x)\frac{a - s_3}{c_y} - \frac{F_4}{f_x}\omega_y
\end{aligned} \tag{3.34}$$

then we substitute ω_z and s_3 by ω_x and ω_y, then we obtain:

$$c_y[(d - \frac{af_y}{c_y} + \frac{a}{c_y})\omega_x - \frac{f_y F_4}{f_x}\omega_y + \frac{F_4}{f_x}\omega_x\omega_y + \frac{a}{c_y}\omega_x^2 + (\frac{af_y}{c_y} - f_y d)][\omega_y - \frac{af_x}{F_4 c_y}c] = 0 \tag{3.35}$$

Eq. (3.35) can be re-written as Eq. (3.31)'s form, which is a cubic bi-varibales polynomial:

$$\begin{aligned}
&C_1{\omega_x}^3 + C_2{\omega_y}^2\omega_x + C_3{\omega_x}^2 + C_4{\omega_y}^2 + C_5\omega_x\omega_y + C_6\omega_x + C_7\omega_y + C_8 = 0 \\
\text{with}\quad &C_1 = -\frac{a^2 f_x f_y}{F_4 c_y} \\
&C_2 = -\frac{c_y F_4}{f_x} \\
&C_3 = (-\frac{a f_x}{F_4})(d - \frac{a f_y}{c_y} + \frac{a}{c_y}) + \frac{f_x ac}{F_4 c_y} + a^2 - b - \frac{f_x}{F_4 c_y}ac \\
&C_4 = -\frac{f_y F_4 c_y}{f_x} + \frac{F_4^2 c_y^2}{f_x^2} \\
&C_5 = dc_y - af_x + a + f_y a + c + 2aF_4 c_y \\
&C_6 = \frac{a f_y}{F_4}(f_y d - a\frac{f_y}{c_y}) + \frac{f_x c}{F_4}(d - a\frac{f_y}{c_y} + \frac{a}{c_y}) + 2adc_y - 2a^2 f_y(b + \frac{f_x}{F_4 c_y}ac) \\
&C_7 = (af_y - f_y c_y d) - f_y c + 2\frac{dF_4 c_y^2}{f_x} - 2\frac{ac_y F_4 f_y}{f_x} \\
&C_8 = \frac{c f_x}{F_4}(\frac{a f_y}{c_y} - f_y d) + c_y^2 d^2 - 2ac_y f_y d + a^2 f_y^2 - f_y^2(b + \frac{f_x}{F_4 c_y}ac)
\end{aligned} \tag{3.36}$$

Transformation from Eq. (3.31) to Eq. (3.32). We first re-write the equation system of four curves basing on Eq. (3.36):

$$C_1^{(1)}{\omega_x}^3 + C_2^{(1)}{\omega_y}^2\omega_x + C_3^{(1)}{\omega_x}^2 + C_4^{(1)}{\omega_y}^2 + C_5^{(1)}\omega_x\omega_y + C_6^{(1)}\omega_x + C_7^{(1)}\omega_y + C_8^{(1)} = 0 \tag{3.37}$$

$$C_1^{(2)}{\omega_x}^3 + C_2^{(2)}{\omega_y}^2\omega_x + C_3^{(2)}{\omega_x}^2 + C_4^{(2)}{\omega_y}^2 + C_5^{(2)}\omega_x\omega_y + C_6^{(2)}\omega_x + C_7^{(2)}\omega_y + C_8^{(2)} = 0 \tag{3.38}$$

$$C_1^{(3)}{\omega_x}^3 + C_2^{(3)}{\omega_y}^2\omega_x + C_3^{(3)}{\omega_x}^2 + C_4^{(3)}{\omega_y}^2 + C_5^{(3)}\omega_x\omega_y + C_6^{(3)}\omega_x + C_7^{(3)}\omega_y + C_8^{(3)} = 0 \tag{3.39}$$

$$C_1^{(4)}{\omega_x}^3 + C_2^{(4)}{\omega_y}^2\omega_x + C_3^{(4)}{\omega_x}^2 + C_4^{(4)}{\omega_y}^2 + C_5^{(4)}\omega_x\omega_y + C_6^{(4)}\omega_x + C_7^{(4)}\omega_y + C_8^{(4)} = 0 \tag{3.40}$$

where $C_1^{(i)}$ is the C_1 coefficient of the i^{th} curve. Now, we define three auxiliary variables r^1, r^2 and r^3 as follows:

$$r^1 = \frac{C_1^1}{C_1^2}, \qquad r^2 = \frac{C_1^1}{C_1^3} \qquad \text{and} \qquad r^3 = \frac{C_3^1 - r^1 C_3^2}{C_3^3 - r^2 C_3^2} \tag{3.41}$$

we can substitute ω_y by ω_x by using Eq. (3.37), (3.38), (3.39) and the three auxiliary variables above as follows:

$$Eq.(3.37) - r^1 Eq.(3.38) - r^3(Eq.(3.39) - r^2 Eq.(3.38)) = 0 \tag{3.42}$$

which is equivalent to the first row of Eq. (3.32) as:

$$T_1^1\omega_x^2 + T_2^1\omega_y^2 + T_3^1\omega_x\omega_y + T_4^1\omega_x + T_5^1\omega_y + T_6^1 = 0 \tag{3.43}$$

where coefficients $T_1^1, \ldots, T_5^1$ can be computed by $C_1^1, \ldots, C_8^3$. Then we use Eq. (3.40) to replace Eq. (3.39) and under the same transformation Eq. (3.42). We will obtain the second row of Eq. (3.32) as:

$$T_1^2\omega_x^2 + T_2^2\omega_y^2 + T_3^2\omega_x\omega_y + T_4^2\omega_x + T_5^2\omega_y + T_6^2 = 0 \tag{3.44}$$

Transformation from Eq. (3.32) to Eq. (3.33). At this stage, ω_y can be substituted by ω_x using Eq. (3.43) and Eq. (3.44) and leads to Eq. (3.33) as:

$$(H_1, H_2, H_3, H_4, H_5)({\omega_x}^4, {\omega_x}^3, {\omega_x}^2, \omega_x, 1)^\top = 0 \tag{3.45}$$

where coefficients H_1 to H_5 correspond to $[T_1^1, \ldots, T_6^1, T_1^2, \ldots, T_6^2]$. In such case, two bi-variables cubic polynomial equation turn into a quartic polynomial equation with one unknown. Finally,Eq. (3.33) can be solved directly with four geometric possible solutions, however, only one is correct. Therefore, we choose the most geometrically consistent value.

3.4.4 Degeneracies Analysis

Generally, instantaneous-motion estimation basing on projected curves (namely on their coefficients $F_1, \ldots, F_5$ of Eq. (3.14)) leads to a unique solution $\{< \mathbf{R}_w, (a_w, b_w)) >, \boldsymbol{\omega}\}$. Nevertheless, some degenerate or singular configurations leads to ambiguous results on $\boldsymbol{\omega}$. We present three degenerate configurations:

Case 1: 3D line located within y-z-plane ($a_w = 0, b_w = \forall x, \mathbf{R}_w = (\forall x, 0, 0)$) and arbitrary instantaneous-rotation along x-axis ($\boldsymbol{\omega} = (\forall x, 0, 0)$).

Case 2: 3D line located within x-z-plane ($a_w = \forall x, b_w = 0, \mathbf{R}_w = (0, \forall x, 0)$) and arbitrary instantaneous-rotation along y-axis ($\boldsymbol{\omega} = (0, \forall x, 0)$).

Case 3: 3D line parallel to x-axis ($a_w = \forall x$, $b_w = \forall x$, $\mathbf{R}_w = (0, \pi/2, 0)$) and arbitrary instantaneous-rotation along y-axis ($\boldsymbol{\omega} = (\forall x, 0, 0)$).

However, the configurations occur rarely in practice with hand-held camera or with a camera embedded on a vehicle.

3.4.4.1 Proof of the three degenerate cases

We present derivation details of the three degenerate cases of the linear 4-curves solution.

Proof. **Degenerate case 1** We assume a 3D line located within y-z-plane and a camera under an arbitrary instantaneous-rotation about x-axis. This leads to the following configuration:

$$\begin{cases} \mathfrak{L} =< \mathbf{R_w}, (a_w, b_w)) >= \{< \mathfrak{R}(\forall x_1, 0, 0), (0, \forall x_2) > \quad |x_1, x_2 \in \mathbb{R}\} \\ \qquad\qquad \boldsymbol{\omega} = \{[\forall x, 0, 0] \quad |x \in \mathbb{R}\} \end{cases} \tag{3.46}$$

where $\mathfrak{R}(a, b, c)$ is a rotation matrix generated by rotation angles a, b and c about x-y-z axis respectively. By substituting Eq. (3.46) into Eq. (3.17), and assuming $\mathbf{R}_C^W = \mathbf{I}$ and $\mathbf{t}_C^W = [0; 0; 0]$, Eq. (3.14) becomes:

$$\begin{aligned} F_1 &= 0 \\ F_2 &= 0 \\ F_3 &= f_y L_0^y = 0 \\ F_4 &= L_0^x \\ F_5 &= c_x L_0^x + L_0^z \end{aligned} \tag{3.47}$$

The equation above indicates that if an arbitrary 3D line within y-z-plane is observed by an RS camera under instantaneous-rotation about x-axis, no matter magnitude of speed, all of these lines will be projected as the same 2D line $u = c_x / f_x$. In other words, a projected curve can be explained by multiple configurations $\{< \mathbf{R_w}, (a_w, b_w)) >, \boldsymbol{\omega}\}$. Thus, configuration in Eq. (3.46) is a degenerate one.

Proof. **Degenerate case 2** This time, we assume a 3D line located within x-z-plane and camera under arbitrary instantaneous-rotation along y-axis during acquisition. This leads to:

$$\begin{cases} \mathfrak{L} =< \mathbf{R_w}, (a_w, b_w)) >= \{< \mathfrak{R}(0, \forall x_1, 0), (\forall x_2, 0) > \quad | x_1, x_2 \in \mathbb{R}\} \\ \qquad\qquad \boldsymbol{\omega} = \{[0, \forall x, 0] \quad | x \in \mathbb{R}\} \end{cases} \tag{3.48}$$

Thus, Eq. (3.14) becomes:

$$\begin{aligned} F_1 &= 0 \\ F_2 &= 0 \\ F_3 &= f_y L_0^y \\ F_4 &= f_x L_0^x = 0 \\ F_5 &= c_y L_0^y + L_0^z \end{aligned} \tag{3.49}$$

The equation above indicates that if an arbitrary 3D line within y-z-plane is observed by an RS camera under instantaneous-rotation about y-axis, no matter magnitude of speed, all of these lines will be projected as the 2D lines $v = c_y / f_x$. Therefore, the configuration in Eq. (3.48) is also a degenerate one.

Proof. **Degenerate case 3** We assume a 3D line parallel to x-axis and camera under arbitrary instantaneous-rotation about x-axis during acquisition. This leads to:

$$\begin{cases} \mathfrak{L} =< \mathbf{R_w}, (a_w, b_w)) >= \{< \mathfrak{R}(0, \pi/2, 0), (\forall x_1, \forall x_2) > \quad | x_1, x_2 \in \mathbb{R}\} \\ \qquad\qquad \boldsymbol{\omega} = \{[\forall x, 0, 0] \quad | x \in \mathbb{R}\} \end{cases} \tag{3.50}$$

Thus, Eq. (3.14) becomes:

$$\begin{cases} F_1 = -f_y b_w \omega_x \\ F_2 = 0 \\ F_3 = f_y L_0^y - c_y b_w \omega_x + a_w \omega_x \\ F_4 = 0 \\ F_5 = c_y L_0^y + L_0^z \end{cases} \tag{3.51}$$

The equation above indicates that if an arbitrary 3D line parallel to x-axis is observed by a RS camera under instantaneous-rotation about x-axis, no matter magnitude of speed, all of these lines will be projected as horizontal 2D lines in image as $F_1 v^2 + F_3 v + F_5 = 0$. Indeed, each of these lines can be explained by the coupling of a_w, b_w and $\boldsymbol{\omega}$. Therefore, the configuration in Eq.(3.50) is also a degenerate one.

3.5 RS Image Correction

In this section we show how to correct the RS image in this section, after successfully extracting the rotational velocity from the detected curves.

3.5.0.1 Compensating Instantaneous-motion

We compensate the effects of $\boldsymbol{\omega}$ in order to correct RS image by performing a forward mapping (eliminating $\mathbf{P}_i$ with Eq. (2.4), Eq. (2.11) and Eq. (2.20)) to all pixels as:

$$\mathbf{m}_i^{GS} = \Pi^{GS}(\mathbf{KR}(v_i)^{-1}\mathbf{K}^{-1}\mathbf{m}_i^{RS}) \tag{3.52}$$

where, the procedure will map original points $\mathbf{q}^{RS}$ to $\mathbf{q}^{GS}$ on global frame. $\mathbf{R}(v_i)$ is computed by using Eq. (2.20).

3.5.0.2 Automatic Selection of Actual 3D Lines

Since both 3D lines and curves will rendered as 2D curves on image, the problem of how to automatically distinguish image curves corresponding to 3D lines from actual 3D curves arises.

Method of [Rengarajan et al., 2016] uses 3 degree polynomial fitting to reject obvious outliers, but not ambiguous ones. The method of [Purkait et al., 2017] uses Huber M-estimator during joint estimation of motion parameters and vanishing directions to reject short line segments which are not sufficiently well oriented according to the vanishing directions. Unfortunately, some of these line segments survive the rejection process because they may be aligned with one of the vanishing directions despite not satisfying MW assumption, thus participating to motion parameter computation.

In this paper, we propose a 2-steps method (shown in Fig. 3.6) inspired from RANSAC technique. This selection process aims not only to reject features corresponding to non straight lines (outliers) but also to maximize the number of features corresponding to actual straight lines (inliers), thus increasing the accuracy and robustness of instantaneous-motion calculation.

Step 1: The goal of this step is to generate a preliminary set of candidate curves. Edge curves are first detected using Canny detector and linked to close small gaps. Then short curves (less than 20 pixels in experiments) are discarded. Finally, curves which sufficiently fit hyperbolic polynomial are selected basing on RMSE fitting error score.

Step 2: The second step is a global consistency verification. It consists in the integration of the R4C method within a RANSAC-like framework:

- Repeat N times

  (i) Select a random sample $\mathbb{S}_i = \{\mathbb{C}_1, \mathbb{C}_2, \mathbb{C}_3, \mathbb{C}_4\}$ of 4 four curves $\mathbb{C}_j$ among the candidate curves.

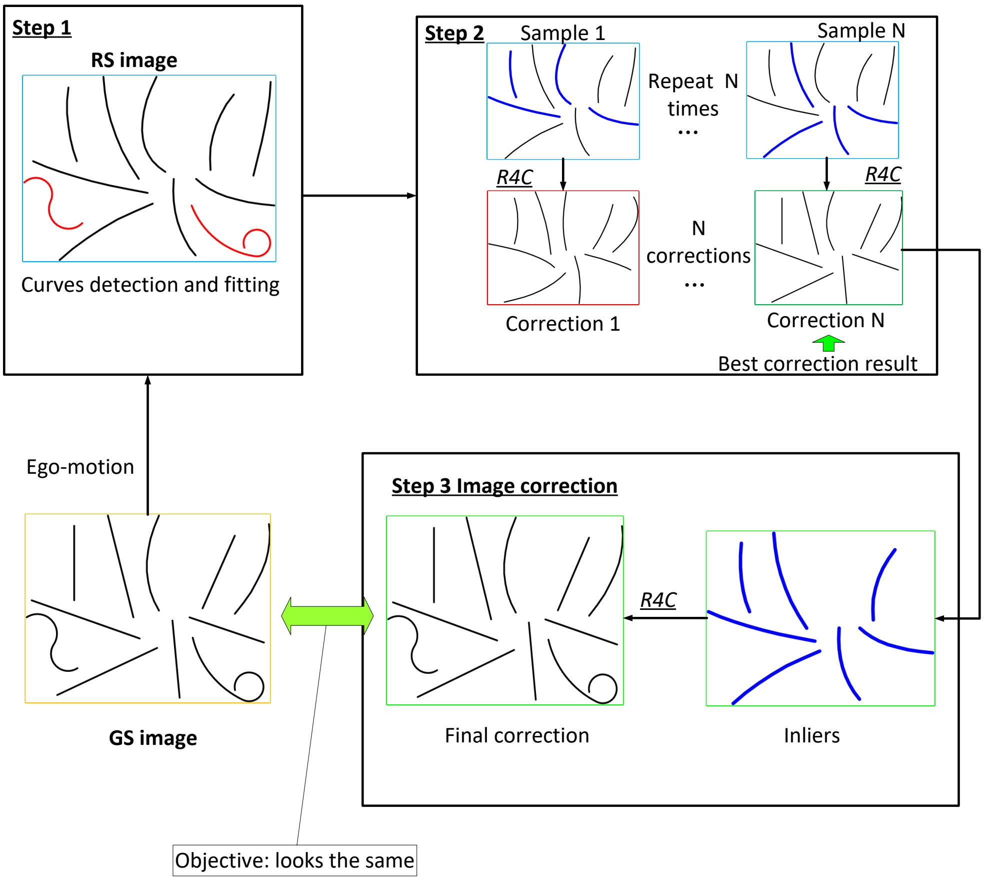


FIGURE 3.6: **Automatic actual 3D line selection.** Both 3D straight lines and curves observed will be projected as 2D curves on RS cameras under instantaneous-motion. We firstly fit all detected curve pixels to hyperbolic polynomials and discard curves with big fitting errors (red curves in step 1). In step 2, we randomly select putative sets of 4 curves (blue curves) and compute camera instantaneous-motion for each set. Then we correct all image candidate curves basing on the obtained instantaneous-motion and compute the global straightness score. After N samples, the sample with the smallest straightness score is chosen as best sample. After discarding curves whose corrected straightness exceeds a defined threshold, we obtain final set of inliers.

(ii) Run R4C to calculate ω_i based on $\mathbb{S}_i$.

(iii) Perform forward mapping Eq. (3.52) to all curves pixels and calculate straightnesses of each curves after correction.

- Select the sample with maximum number of inliers (straightness of the corrected curve smaller than ϵ pixels) as best solution over all samples.
- Refine the best solution by performing R4C again using the inliers.

The straightness of a curve is calculated by mean square of perpendicular offset of each curve pixel to its corresponding least-squares- fitting line. We set $\epsilon = 1$ experimentally and the number of samples N is set automatically used method in [Hartley and Zisserman, 2003].

Fig. 3.7 shows an example of outlier rejection procedure that some outliers survive from the first selection criteria. However, the RANSAC-like automatic feature selection reject those curves which are not corresponding to the actual 3D lines.

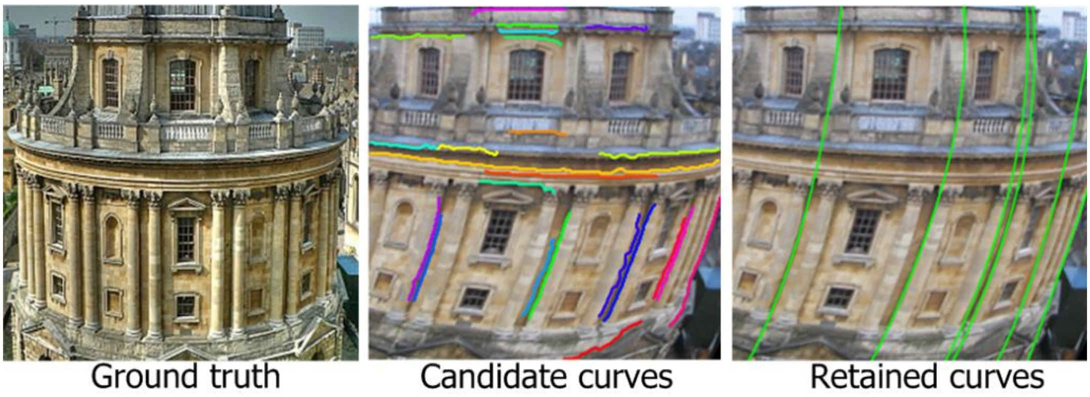


FIGURE 3.7: **An example of automatic feature slection.** Curves detected from the façade survived the first selection criteria, but not the 2$^{\text{nd}}$ round where we perform the automatic feature selection to reject the outliers.

3.6 3-step RSSfM using Lines

3.6.1 Step 1: Compensating the Effect of Rotational Velocities for Each Image

In order to compensate the effects of $\boldsymbol{\omega}$, we perform an inverse mapping to all point-matches among all the RS images using Eq. (3.52).

3.6.2 Step 2: SfM Using Extracted Rotational Velocities

After extracting rotational velocities for each image, we still need to recover both motion between images and linear-velocities during acquisition. This is achieved as follows: first each image is rectified by compensating the rotational velocities computed in the previous section. This results in a new image pair which looks like if each camera undergoes linear motion (pure translation) during acquisition. Thus, the epipolar geometry of this image pair is computed along with the translational instantaneous velocities of the cameras using the linear RS model. The advantage of using linear RS model in SfM is to avoid planar degenerate solutions described in [Albl et al., 2016b, Ito and Okatani,]. This is caused by the fact that using angular-velocities as unknown parameters will make the algorithm collapse into specific values during non-linear optimization. Thus, using linear RS model by fixing rotational velocities can avoid this degeneracy.

After the image correction, we can solve the relative pose problem of linear RS cameras using 5×5 essential matrix with point matches $[u_i, v_i]^T \leftrightarrow [u_i', v_i']^\top$ proposed by [Dai et al., 2016]:

$$\left| v_i'^2 \quad v_i'u_i' \quad v_i' \quad u_i' \quad 1 \right| \mathbf{E}_{5\times5} \left| v_i^2 \quad v_iu_i \quad v_i \quad u_i \quad 1 \right|^\top = 0 \tag{3.53}$$

With at least 20 point correspondences, $\mathbf{E}_{5\times5}$ can be computed as usual using a DLT (Direct Linear Transform) Algorithm. Then, the relative pose $[\mathbf{R}, \mathbf{t}]$ and the translational instantaneous velocities are extracted linearly from $\mathbf{E}_{5\times5}$. Finally, 3D points are reconstructed by triangulation [Albl et al., 2016a].

3.6.3 Step 3: Camera-based RS Bundle Adjustment

We present a novel camera-based RS bundle adjustment (**C-RSBA**) which is different from existing works that calculate reprojection errors with image measurements, thus imposing a constant raw index during optimization. This enables to refine the parameters obtained thanks to the straightness constraint by avoiding degenerate configurations, thus outperforming existing RSBA methods. Since the C-RSBA can serve for all the RS vision applications using the reprojection-based iterative refinement such as pose estimation [Ait-Aider et al., 2006, Magerand et al., 2012], VO [Schubert et al., 2018] and SLAM [Kim et al., 2016], we will discuss **C-RSBA** in details in chapter 4.

3.7 Experiments

In this section, we compare our image correction method (R4C with automatic feature selection) to existing works using both synthetic and real data. The results of the proposed 3-step RSSfM will be presented in Chapter 4.

Experiment setting. The experiments were conducted on a i5 CPU 2.8G Hz with 4G RAM. It took around 4.1s for curve detection and fitting, 1.9s for instantaneous-motion estimation with automatic feature selection, and 0.1s for image correction with 240 × 320 size. The proposed method was implemented in MATLAB. An improvement can be expected using C++.

Comparison to state-of-the-art. The proposed RS correction method was compared to three state-of-the-art techniques:

- **R4C**: The proposed linear 4-curve RS correction method with RANSAC-like automatic feature selection.
- **Rengarajan**: The single RS correction method using straightness constraint with MW assumption [Rengarajan et al., 2016][1].
- **CNN**: The single RS correction method using Convolutional Neural Network [Rengarajan et al., 2017][2].
- **Purkait**: The single RS correction method using vanising direction constraint with MW assumption [Purkait et al., 2017][3].

3.7.1 Synthetic RS Image Experiments

3.7.1.1 Grid Scene

We simulated a grid scene where MW assumption holds on as required by **Rengarajan** and **Purkait** (but not by our method). Images corresponding to random angular-velocities were generated using the following virtual camera parameters: focal = 1 unit, resolution as 640 × 480 and scan speed=$7.5 \times 10^-5 s/row$. Then values of ω were computed from deformed edges using **R4C**. While ground-truths are available, we evaluated instantaneous-rotation accuracy using both visual checking and mean value of estimated rotation errors $\bar{e}^{rotate}$ [Rengarajan et al., 2017] of each image row (calculated using

[1] The results are supplied by the authors upon request.

[2] https://github.com/yogeshbalaji/CVPR17∼Unrolling∼the∼shutter

[3] Self-implemented after having discussion with the authors.

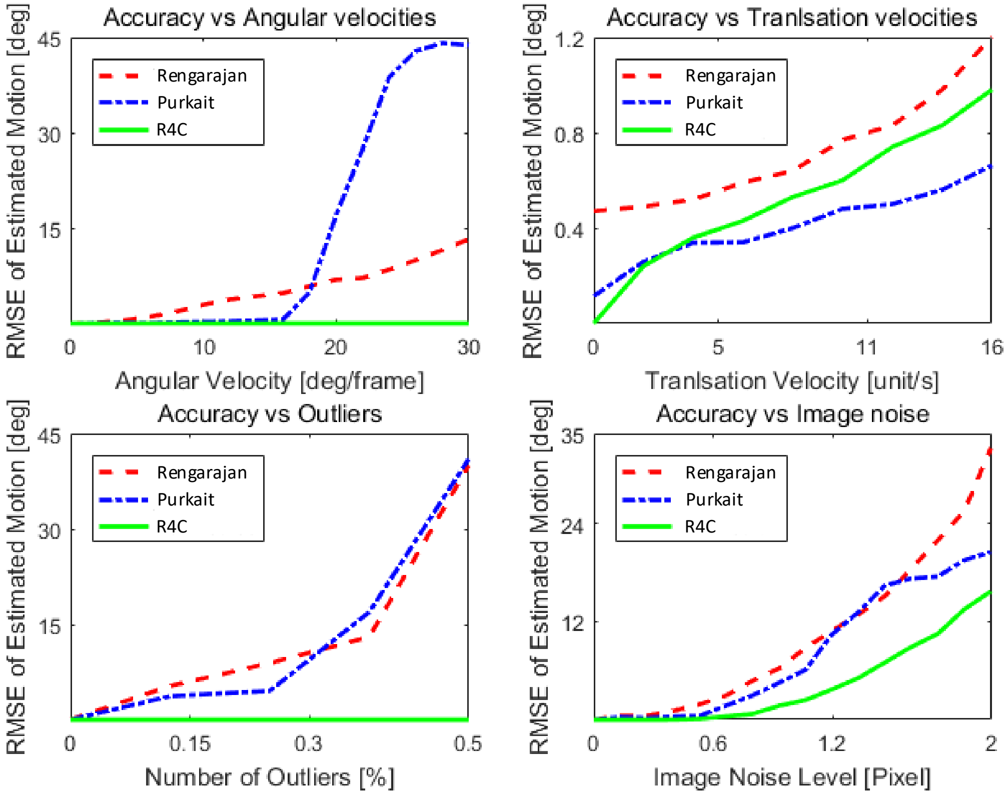


FIGURE 3.8: Comparison of the proposed method with **Rengarajan** and **Purkait** under different configurations (varying angular velocities, translation velocity, outlier curves number and image noise). We use mean value of estimated rotation errors $\bar{e}^{rotate}$ as a tool to quantitatively evaluate accuracy of instantaneous-motion estimation.

Eq. 2.20). 100 values of ω were generated randomly in different directions. Since **Rengarajan** uses z-axis rotate model fails in most cases. We compared our method to two other single-image based RS correction methods also using line features, namely **Rengarajan** and **Purkait**.

Gentle configuration: The results in Fig. 3.9 (first row) show that our method, **Rengarajan** and **Purkait** obtain correct results under gentle conditions ($\|\mathbf{d}\| = 5$ unit/s, $\|\omega\| = 5$ during acquisition, RS image with average 0.5 pixel noise and no outlier curves). However, significant differences appear when RS effect becomes more important.

Accuracy vs Angular Velocity: Experiments were carried out with $|\omega|$ varying from 0 to 30 degree/frame. Results in Fig. 3.8, 3.9 show that the RS instantaneous-motion estimation errors of **Rengarajan** climb from 0 to 14 degrees while errors of **Purkait** keep low under 15 deg/frame but dramatically increase with bigger ω. Inversely, **R4C** maintains the error under 1 deg.

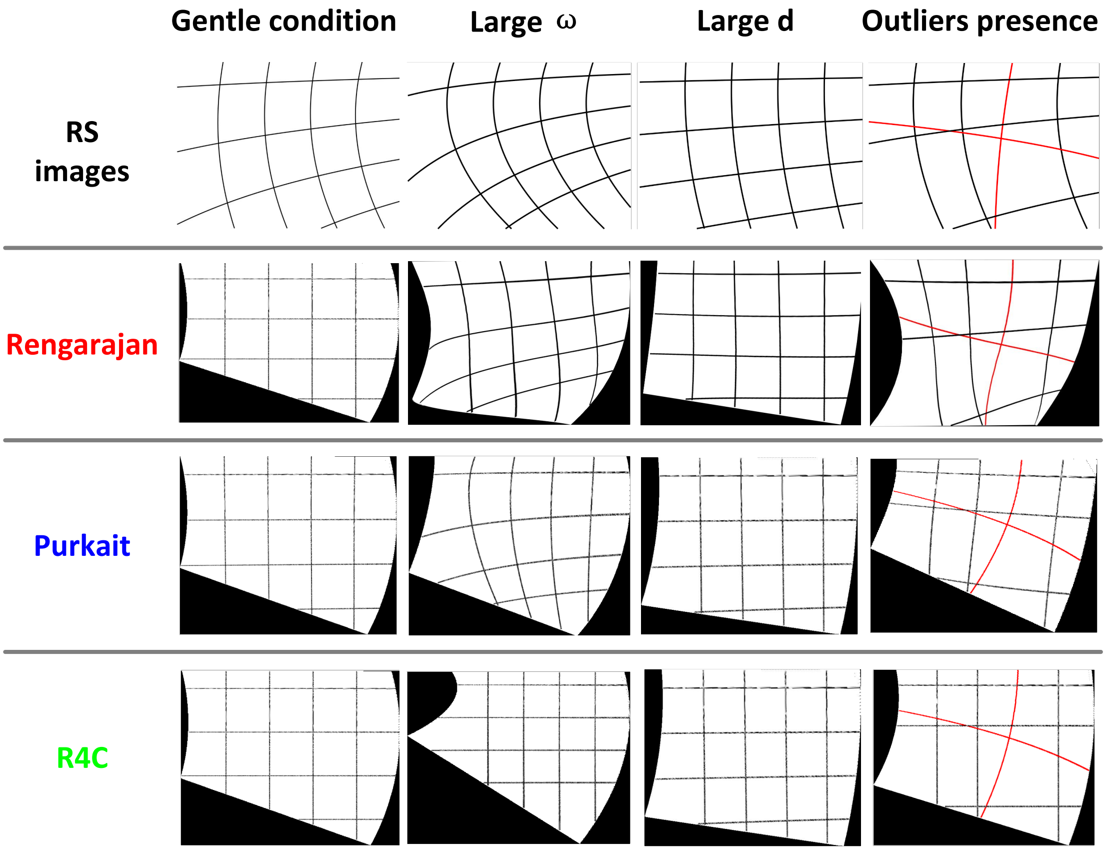
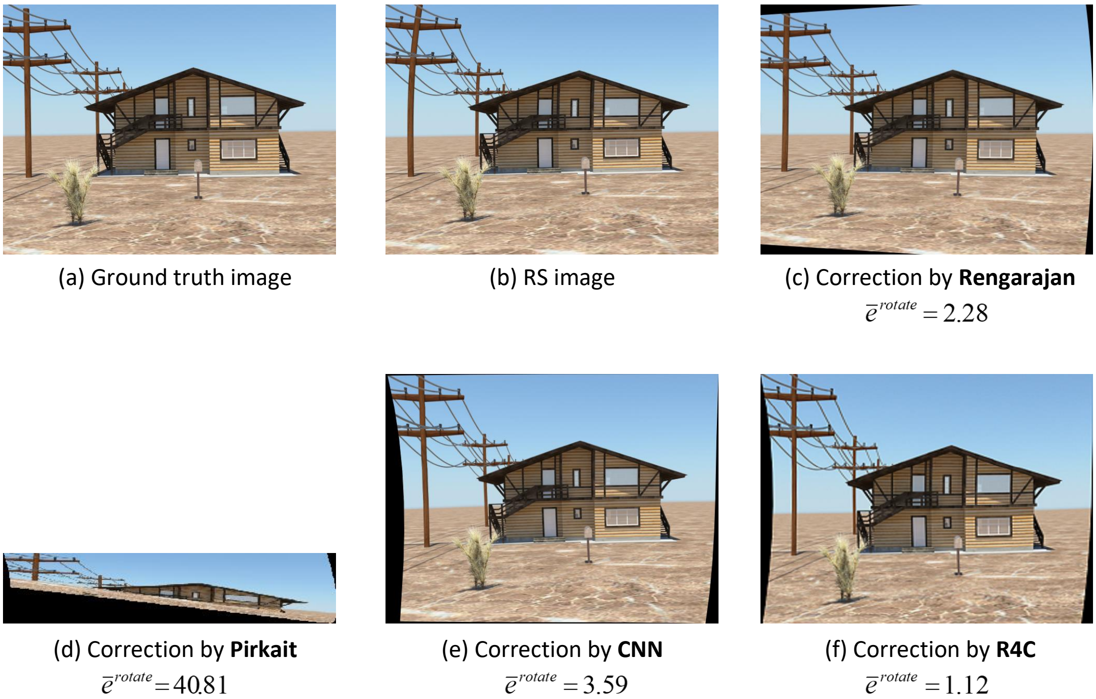


FIGURE 3.9: Comparison of the proposed method with **Rengarajan** and **Purkait** under gentle condition (first column), large ω (second column), large $\mathbf{d}$ (third column) and outlier presence (fourth column).

Accuracy vs Translational Velocity: Since all tested methods assume translational velocity is negligible during image exposure, we now verify accuracy of these three methods if RS translation velocity effect is significant. The translation velocity $\mathbf{d}$ was increased from 0 to 12 unit/s which is extremely high and rarely occurs in a real application context. The results in Fig. 3.8 shows that the three methods perform stable and achieve errors under 1.2 degree with big translation velocities.

Accuracy vs Outlier Curves: In this experiment, we simulated 3D straight lines and added 3D curves as outliers. The number of outliers was increased from 0% to 50%. Results in Fig. 3.8 and Fig. 3.9 show that both **Rengarajan** and **Purkait** fail in presence of outliers. In contrast, thanks to automatic feature selection, **R4C** obtains correct correction in different settings.

Accuracy vs Pixel Noise: We fixed the camera translational speed to 5 unit/s and the angular velocity to 5 deg/frame. We added a random Gausian noise to projected curve points(mean std dev increases from 0 to 2 pixels). The results in Fig. 3.8 demonstrate that **R4C** is much more robust against increasing noises compared to **Rengarajan** and **Purkait**.



(a) Ground truth image

(b) RS image

(c) Correction by **Rengarajan** $\bar{e}^{rotate} = 2.28$

(d) Correction by **Pirkait** $\bar{e}^{rotate} = 40.81$

(e) Correction by **CNN** $\bar{e}^{rotate} = 3.59$

(f) Correction by **R4C** $\bar{e}^{rotate} = 1.12$

FIGURE 3.10: Comparison of **R4C** (f) with **Rengarajan** (c), **CNN** (d) and **Purkait** (e) on a synthetic RS image dataset [Forssén and Ringaby, 2010]. The correction results are evaluated by using mean value of estimated rotation errors $\bar{e}^{rotate}$ of each row compared to ground truth.

3.7.1.2 Complex Urban Scene

We evaluated performances of our method **R4C** compared to **Rengarajan**, **CNN** and **Purkait** using synthetic RS images.

'House' dataset. Our first experiment based on a public synthetic RS image dataset [Forssén and Ringaby, 2010] which contains multiple RS videos filming a house scene. Since ground-truth of camera instantaneous-motions are known, we keep using $\bar{e}^{rotate}$ to evaluate accuracy of corrections.

The results in Fig. 3.10 shows that **Purkait** fails since house roof and current lead violate MW assumption. **R4C**, **Rengarajan** and **CNN** obtain corrected images visually close to ground-truth image. However, both **Rengarajan** and **CNN** use simplified instantaneous-motion model which can not recover instantaneous-rotation along all three x-y-z axis. In other words, visually acceptable correction does not ensure consistency of geometry.

The quantitative evaluation results in Fig. 3.10 using $\bar{e}^{rotate}$ demonstrate that our method not only offers visually pleasant corrected image, but also recovers images which better fit GS-based 3D geometry.

Urban scene dataset. We also perform our method with different baselines using RS images from urban scene dataset [Rengarajan et al., 2017] and results are shown in Fig. 3.11.

Since MW assumption holds on in the first RS image of Fig. 3.11, except **Rengarajan**, there are no significant curvatures left in the corrected images. However, **Purkait** keeps effects of vertical shrinking while **CNN** and our method obtain better visual corrections.

The second RS image shows a circular building with many 3D curves. The correction results demonstrate that our proposed method can successfully filter outliers while

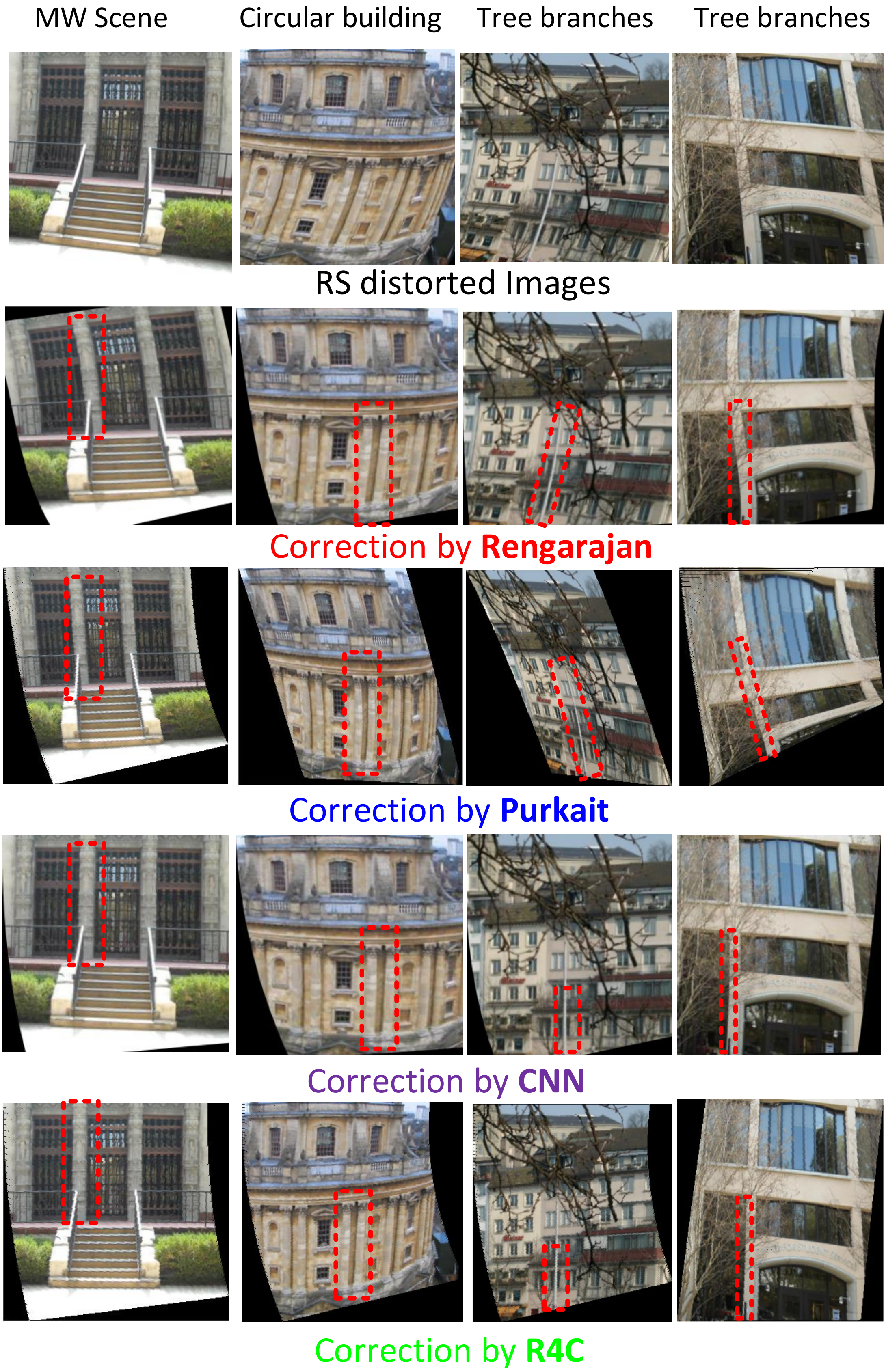


FIGURE 3.11: Visual comparison of the proposed method **R4C** with baselines **Rengarajan**, **CNN** and **Purkait**. The red boxes highlights the superiority of our method.

Rengarajan and **Purkait** fail. One can note that the corrected image of our method preserves curves belonging to the circular facade, meanwhile, **CNN** straights every curve. Moreover, the curves obtained by our method **R4C** fit better ellipse sections being the perspective projection of circles.

The third and fourth images are veiled by tree branches which can be regarded as outliers. As a result, distortions are observed on corrected images of **Rengarajan** and **Purkait** meanwhile **R4C** and **CNN** obtain similar corrections, however small curvature remains on results of **CNN**.

3.7.2 Real RS Image Experiments

RS video dataset: We conduct the first real images experiment on a RS video dataset [Ringaby and Forssén, 2012]. Comparison of our frame-by-frame RS corrections with methods of **Rengarajan**, **CNN** and **Purkait** are shown in Fig. 3.12. We use the approach described in [Purkait et al., 2017] to do a quantitative evaluation by counting mean value of the number of found inliers $|R_F|$ (point matches after estimating the fundamental matrix) from corrected frame pairs. The results show our method obtains the higher inlier number and demonstrate that it can better recover the consistency of projection geometry.

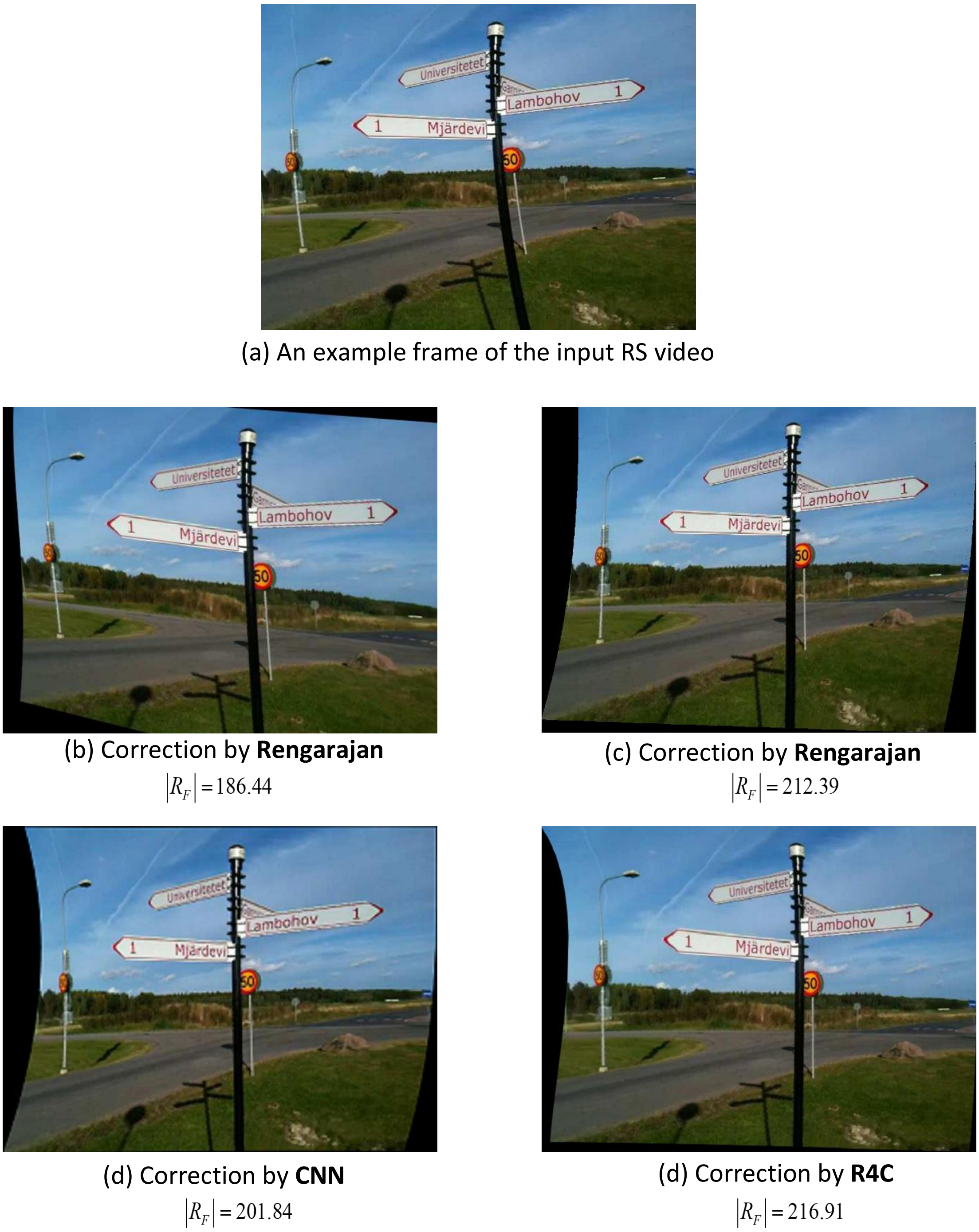
Hand-held web-cam dataset: We also compared our method on a challenging complex urban dataset which was captured by a Logitech camera with strong RS effects. It can be seen in Fig. 3.13 that **Rengarajan** obtain relatively acceptable corrections for the first image but fails for the second image due to the difficulty in grouping curves. Methods of **CNN** and **Purkait** fail in both cases because of the complexity of the scenes and the large angular velocities during acquisition. In contrast, we can see that our method obtains visually better corrections in both RS images.

3.8 Discussion and Conclusion

In this chapter, we first presented a novel RS correction method which uses line features. Unlike existing methods, which uses iterative solutions and make MW assumption, our method R4C computes linearly the camera instantaneous-motion using few image features. Besides, the method was integrated in a RANSAC-like framework which enables us to reject outlier curves making image correction more robust and fully automated.

Extensive experiments demonstrated the robustness and the accuracy of the proposed method in variable complex urban scenes or under extreme filming conditions. Specifically, our method not only produces visually pleasant corrections, but also is able to preserve consistency of geometry. Thus, the proposed method in this paper can serve for rolling shutter image correction as well as for pre-processing images in other computer vision applications such as feature matching and tracking.

Thanks to this RS correction method, we also presented a 3-steps method which solve RSSfM. Unlike with approaching methods, a general motion model is assumed and no a priori knowledge on the 3D lines is needed. Moreover, the first two steps of proposed solution are linear and works with fewer matches than previous methods.



(a) An example frame of the input RS video

(b) Correction by **Rengarajan**

$|R_F| = 186.44$

(c) Correction by **Rengarajan**

$|R_F| = 212.39$

(d) Correction by **CNN**

$|R_F| = 201.84$

(d) Correction by **R4C**

$|R_F| = 216.91$

FIGURE 3.12: An example frame from a RS video [Ringaby and Forssén, 2012] (a). The correction results by **Rengarajan**, **CNN**, **Purkait** and by our method **R4C** are shown in (b)-(e). A quantitative evaluation using the mean number of found inliers $|R_F|$ between corrected frame pairs are also shown below each corrected image.

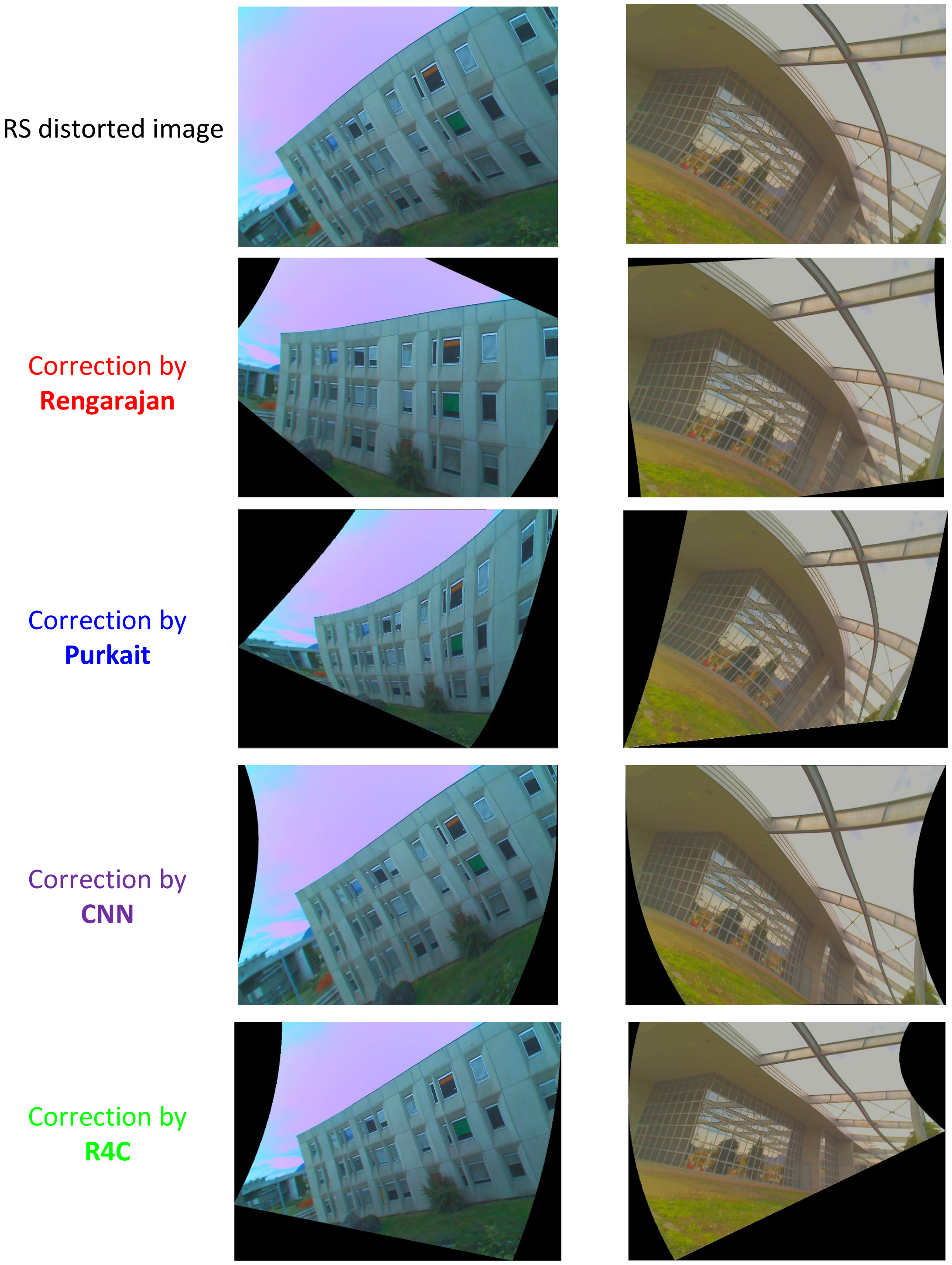


FIGURE 3.13: omparison of image correction results on two real RS images of a complex urban scene with strong RS effects against to methods of [Rengarajan et al., 2016, Purkait et al., 2017, Rengarajan et al., 2017].

Chapter 4

RS Bundle Adjustment Revisited

4.1 Context

In this chapter, we address on the bundle adjustment (BA) in RSSfM. In existing works, reprojection error is calculated by using row indexes of corresponding image measurements. We point out that this measurement-based method brings the risk of obtaining degenerated solution [Albl et al., 2016b]. Alternatively, we present a novel RS projection method based on camera motion and instantaneous-motion only without using image measurements, called camera-based projection. Then we incorporate it into RS BA as the final step of the proposed 3-steps RSSfM (chapter. 3).

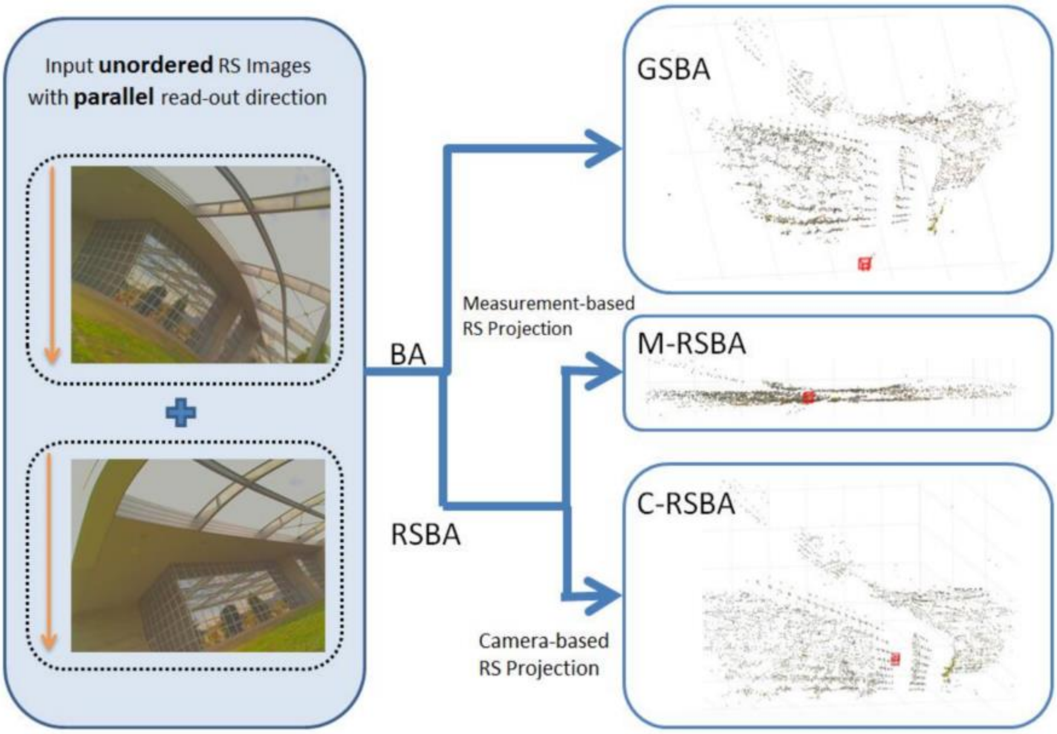


FIGURE 4.1: **RS SfM with three BA frameworks.** GSBA performs poorly in reconstruction due to RS effect presence. M-RSBA which uses image measurements to calculate reprojection points collapse into degeneracy and provides incorrect reconstruction when input images with parallel read-out directions. In contrast the proposed method C-RSBA which uses camera-based RS projection algorithm survives the degeneracy.

We provide analysis of why measurement-based RS BA tends to collapse into degeneracy, while our proposed camera-based RS BA survives. Thus, as shown in Fig. 4.1, different from the state-of-the-art works which require input images with distinct readout directions, our method allows more common and nature at capture style.

Both synthetic and real experiments we demonstrate that with the help of the proposed RS BA method, our 3-steps RSSfM pipeline (chapter. 3) outperforms existing works in accuracy and removes constraint on filming style.

4.2 Related Work

Estimating parameters by minimizing reprojection (photometric) error is a common technique in 3D vision applications. It has been widely used in RS applications, for instance, in object or pose calculation [Ait-Aider et al., 2006], SfM [Im et al., 2018, Duchamp et al., 2015, Albl et al., 2016b] and SLAM [Kim et al., 2016]. Whereas the reprojection of points involves only the pose and the point cartesian coordinates with GS images, it requires image measurements, that is to say the line indexes of each point, with RS images . The problem is that for reprojection to be geometrically consistent, these indexes must evolve during iterative optimization. We found out that there are two main drawbacks of this measurement-based projection method:

- Measurement-based projection cannot simulate true RS projection procedure and leads to accuracy loss.
- Measurement-based projection bring risks of collapse into degeneracy during RS BA (Fig. 4.1).

Thus, we need a new algorithm to calculate RS projection point and incorporate it with RS BA to improve SfM accuracy and avoid degeneracy.

4.3 RS Reprojection

4.3.1 Formulation of Rolling Shutter Bundle Adjustment (RSBA)

Considering a sequence of two or more images, a GSBA can be performed to refine motion $[\mathbf{R}|\mathbf{t}]$ and scene structure $\mathbf{P}$ (as shown in Eq. (2.10).

In contrast, the objective of RSBA is to refine motion $[\mathbf{R}|\mathbf{t}]$, camera instantaneous-motion $[\boldsymbol{\omega}|\mathbf{d}]$ and scene structure $\mathbf{P}$ simultaneously. Thus, Eq. (2.10) changes to:

$$\begin{aligned} &\left\{ \{\mathbf{P}_i^*\}, [\mathbf{R}^{j*}|\mathbf{t}^{j*}], [\boldsymbol{\omega}^{j*}|\mathbf{d}^{j*}] \right\} = \arg\min \sum_{j=1}^{m} \sum_{i=1}^{n} V_i^j \left\| \mathbf{e}_i^j \right\|^2 \\ &\text{with} \quad \mathbf{e}_i^j = \tilde{\mathbf{m}}_i^j - \mathfrak{p}(\mathbf{P}_i) \end{aligned} \tag{4.1}$$

where $\mathbf{e}_i^j$ is reprojection error, which is the distance between measured image point $\tilde{\mathbf{m}}_i^j$ and the reprojection point of $\mathbf{P}_i$ on j^{th} image. V_i^j indicates the visible index which equals to 1 if 3D point $\mathbf{P}_i$ is seen on the j^{th} image. $\mathfrak{p}$ is the RS projection operator which transforms the 3D point $\mathbf{P}_i$ in world coordinate system into image point $\mathbf{m}_i^j$.

4.3.2 Measurement-based Projection

To the best of our knowledge, all existing works [Ait-Aider et al., 2006, Hedborg et al., 2012, Albl et al., 2016b] used row index $\tilde{v}_i^j$ of measurements $\tilde{\mathbf{m}}_\mathbf{i}^\mathbf{j}$ to calculate reprojected points $\mathbf{m}_\mathbf{i}^\mathbf{j}$. This method is called measurement-based projection ($\mathfrak{p}^m$):

$$\begin{aligned} s_i \begin{bmatrix} \mathbf{m}_i \\ 1 \end{bmatrix} &= \mathfrak{p}^m(\mathbf{P}_i) = \mathbf{K}[\mathbf{R}(\tilde{v}_i) \quad \mathbf{t}(\tilde{v}_i)] \begin{bmatrix} \mathbf{P}_i \\ 1 \end{bmatrix} \\ \text{with} \quad \mathbf{R}(v_i) &= (\mathbf{I} + [\boldsymbol{\omega}]_\times \tilde{v}_i)\mathbf{R}_0 \\ \mathbf{t}(v_i) &= \mathbf{t}_0 + \mathbf{d}\tilde{v}_i \end{aligned} \tag{4.2}$$

where the row index of measurement information $\tilde{v}_i^j$ is the key to calculate the RS projection point.

4.3.3 Camera-based Projection

Alternatively, we propose a novel approach to calculate reprojected points purely basing on RS camera model (pose $[\mathbf{R}, \mathbf{t}]$ and instantaneous-motion $\boldsymbol{\omega}, \mathbf{d}$), without using image measurement information. We call this camera-based RS projection ($\mathfrak{p}^c$):

$$\begin{aligned} \mathbf{m}_i &= \begin{pmatrix} u_i \\ v_i \end{pmatrix} = \mathfrak{p}^c(\mathbf{P}_i) \\ \text{where} \quad v_i &= \frac{-b \pm \sqrt{-4ac + b^2}}{2a} \\ u_i &= \frac{(\mathbf{KR})^{(1)}\mathbf{P}_i + \hat{\mathbf{R}}^{(1)}\mathbf{P}_i v_i + (\mathbf{Kt})^{(1)} + (\mathbf{Kd})^{(1)} v_i}{(\mathbf{KR})^{(3)}\mathbf{P}_i + \hat{\mathbf{R}}^{(3)}\mathbf{P}_i v_i + (\mathbf{Kt})^{(3)} + (\mathbf{Kd}^{(3)}) v_i} \\ \text{with} \quad \hat{\mathbf{R}} &= \mathbf{K}[\boldsymbol{\omega}]_\times \mathbf{R} \\ a &= \hat{\mathbf{R}}^{(3)}\mathbf{P}_i + (\mathbf{Kd})^{(3)} \\ b &= \mathbf{R}^{(3)}\mathbf{P}_i + (\mathbf{Kt})^{(3)} - \hat{\mathbf{R}}^{(2)}\mathbf{P}_i - (\mathbf{Kd})^{(2)} \\ c &= -\mathbf{R}^{(2)}\mathbf{P}_i - (\mathbf{Kt})^{(2)} \end{aligned} \tag{4.3}$$

where $\mathbf{M}^{(r)}$ is the r^{th} row of matrix $\mathbf{M}$. a, b and c are three auxiliary variables while $\hat{\mathbf{R}}$ is an auxiliary matrix.

Proof of Eq. (4.3): Substituting Eq. (2.20) into Eq. (2.11):

$$s_i \begin{pmatrix} \mathbf{m}_i \\ 1 \end{pmatrix} = \begin{pmatrix} s_i u_i \\ s_i v_i \\ s_i \end{pmatrix} = (\mathbf{I} + [\boldsymbol{\omega}]_\times v_i)\mathbf{R}\mathbf{P}_i + \mathbf{t} + \mathbf{d}v_i \tag{4.4}$$

Then, by substituting the third row of Eq. (4.4) into the second row of equation above, we have a quadratic equation where v_i is the only unknown: $av_i^2 + bv_i + c = 0$. We provide recipes of solving this equation later. After we obtain v_i, u_i can be easily calculated by substituting third row into the first row of Eq. (4.4).

4.3.3.1 Double-projection

The quadratic equation in Eq. (4.3) yields two geometric feasible solutions v_1 and v_2 named as double-projection pattern (shown in Fig. 4.2). In common and practical configurations, there are usually one solution located within image range while another one

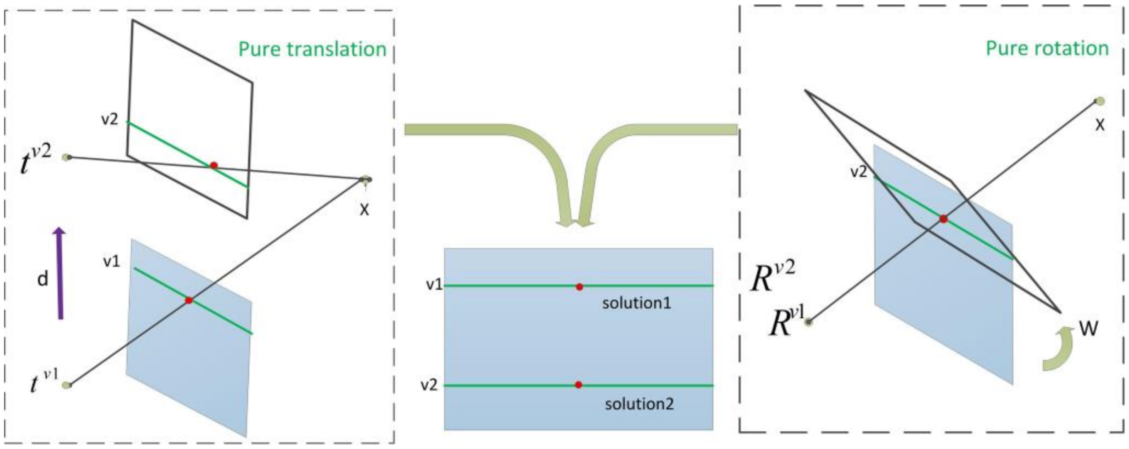


FIGURE 4.2: **Two examples of double-projections pattern.** On the left, a RS camera is under pure translation heading to $[0;1;0]^T$ rapidly. Besides, a example of pure-rotation with axis $(1,0,0)$ shown on the right. During the acquisition, a 3D point **X** will be observed twice at row $v1$ and $v2$ if the speeds are big enough.

far away from image range. We analyze two typical cases in Fig. 4.2. A 3D point with 10 unit depth projected as two points within image range requires translation speed at least 500 unit/s and angular speed 50 rad/s which are rarely achieved in real applications.

Thus, since only one solution is consistent with the pose in practice. We propose to always select the solution that is nearer measurement to image by comparing reprojection values obtained by using v_1 and v_2 respectively during bundle adjustment (each round in non-linear optimization). This selection provides a solution that maintains projection point within the camera field of view.

It is important to note that camera-based RS projection approach not only can be used in RS SfM or existing works [Hedborg et al., 2012, Sau, 2013, Duchamp et al., 2015, Kim et al., 2016] , but also in other applications basing on reprojection errors minimization such as pose estimation [Ait-Aider et al., 2006, Magerand et al., 2012].

4.3.4 Comparison of the Two RS Projection Algorithms

The differences between classical GSBA, measurement-based RSBA and camera-based RSBA are summarized in table. 4.1.

	GSBA	M-RSBA	C-RSBA
Input	$\mathbf{P}, \mathbf{R}, \mathbf{t}$	$\mathbf{P}, \mathbf{R}, \mathbf{t}, \boldsymbol{\omega}, \mathbf{d}, \tilde{v}$	$\mathbf{P}, \mathbf{R}, \mathbf{t}, \boldsymbol{\omega}, \mathbf{d}$
Number of solution	1	1	2

TABLE 4.1: Comparison of GSBA, M-RSBA and C-RSBA.

4.4 Measurement-based RSBA

Measurement-based RSBA (M-RSBA) uses measurement-based RS projection algorithm to calculate the reprojection point during RSBA. In other words, reprojection is computed using $\mathfrak{p}^m(\mathbf{P}_i)$ described in Eq. (4.2).

Disadvantages of M-RSBA: M-RSBA uses measurements as pre-knowledge to calculate reprojection points and makes exposure-delay of each point fixed. However, during optimization, exposure-delays should change at each iteration in order to maintain structure and motion consistency according to row-index of respected reprojections. Therefore, we indicate two drawbacks of M-RSBA:

- It can't simulate true projection during optimization which conducts accuracy loss.
- It brings the risks of degeneracy as shown in [Albl et al., 2016b].

4.5 Camera-based RSBA

Camera-based RSBA (C-RSBA) uses Camera-based RS projection algorithm to calculate the reprojection point during RSBA. In other word, $\mathfrak{p}(\mathbf{P}_i)$ in Eq. (4.1) can be computed using $\mathfrak{p}^c(\mathbf{P}_i)$ described in Eq. (4.3).

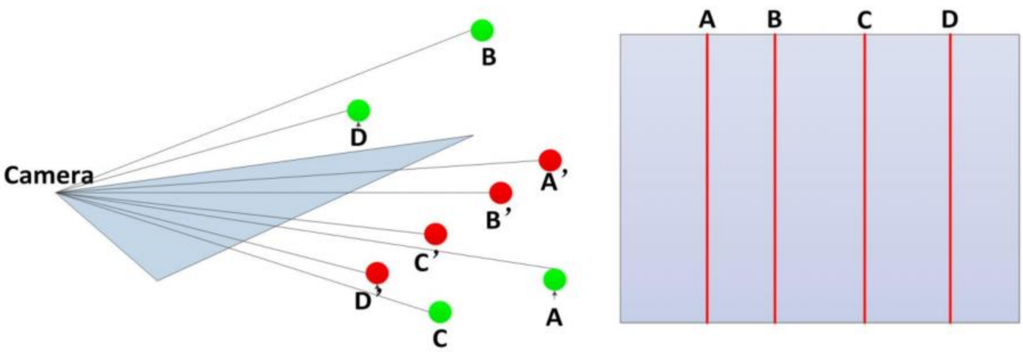


FIGURE 4.3: **How planar degeneracy raised?** Any RS image generated by 3D scene (green points) can also be explained by 3D points located in plane $y = 0$ with $\boldsymbol{\omega} = [1;0;0]$ during acquisition. In such situation, all 3D points will projected to each row of image (on right side), thus, planar degeneracy raised.

4.5.1 Planar Degeneracy

[Albl et al., 2016b] investigated mechanism of planar degeneracy which often arises during RSBA using measurement-based projection: Multiple RS views with parallel read-out directions will collapse into solutions that consist in cameras with $\boldsymbol{\omega} = [-1;0;0]$ and 3D points located on $y = 0$ plane since image noise presence as shown in Fig. 4.3.

Under this degeneracy, each 3D point can be observed in any row on image. During RSBA, we can always find projection point on row v_i that satisfies constraints of measurements-based projection in Eq. (4.3). Since image noise presence, this degeneracy provides solution superior to ground-truth.

4.5.2 Why does C-RSBA Survives Planar Degeneracy?

In contrast, we found out that by using camera-based method to calculate reprojection errors, C-RSBA survives planar degeneracy. The observations for the difference in behavior of C-RSBA and M-RSBA is given bellow.

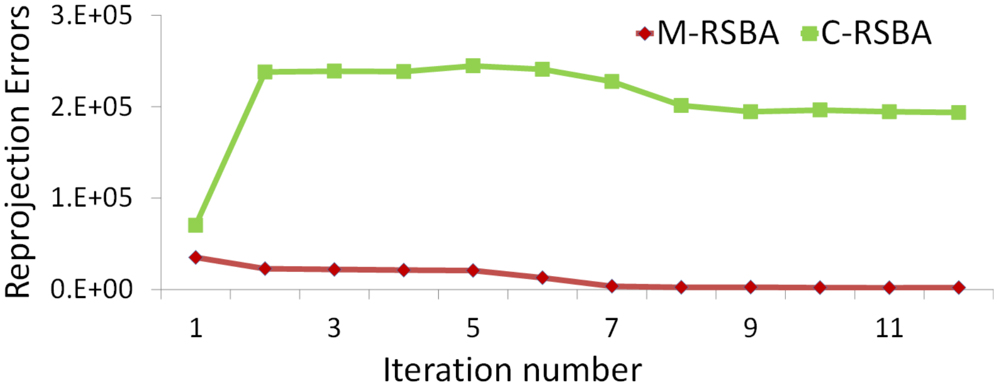


FIGURE 4.4: **Comparaison of C-RSBA and M-RSBA reprojection errors.** We perform M-RSBA using multiple RS views with parallel read-out directions. In each iteration during optimization, we also calculate overall reprojection errors by $\mathfrak{p}^m$ for M-RSBA (blue) and $\mathfrak{p}^c$ for C-RSBA (green) based.

Proposition: When RSBA collapses towards planar degeneracy, reprojection errors calculated by $\mathfrak{p}^m$ gradually descend to 0 while errors using $\mathfrak{p}^c$ become huge (Fig. 4.4).

Without formal demonstration, we show two examples to compare the performances of M-RSBA and C-RSBA with approximate critical configurations. We assume a single RS camera and instantaneous-motion close to planar critical configuration as $\boldsymbol{\omega} = [-1, 0, 0]^\top$, $\mathbf{d} = \mathbf{0}$ and 3D points (in camera coordinate system) close to $\mathbf{P} = [X, 0, Z]^T$. $\mathbf{C}$ indicates camera pose and instantaneous-motion parameters.

Example 1 (M-RSBA): Assuming RS camera with referenced pose $[\mathbf{I}, \mathbf{0}]$. We firstly set $\boldsymbol{\omega} = [-1 + \Delta_\omega, 0, 0]^T$ and $\mathbf{P} = [X, 0, Z]^\top$. Where Δ_ω are tiny changes of ω_x.

$$\mathbf{e} = \begin{bmatrix} e_u \\ e_v \end{bmatrix} = \tilde{\mathbf{m}} - \mathfrak{p}^m(\mathbf{R}, \mathbf{t}, \boldsymbol{\omega}, \mathbf{d}, \mathbf{P}, \tilde{v}) = \begin{bmatrix} \tilde{u} - \frac{X}{Z} \\ \tilde{v}\Delta_\omega \end{bmatrix} \tag{4.5}$$

when Δ_ω is changing closer and closer to 0, the second component of reprojection error e_v is also becoming smaller and tends to 0.

Then, we investigate reprojection errors when Y of $\mathbf{P}$ is optimized towards 0. we set $\mathbf{P} = [X, \Delta_Y, Z]^T$. Where Δ_Y are tiny changes of Y.

$$\mathbf{e} = \begin{bmatrix} e_u \\ e_v \end{bmatrix} = \tilde{\mathbf{m}} - \mathfrak{p}^m(\mathbf{R}, \mathbf{t}, \boldsymbol{\omega}, \mathbf{d}, \mathbf{P}, \tilde{v}) = \begin{bmatrix} \tilde{u} - \frac{X}{Z} \\ \frac{\tilde{v}^2+1}{\tilde{v} - Z/\Delta_Y} \end{bmatrix} \tag{4.6}$$

Similarly, when Δ_Y is closing to 0, e_v are reduced to 0. Therefore, decrease of Δ_ω and Δ_Y makes e_v reduced to 0.

Therefore, the reprojection error by using $\mathfrak{p}^m$ is:

$$\mathbf{e} = \begin{bmatrix} e_u \\ e_v \end{bmatrix} = \tilde{\mathbf{m}} - \mathfrak{p}^m(\mathbf{C}, \mathbf{P}, \tilde{v}) = \begin{bmatrix} \tilde{u} - \frac{X}{Z} \\ 0 \end{bmatrix} \tag{4.7}$$

Simultaneously, $[X, 0, Z]^\top$ is further optimized to make e_u also reduced to 0. Finally, overall error $\mathbf{e}$ will descend to 0.

Example 2 (C-RSBA): We first assume $\omega_\times$ of instantaneous-motion close to -1 as $[-1+\Delta_\omega, 0, 0]^T$. 3D point located at $y = 0$ plane.

$$e = \begin{bmatrix} e_u \\ e_v \end{bmatrix} = \tilde{\mathbf{m}} - \mathfrak{p}^{\mathbf{c}}(\mathbf{R}, \mathbf{t}, \boldsymbol{\omega}, \mathbf{d}, \mathbf{P}) = \begin{bmatrix} \tilde{u} - \frac{X}{Z} \\ \tilde{v} \end{bmatrix} \tag{4.8}$$

when $\boldsymbol{\omega}$ is close to $[-1, 0, 0]^\top$, the projection points will gather on a line $v = 0$. Thus, e_v becomes $\sum \left| v_i^j \right|$.

Again, we investigate reprojection errors with $\mathbf{P} = [X, \Delta_Y, Z]^\top$ and $\boldsymbol{\omega} = [-1, 0, 0]^\top$:

$$e = \begin{bmatrix} e_u \\ e_v \end{bmatrix} = \tilde{\mathbf{m}} - \mathfrak{p}^{\mathbf{c}}(\mathbf{R}, \mathbf{t}, \boldsymbol{\omega}, \mathbf{d}, \mathbf{P}) = \begin{bmatrix} \tilde{u} - \frac{X}{Z} \\ \tilde{v} \end{bmatrix} \tag{4.9}$$

In such a case, reprojected points located at infinity on image plane which is regarded with reprojection point at $v = 0$ as double-projection pattern. However, according to $\mathfrak{p}^{\mathbf{c}}$, we always choose $v = 0$ as final solution. Thus e_v will becomes $\left| v_i^j \right|$.

Therefore, the reprojection error by using $\mathfrak{p}^c$ is,

$$e = \begin{bmatrix} e_u \\ e_v \end{bmatrix} = \tilde{\mathbf{m}} - \mathfrak{p}^{\mathbf{c}}(\mathbf{C}, \mathbf{P}) = \begin{bmatrix} \tilde{u} - \frac{X}{Z} \\ \tilde{v} \end{bmatrix} \tag{4.10}$$

The overall reprojection becomes $\left| v_i^j \right|$ which may be even larger than reprojection error of start point.

Discussion. Through example 1 and example 2, one can observe that planar degenerate solution is a perfect minimum for cost function of M-RSBA while as plateau for C-RSBA. This explains how C-RSBA successfully avoids this degeneracy. An example of reprojection errors of M-RSBA and C-RSBA when configurations are slipping towards planar degeneracy (shown in Fig. 4.4) illustrates our proposition.

Thus, without constraints on camera motions (e.g. perpendicular read-out directions), C-RSBA successfully handles planar degeneracy.

4.6 Experiments

The proposed method C-RSBA was evaluated on both synthetic and real data. It was also compared to two BA method:

- Classical GS BA method, close to [Lourakis and Argyros, 2009].
- State-of-the art RS BA method M-RSBA [Albl et al., 2016b].

4.6.1 BA and Read-out directions

The angles between read-out directions among the image sequence have a significant impact on final reconstruction quality. Thus we designed a simulation experiment to evaluate GSBA, M-RSBA (initialized by GSBA) and C-RSBA (initialized by the proposed linear two-step method with respect to this parameter).

Three cameras generated randomly on a sphere with a radius of 1 unit and heading to a cubical scene with varying average scanning angles from 0 to 90 deg. In Fig. 4.5, a deformed 3D cube is being reconstructed by GSBA in both parallel and perpendicular

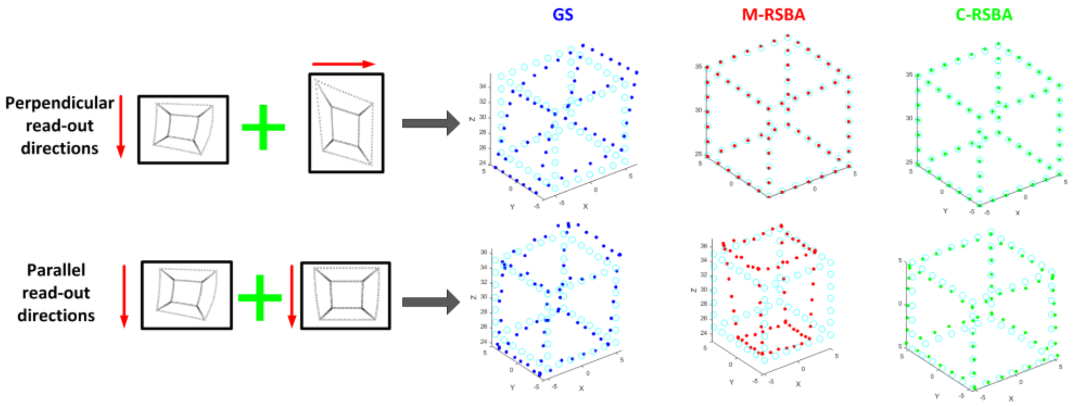

GS
M-RSBA
C-RSBA
Perpendicular read-out directions
Parallel read-out directions

FIGURE 4.5: Reconstruction results of GSBA (blue), M-RSBA (red) and C-RSBA (green) by using images with parallel and perpendicular read-out directions in comparison to ground-truth (cyan).

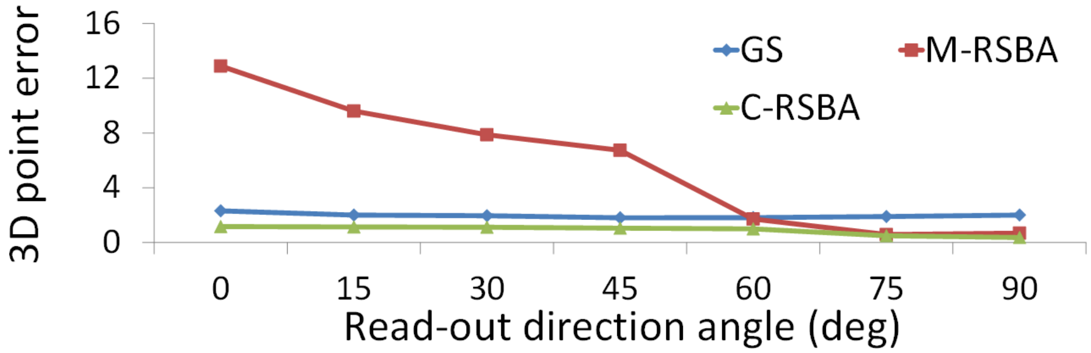


FIGURE 4.6: Reconstruction errors of GSBA, M-RSBA and C-RSBA with read-out direction angles varying from 0 to 90 degrees. M-RSBA provides better results than GSBA only when read-out direction angles are big (up to 60 degrees). C-RSBA obtains accurate reconstructions stably with varying direction angles.

read-out directions cases. M-RSBA obtains correct reconstruction using images with perpendicular read-out directions but fails in parallel one, which is a common configuration in practical applications. The proposed C-RSBA reconstructs a correct 3D scene in both parallel and perpendicular cases.

In order to draw a quantitative conclusion, we used the sum of distances between reconstructed 3D points and ground-truth 3D points as a criteria to evaluate SfM performances. Results in Fig. 4.6 show that M-RSBA achieves better reconstruction than GSBA when read-out direction angles are bigger than 60 deg, while C-RSBA obtains higher-accuracy and is more stable with close read-out directions (below 30 deg).

4.6.2 Noise Level Effect

In order to evaluate the effect of noise level on RSBA, we added random Gaussian noise to image measurements. We designed five comparison groups: GSBA, M-RSBA and C-RSBA using parallel and perpendicular scanning (read-out) directions RS images respectively. The results are shown in Fig. 4.7. M-RSBA and C-RSBA achieve the same SfM

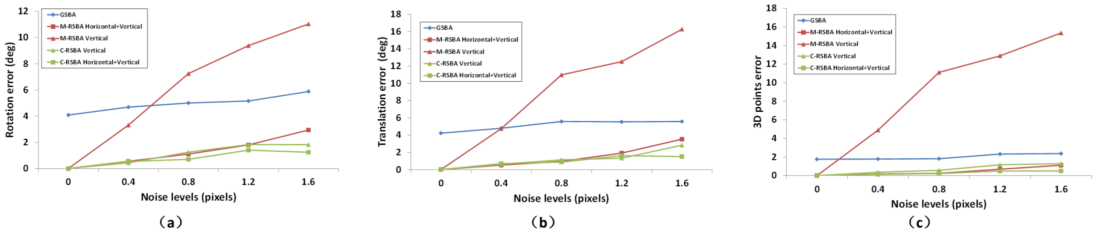


FIGURE 4.7: Simulated experiments with randomly distributed RS cameras. For each configuration, we perform GSBA, M-RSBA and C-RSBA with parallel (vertical) and perpendicular (vertical+horizontal) read-out directions respectively to achieve SfM with different image noise levels. The estimation errors of rotation **(a)**, translation **(b)** and reconstruction **(c)** demonstrate that with small read-out direction angle, M-RSBA fails while C-RSBA provides highest quality estimation in both parallel and perpendicular read-out directions cases.

accuracy superior to GSBA in noise free cases by using parallel read-out direction. However, M-RSBA collapse into degeneracy deeper and deeper with increasing noise levels and even much worse than GSBA. In contrast,estimation. Another interesting observation is that errors of C-RSBA using parallel readout directions are even lower than errors of M-RSBA with perpendicular directions.

4.6.3 3-steps RSSfM Method with RSBA

We incorporated C-RSBA into the pipeline of the 3-steps RSSfM method described in chapter 3 and show the final results in this experiment.

4.6.3.1 Synthetic RS Images

Synthetic RS benchmark [Forssén and Ringaby, 2010]: We evaluated our three-steps RS SfM approach on synthetic RS images benchmarked by Forssen et al. [Forssén and Ringaby, 2010]. Comparison of reconstructed 3D scene by our proposed method and GS approach are given in Fig. 4.8.

Synthetic RS benchmark [Kim et al., 2016]: We further evaluated performances of GSBA, M-RSBA and 3-steps method using synthetic RS images form public benchmark 'Sequence 77' [Kim et al., 2016]. Contrarily to the approach used in [Kim et al., 2016] which requires smooth and continuous camera trajectory, we address the case of unordered sets of images. Thus, instead of using every frame of the video, we randomly picked 24 non-successive frames. The results in Fig. 4.9 show that C-RSBA successfully estimates camera poses near ground-truth trajectory and achieves correct 3D scene reconstruction. In contrast, GSBA and M-RSBA fail in achieving RS SfM.

4.6.3.2 Real Images

Finally we compared GSBA, M-RSBA and C-RSBA on two real RS image sequences. The first data sets [Hedborg et al., 2012] captured by an iPhone4 camera for facade of warehouse and a road along wall. The second dataset shows a real complex building captured by a Logitech camera with strong RS effects. All images were captured with small read-out direction angles. The results shown in Fig. 4.10 confirmed our predication in

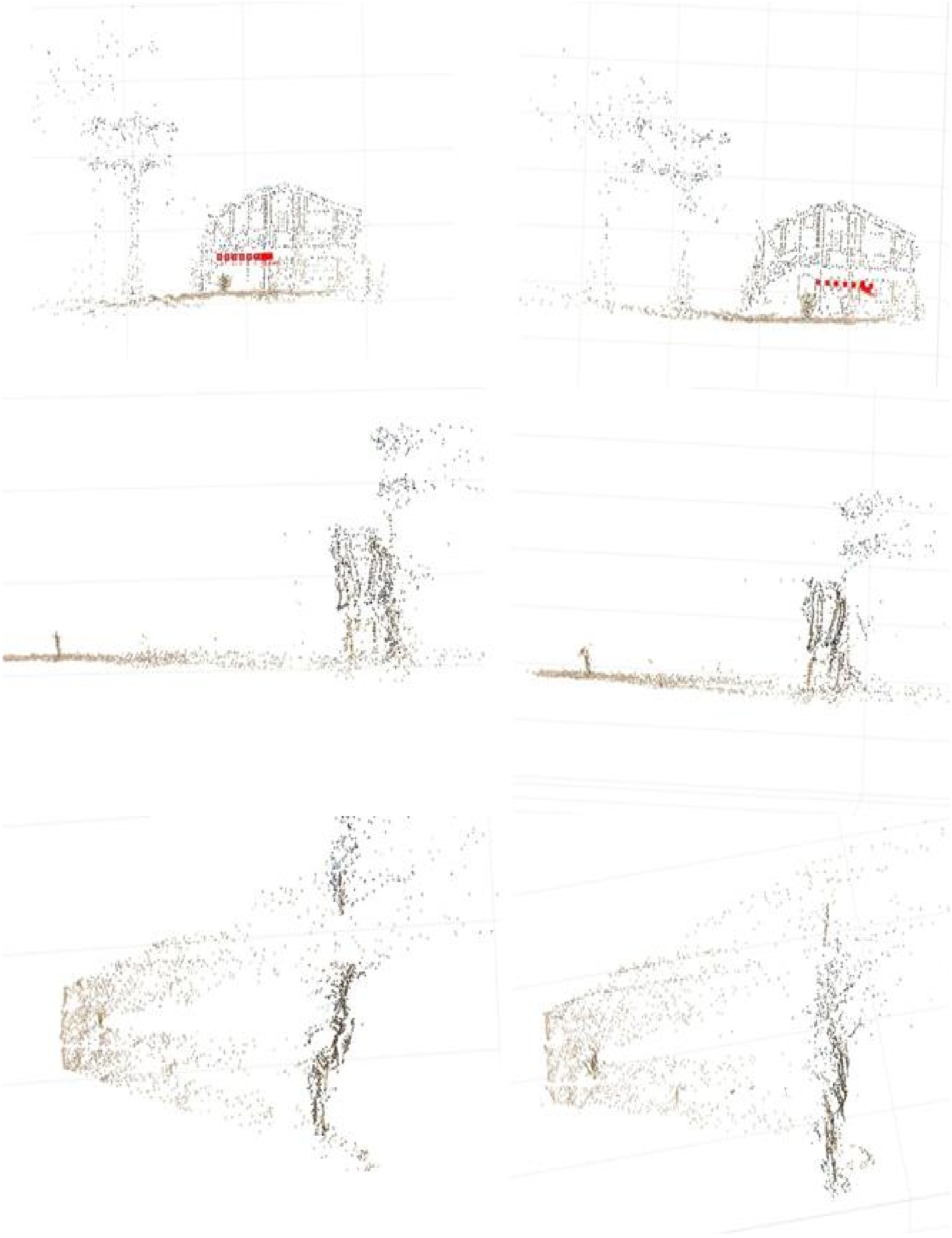


FIGURE 4.8: **Reconstructed 3D scene by proposed method and GS method.** 3D reconstructed scene by GS BA (left) and our proposed method (right) from front view (first row), side view (second row) and top view (third row).

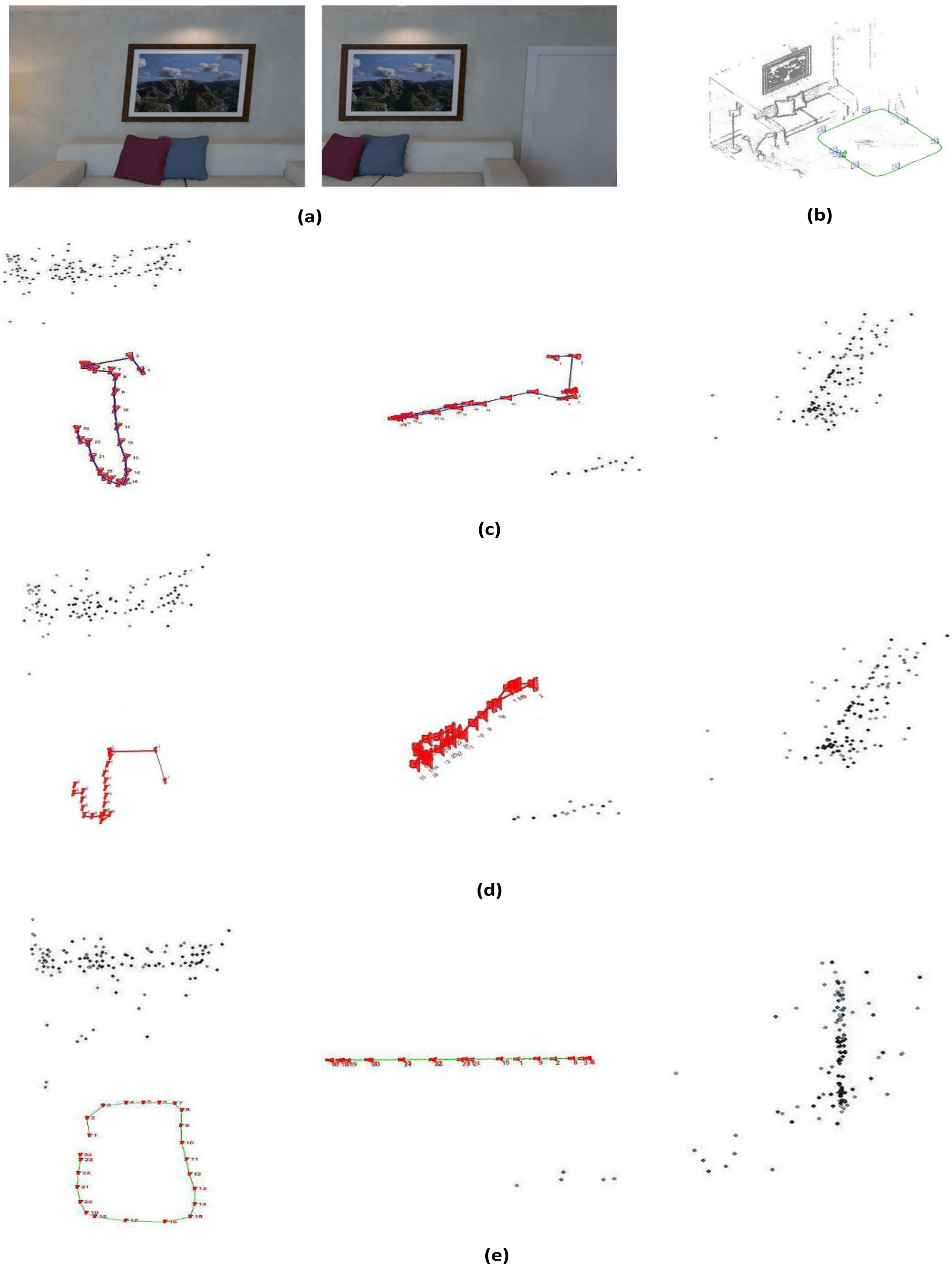


FIGURE 4.9: SfM with unordered images. **(a)** 24 non-successive frames from 'Sequence 77' as an input RS image set. SfM results of GSBA**(c)**, M-RSBA**(d)** and C-RSBA**(e)** (the first and second columns are top and side views respectively). Compared to ground-truth shown in **(b)** [Kim et al., 2016], GSBA and C-RSBA failed in motion calculation and obtains deformed 3D scene (tilt walls). While C-RSBA provided accurate camera poses and correct reconstruction. Loop-closing optimization and smooth trajectory assumption were not used.

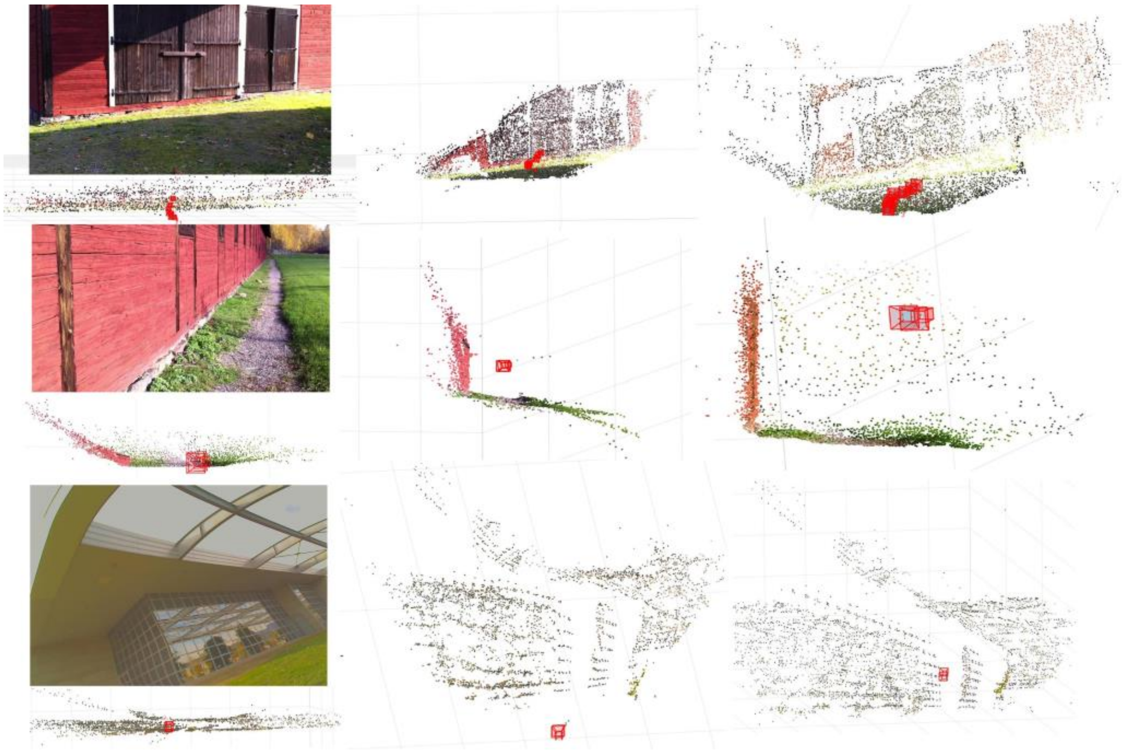


FIGURE 4.10: SfM with unordered real RS images. Input images **(left)** are with similar read-out direction and M-RSBA reconstructions are below. Results of GSBA **(middle)** and C-RSBA **(right)** are compared together. Obviously, M-RSBA suffers from planar degeneracy, while significant deformations can also be observed in GSBA reconstructions. However, C-RSBA provides correct reconstructed 3D scene.

section. 4.5.2 and the results of simulation experiments. GSBA suffers from distorted reconstruction. We can observe that the more strong distortion in RS image, the more deformations after SfM. It is important to realize that M-RSBA can not handle the case where input RS images with small scanning direction angles (down from 60 degrees). Strong deformations close to a plane were observed in 3D scene reconstructed with M-RSBA. Quite the contrary, C-RSBA provides significantly better reconstructions than GSBA and M-RSBA which collapse into degeneracy.

We also tested the whole 3-step pipeline on two a benchmarks [Sturm et al., 2012] in man-made indoor environment with available ground truth. Fig. 4.11 demonstrates that estimated trajectories by using our proposed approach are much closer to ground truth than by GS BA method.

4.7 Discussion and Conclusion

In this chapter, we propose a novel C-RSBA, which can successfully avoid planar degeneracy without any constraint on read out direction as in existing approaches. Note that image capture style with similar read-out directions are extremely natural and common in real applications while requirements of two distinct read-out directions will strongly limit the application range. Experiments with both real and synthetic data prove that the proposed method outperforms existing ones and can handle degeneracies pointed out in the literature. C-RSBA was successfully used as a final step in RSSfM pipeline. We believe that this work will help to take an extra step toward the use of RS cameras in SfM

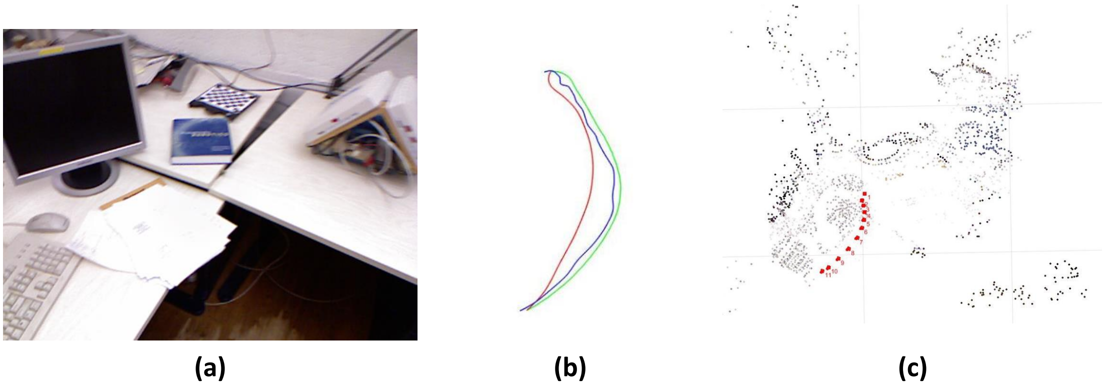


FIGURE 4.11: (a)Example RS frame. (b) Comparison of proposed estimated trajectories by 3-steps algorithm (blue) and GS BA (red) against to ground-truth (green). (d) Reconstructed 3D scene.

applications. Finally, since it can handle very strong RS effects, the proposed method can also be seen as a monocular instantaneous-speed measurement technique.

Chapter 5

RS Homography and its Applications

5.1 Introduction

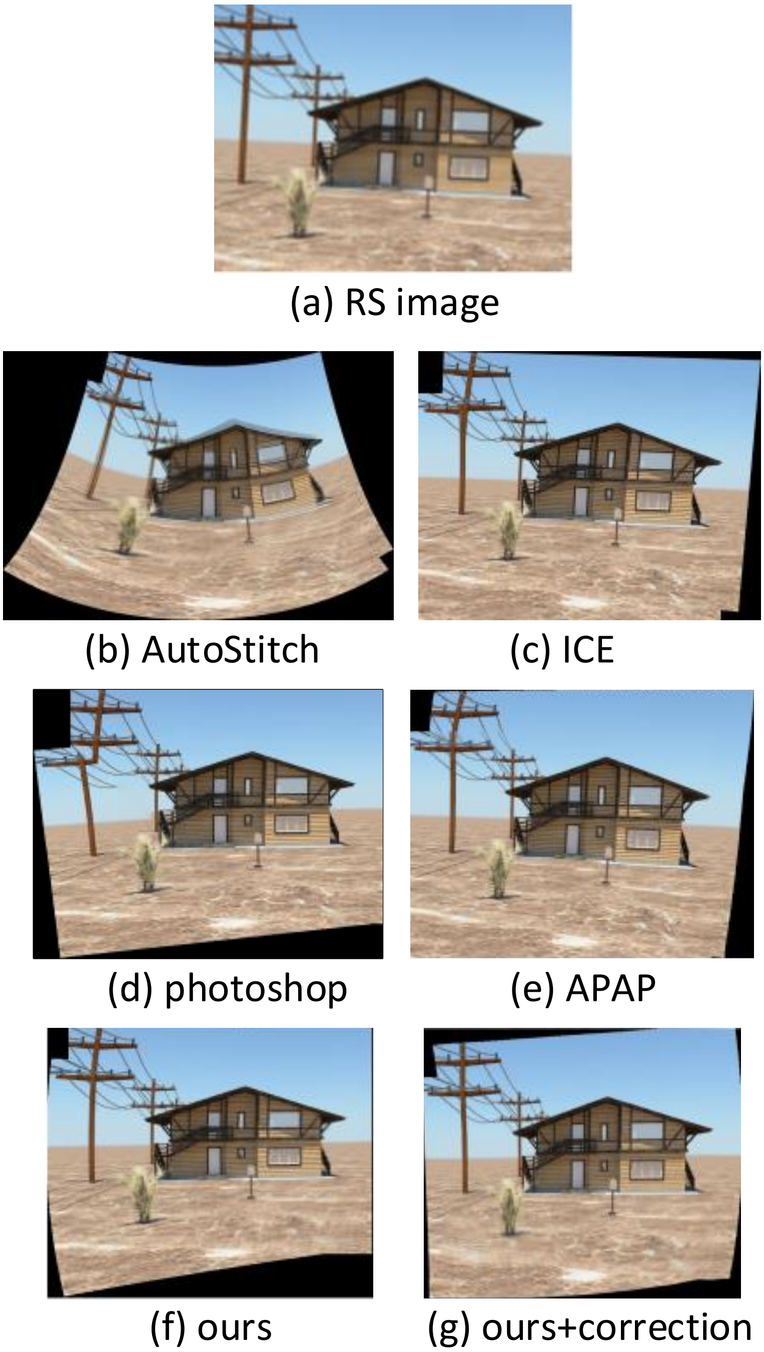
Homography is one of the important concepts in 3D vision and has been studied for a long period. However, all the existing methods are only applicable to conventional GS cameras. Homography for RS cameras, has not been addressed before.

In this chapter, we investigate the computation of the homography matrix based on correspondences from a RS pair. We show that at least 36 point correspondences are needed in theory to compute such a matrix linearly, and then we derive a practical method which works with 13 correspondences. In addition, we present two essential applications in computer vision that use RS homography: *1)* Plane-based relative pose estimation and *2)* image stitching. We experimentally show that the proposed methods outperform state-of-the-art techniques as well as well-known commercial applications, basing on many synthetic and real datasets.

5.2 Related Work and Motivation

Estimating the camera motion by using point correspondences is one of the most studied minimal problems in computer vision. For example, with GS, at least 3 point matches are needed to estimate the absolute pose [Gao et al., 2003], while at least 5 are needed to recover the relative pose between two calibrated GS views [Nistér, 2004]. Given the higher complexity of RS projection model [Meingast et al., 2005], more points are commonly needed. Methods for structure and motion estimation with RS images can be grouped into two categories (summarized in table. 5.1): optical flow and epipolar geometry.

Optical Flow methods: In [Zhuang et al., 2017], 8pt and 9pt linear solvers were developed to recover the relative pose of a RS camera that undergoes constant velocity and acceleration motion respectively. Unfortunately, consistency between the camera poses and their motion only holds with high-frame rates and smooth movements. In addition to the resulting high computation load, unordered image sets can not be processed.



(a) RS image

(b) AutoStitch

(c) ICE

(d) photoshop

(e) APAP

(f) ours

(g) ours+correction

FIGURE 5.1: RS images (a). Stitching results obtained with well-known commercial stitching applications such as AutoStitch [Brown and Lowe, 2007] (b) Microsoft Image Composite Editor (ICE) [ICE,] (c) Adobe Photoshop [pho,] (d) state-of-the-art multiple homographies stitching method APAP [Zaragoza et al., 2014] (e). The stitching results and the correction of the RS effects obtained with the the proposed method are shown in (f) and (g).

TABLE 5.1: Comparison of properties of existing RS relative pose solvers and the proposed solvers based on RS homography (**RSH**) in this chapter.

	GS homography	**RS epipolar** **[Dai et al., 2016]**	**Differential RS geometry** **SfM [Zhuang et al., 2017]**	**Full RSH** **homography**	**Simplified RSH** **homography**	**pure rotate RSH** **homography**
Small-motion assumption			✓			
Pure rotation	✓		✓	✓	✓	✓
Number of points (linear solution)	4	44	8	35.5	13	13
Number of points (nonlinear solution)	4	17	8	10	10	4

Epipolar Geometry: In multi-view reconstruction, many common configurations become critical with RS cameras and lead to reconstruction ambiguities. Authors in [Albl et al., 2016b] provide mathematical analysis for configurations with one, two or more views. They provide practical recipes on how to photograph with RS cameras to avoid reconstruction errors. This method can be used to unblock some situations but it is not a solution to the standard general SfM problem. Authors in [Dai et al., 2016] introduce 20pt and 44pt linear solvers for pure translational and uniform motion models respectively. However, the pure translational motion assumption is not feasible to model the camera motion in most of practical applications. Although more general, the 44-point solution requires too many correspondences and is therefore not suitable for use with RANSAC (Random Sample Consensus).

In summary, estimating relative pose of RS cameras remains an open problem and there is a need for new methods which require less input data (i.e. number of matches) and which work for various camera configurations. With some acceptable constraints on the scene structure or on the camera motion, homography could be used instead of epipolar geometry to recover the relative pose with less point matches [Hartley and Zisserman, 2003,Zhou et al., 2012,Saurer et al., 2017]. It has many applications namely image rectification, image registration and plane-based camera relative pose estimation. However, none of previous work has attempted to adapt the homography to the RS case. Authors in [Forssén and Ringaby, 2010,Grundmann et al., 2012,Liu et al., 2013,Vasu et al., 2018] try to solve the RS correction problem by building multiple independent homographies between each of image rows or row-blocks.These parameterized homographies are actually local GS homographies. In order to avoid discontinuities across rows (blocks), complex functions are used to smoothly interpolate the homographies. But the major issue of this approach is that points on a given row (row-block) in the first image have to be matched with points which also belong to a row (row-block) in the second image. This obviously limits the number of matches even with small inter-frame motion.

In this chapter, we address the problem of computing the homography from two RS images. We first propose a theoretical 36pt linear solution and then derive a practical 13pt minimal solver that gives good estimates of the geometry between two RS views. We also investigate the use of the proposed method for two major computer vision applications:

Plane-based Relative Pose Estimation: Although the RS relative pose problem has been addressed before [Dai et al., 2016, Zhuang et al., 2017], a more efficient and robust solution is proposed by using RS homography in this paper. This solution can also be used in plane-based RS SfM and SLAM.

Image Stitching: Users nowadays frequently create panoramas from videos by rotating RS cameras, e.g. 'Pano' mode in iPhone. Besides, 360 VR images such as Street View could also be taken with RS cameras installed on a moving platform such as the Google car [Klingner et al., 2013]. As shown in Fig. 5.1(b-e), the most well-known commercial stitching software or state-of-the-art methods, which are based on the GS model, lead to poor results in the presence of RS effects. Therefore, designing a RS stitching method by using RS homography will be of valuable significance.

5.2.1 Chapter outlines

The rest of the chapter is organized as follows:

- Then we derive the full and the simplified RS homography matrices in section 5.4 followed by the solutions for the RS homography matrix computation from point matches in section 5.5.
- Next we show how to estimate the RS relative pose basing on a planar scene in section 5.6 and also RS image stitching by using RS homography in section 5.7.
- Finally, we present experimental results obtained thanks to the proposed methods in section 5.8.

5.3 GS Homography

Let us assume that a planar object is observed from two GS cameras at the poses $[\mathbf{I}|\mathbf{0}]$ and $[\mathbf{R}_0|\mathbf{t}_0]$. The transformation between the two corresponding image points $\mathbf{q}_i$ and $\mathbf{q'}_i$ can be written as:

$$\alpha_i \mathbf{q'}_i = \mathbf{H}\mathbf{q}_i = (\mathbf{R}_0 - \frac{\mathbf{t}_0\mathbf{n}^\top}{d})\mathbf{q}_i, \qquad \alpha_i = z'/z \tag{5.1}$$

where α_i is a scale factor that depends on the depth of $\mathbf{P}_i$ in each camera. $\mathbf{H}_{GS}$ is the 3×3 GS homography matrix, $\mathbf{n}$ is the normal vector of the observed plane and d is the distance from the first camera to the plane under the constraint $\mathbf{n}^\top\mathbf{P}_i + d = 0$. Note that we assume that each of the two cameras is calibrated. Thus, the normalized image point q_i can be obtained by multiplying image measurement point m_i by $\mathbf{K}^{-1}$.

5.4 RS Homography

5.4.1 RS Relative Pose

Let us consider n 3D points $\mathbf{P}_i$ imaged by two RS cameras at poses $[\mathbf{R}_{v_i}|\mathbf{t}_{v_i}]$ and $[\mathbf{R}_{v'_i}|\mathbf{t}_{v'_i}]$, as $\mathbf{q}_i = [u_i, v_i, 1]^\top$ and $\mathbf{q'}_i = [u'_i, v'_i, 1]^\top$ in the two images respectively. $[\mathbf{I}|\mathbf{0}]$ and $[\mathbf{R}_0|\mathbf{t}_0]$ are the camera poses of the first row of each image. Thus, the rotation $\mathbf{R}_i$ and the translation $\mathbf{t}_i$ between the the row v_i in the first image and the row v'_i in the second image are:

$$\begin{aligned}\mathbf{R}_i &= (\mathbf{I} + [\boldsymbol{\omega}_2]_\times v'_i)\mathbf{R}_0(\mathbf{I} - [\boldsymbol{\omega}_1]_\times v_i) \\ \mathbf{t}_i &= \mathbf{t}_0 + \mathbf{d}_2 v'_i - (\mathbf{I} + [\boldsymbol{\omega}_2]_\times v'_i)\mathbf{R}_0(\mathbf{I} - [\boldsymbol{\omega}_1]_\times v_i)\mathbf{d}_1 v_i\end{aligned} \tag{5.2}$$

where $\{\boldsymbol{\omega}_1, \mathbf{d}_1\}$ and $\{\boldsymbol{\omega}_2, \mathbf{d}_2\}$ are instantaneous-motion parameters of the two RS cameras.

Proof of Eq. (5.2). We assume a 3D point $\mathbf{P}$ in world coordinates is expressed as $\mathbf{P}_{v_i}$ and $\mathbf{P}_{v'_i}$ in the two camera coordinate systems. The transformations are written as:

$$\mathbf{P}_{v_i} = \mathbf{R}_{v_i}\mathbf{P} + \mathbf{t}_{v_i} \tag{5.3}$$

and

$$\mathbf{P}_{v'_i} = \mathbf{R}_{v'_i}\mathbf{P} + \mathbf{t}_{v'_i} \tag{5.4}$$

By substituting Eq. (5.3) into Eq. (5.4) and eliminating $\mathbf{P}$, we obtain the transformation between $\mathbf{P}_{v_i}$ and $\mathbf{P}_{v'_i}$ as:

$$\begin{aligned}\mathbf{P}_{v'_i} &= \mathbf{R}_{v'_i}\mathbf{P} + \mathbf{t}_{v'_i} = \mathbf{R}_{v'_i}(\mathbf{R}^\top_{v_i}(\mathbf{P}_{v_i} - \mathbf{t}_{v_i})) + \mathbf{t}_{v'_i} \\ &= \underbrace{\mathbf{R}_{v'_i}\mathbf{R}^\top_{v_i}}_{\mathbf{R}_i}\mathbf{P}_{v_i} + \underbrace{\mathbf{t}_{v'_i} - \mathbf{R}_{v'_i}\mathbf{R}^\top_{v_i}\mathbf{t}_{v_i}}_{\mathbf{t}_i}\end{aligned} \tag{5.5}$$

Thus, the relative pose can be expressed as:

$$\mathbf{R}_i = \mathbf{R}_{v'_i}\mathbf{R}^\top_{v_i} \qquad \mathbf{t}_i = \mathbf{t}_{v'_i} - \mathbf{R}_{v'_i}\mathbf{R}^\top_{v_i}\mathbf{t}_{v_i} \tag{5.6}$$

Then we substitute Eq (2.20) into Eq. (5.5), and we obtain:

$$\begin{aligned}\mathbf{R}_i &= (\mathbf{I} + [\boldsymbol{\omega}_2]_\times v'_i)\mathbf{R}_0(\mathbf{I} - [\boldsymbol{\omega}_1]_\times v_i) \\ &= \mathbf{R}_0 - \mathbf{R}_0[\boldsymbol{\omega}_1]_\times v_i + [\boldsymbol{\omega}_2]_\times\mathbf{R}_0 v'_i \quad - [\boldsymbol{\omega}_2]_\times\mathbf{R}_0[\boldsymbol{\omega}_1]_\times v_i v'_i \\ \\ \mathbf{t}_i &= \mathbf{t}_0 + \mathbf{d}_2 v'_i - (\mathbf{I} + [\boldsymbol{\omega}_2]_\times v'_i)\mathbf{R}_0(\mathbf{I} - [\boldsymbol{\omega}_1]_\times v_i)\mathbf{d}_1 v_i \\ &= \mathbf{t}_0 - \mathbf{R}_0\mathbf{d}_1 v_i + \mathbf{d}_2 v'_i + \mathbf{R}_0[\boldsymbol{\omega}_1]_\times\mathbf{d}_1 v_i^2 \quad - [\boldsymbol{\omega}_2]_\times\mathbf{R}_0\mathbf{d}_1 v_i v'_i + [\boldsymbol{\omega}_2]_\times\mathbf{R}_0[\boldsymbol{\omega}_1]_\times\mathbf{d}_1 v_i^2 v'_i\end{aligned} \tag{5.7}$$

5.4.2 RS Homography Matrix

When instantaneous-motion occurs during the acquisition, $\mathbf{H}$ between two RS cameras varies with different row combinations. The relative pose between row v_i in the first image and the row v'_i in the second image is defined in Eq. (5.2). Similarly the plane normal and the distance to the plane are also changing dynamically with different row indexes. By using linear instantaneous-motion model, we can express the normal vector and distance to the plane w.r.t row v_i as:

$$\begin{aligned}\mathbf{n}_i^\top &= \mathbf{n}_0^\top(\mathbf{I} - [\boldsymbol{\omega}_1]_\times v_i) \\ d_i &= d_0 - \mathbf{n}_0^\top(\mathbf{I} - [\boldsymbol{\omega}_1]_\times v_i)\mathbf{d}_1 v_i\end{aligned} \tag{5.8}$$

where $\mathbf{n}_0$ and d_0 are the normal vector and the distance for the first row.

By substituting Eq. (5.2) and (5.8) into Eq. (5.1), we obtain the expression of RS homography matrix as:

$$\mathbf{H}_{RS,i} = \mathbf{H}_{GS} + \mathbf{H}_1 v_i + \mathbf{H}_2 v'_i + \mathbf{H}_3 v_i v'_i + \mathbf{H}_4 v_i^2 + \mathbf{H}_5 v_i^2 v'_i + \mathbf{H}_6 v_i^3 + \mathbf{H}_7 v_i^3 v'_i \tag{5.9}$$

where $\mathbf{H}_{GS}, \mathbf{H}_1 ... \mathbf{H}_7$ are 3×3 atomic matrices.

Proof of Eq. (5.8) When the RS camera is at the pose of its first row, the plane constraint is:

$$\mathbf{n}_0^\top\mathbf{P}_i + d_0 = 0 \tag{5.10}$$

By substituting the transform from $\mathbb{P}$ to $\mathbf{P}_{v_i}$ in Eq. (5.3) into Eq. (5.10), we obtain:

$$\mathbf{n}_0^\top(\mathbf{R}^\top_{v_i}(\mathbf{P}_{v_i} - \mathbf{t}_{vi})) + d_0 = \underbrace{\mathbf{n}_0^\top\mathbf{R}^\top_{v_i}}_{\mathbf{n}^\top_{v_i}}\mathbf{P}_{v_i} + \underbrace{d_0 - \mathbf{n}_0^\top\mathbf{R}^\top_{v_i}\mathbf{t}_{v_i}}_{d_{v_i}} = 0 \tag{5.11}$$

By substituting Eq. (2.20) into Eq. (5.11), we finally obtain Eq. (5.8).

Proof of Eq. (5.9) For convenience, we denote $\mathbf{R}_i$, $\mathbf{t}_i$ in Eq. (5.2) and $\mathbf{n}_i^\top$, d_i in Eq. (5.8) as:

$$\begin{aligned} \mathbf{R}_i &= \mathbf{R}_0 + \mathbf{R}_1 v_i + \mathbf{R}_2 v_i' + \mathbf{R}_3 v_i v_i' \\ \mathbf{t}_i &= \mathbf{t}_0 + \mathbf{t}_1 v_i + \mathbf{t}_2 v_i' + \mathbf{t}_3 v_i^2 + \mathbf{t}_4 v_i v_i' + \mathbf{t}_5 v_i^2 v_i' - \frac{\mathbf{n}_i^\top}{d_i} \\ &\approx -\frac{\mathbf{n}_0^\top - \mathbf{n}_0^\top [\boldsymbol{\omega}_1]_\times v_i}{d_0} = \mathbf{N}_0 + \mathbf{N}_1 v_i \end{aligned} \tag{5.12}$$

where,

$$\begin{cases} \mathbf{R}_0 = \mathbf{R}_0 \\ \mathbf{R}_1 = -\mathbf{R}_0[\boldsymbol{\omega}_1]_\times \\ \mathbf{R}_2 = [\boldsymbol{\omega}_2]_\times \mathbf{R}_0 \\ \mathbf{R}_3 = -[\boldsymbol{\omega}_2]_\times \mathbf{R}_0 [\boldsymbol{\omega}_1]_\times \end{cases} \begin{cases} \mathbf{t}_0 = \mathbf{t}_0 \\ \mathbf{t}_1 = -\mathbf{R}_0 \mathbf{d}_1 \\ \mathbf{t}_2 = \mathbf{d}_2 \\ \mathbf{t}_3 = \mathbf{R}_0[\boldsymbol{\omega}_1]_\times \mathbf{d}_1 \\ \mathbf{t}_4 = -[\boldsymbol{\omega}_2]_\times \mathbf{R}_0 \mathbf{d}_1 \\ \mathbf{t}_5 = [\boldsymbol{\omega}_2]_\times \mathbf{R}_0 [\boldsymbol{\omega}_1]_\times \mathbf{d}_1 \end{cases} \begin{cases} \mathbf{N}_0 = -\frac{\mathbf{n}_0^\top}{d_0} \\ \mathbf{N}_1 = \frac{\mathbf{n}_0^\top [\boldsymbol{\omega}_1]_\times}{d_0} \end{cases} \tag{5.13}$$

By substituting Eq. 5.12 into Eq.5.1, we can obtain Eq. (5.9):

$$\begin{aligned} \mathbf{H}_{RS} &= \mathbf{R}_i - \frac{\mathbf{t}_i \mathbf{n}_i^\top}{d_i} \\ &= (\mathbf{R}_0 + \mathbf{R}_1 v_i + \mathbf{R}_2 v_i' + \mathbf{R}_3 v_i v_i') \\ &\quad + (\mathbf{t}_0 + \mathbf{t}_1 v_i + \mathbf{t}_2 v_i' + \mathbf{t}_3 v_i^2 + \mathbf{t}_4 v_i v_i' + \mathbf{t}_5 v_i^2 v_i')(\mathbf{N}_0 + \mathbf{N}_1 v_i) \\ &= \underbrace{(\mathbf{R}_0 + \mathbf{t}_0 \mathbf{N}_0)}_{\mathbf{H}_{GS}} + \underbrace{(\mathbf{R}_1 + \mathbf{t}_1 \mathbf{N}_0 + \mathbf{t}_0 \mathbf{N}_1)}_{\mathbf{H}_1} v_i + \underbrace{(\mathbf{R}_2 + \mathbf{t}_2 \mathbf{N}_0)}_{\mathbf{H}_2} v_i' \\ &\quad + \underbrace{(\mathbf{R}_3 + \mathbf{t}_4 \mathbf{N}_0 + \mathbf{t}_2 \mathbf{N}_1)}_{\mathbf{H}_3} v_i v_i' + \underbrace{(\mathbf{t}_3 \mathbf{N}_0 + \mathbf{t}_1 \mathbf{N}_1)}_{\mathbf{H}_4} v_i^2 \\ &\quad + \underbrace{(\mathbf{t}_5 \mathbf{N}_0 + \mathbf{t}_4 \mathbf{N}_1)}_{\mathbf{H}_5} v_i^2 v_i' + \underbrace{(\mathbf{t}_3 \mathbf{N}_1)}_{\mathbf{H}_6} v_i^3 + \underbrace{(\mathbf{t}_5 \mathbf{N}_1)}_{\mathbf{H}_7} v_i^3 v_i' \end{aligned} \tag{5.14}$$

5.4.3 Simplified RS Homography Matrix

5.4.3.1 Approximation of RS Relative Pose.

Under the small rotation assumption, the second and higher order terms in Eq. (5.2) can be ignored. This simplification is also used in [Albl et al., 2015, Ito and Okatani, , Purkait and Zach, 2017, Lao et al., 2018a]. This approximation can be justified in that we force the translational speed vectors $\mathbf{d}_1$ and $\mathbf{d}_2$ to be constant in the world coordinate system, which is physically coherent with the constant velocity kinematic model. Therefore, we obtain an approximate expression of RS relative pose:

$$\begin{aligned} \mathbf{R}_i &= \mathbf{R}_0 - \mathbf{R}_0[\boldsymbol{\omega}_1]_\times v_i + [\boldsymbol{\omega}_2]_\times \mathbf{R}_0 v_i' \\ \mathbf{t}_i &= \mathbf{t}_0 + \mathbf{d}_2 v_i' - \mathbf{R}_0 \mathbf{d}_1 v_i \end{aligned} \tag{5.15}$$

5.4.3.2 Approximation of the Plane Pose.

In practice, since the translation during acquisition is commonly much smaller than the distance from the camera to the scene plane, we can ignore the terms affected by translational velocities. In addition, we drop the second order terms, and obtain the approximate expressions:

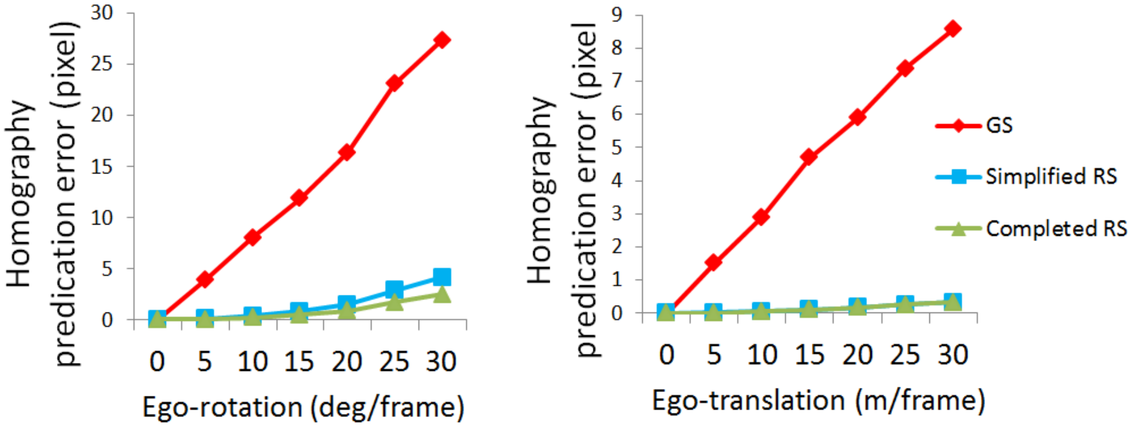


FIGURE 5.2: Comparison of the GS, completed RS and simplified RS homographies.

$$\begin{aligned} \mathbf{n}_i^\top &= \mathbf{n}_0^\top(\mathbf{I} - [\boldsymbol{\omega}_1]_\times v_i) \\ d_i &= d_0 - \mathbf{n}_0^\top \mathbf{d}_1 v_i \approx d_0 \end{aligned} \tag{5.16}$$

5.4.3.3 Derivation of the Simplified RS Homography Matrix

Using both approximations in Eq. (5.15) and (5.16), the RS homography matrix $\mathbf{H}_{RS}$ between the row v_i and the row v'_i in the two images can be simplified as follows:

$$\begin{aligned} \mathbf{H}_{RS,i} =& \mathbf{H}_{GS} + \mathbf{A}_1 v_i + \mathbf{A}_2 v'_i \\ \text{where,} \quad \mathbf{A}_1 =& -\mathbf{R}_0[\boldsymbol{\omega}_1]_\times + \frac{\mathbf{R}_0\mathbf{d}_1\mathbf{n}_0^\top}{d_0} + \frac{\mathbf{t}_0\mathbf{n}_0^\top[\boldsymbol{\omega}_1]_\times}{d_0} \\ \mathbf{A}_2 =& [\boldsymbol{\omega}_2]_\times \mathbf{R}_0 - \frac{\mathbf{d}_2\mathbf{n}_0^\top}{d_0} \end{aligned} \tag{5.17}$$

$\mathbf{A}_1$ and $\mathbf{A}_2$ are two atomic matrices. Note that the RS homography matrix consists of the GS homography matrix $\mathbf{H}_{GS}$ defined in Eq. (5.1) and of the two matrices $\mathbf{A}_1$, $\mathbf{A}_2$ which contain the instantaneous-motion parameters. For readability, we denote $\mathbf{H}_{RS}$ as $\mathbf{H}$ in the rest of the paper.

Proof of the Eq. (5.17). By substituting Eq. (5.15) and (5.16) into Eq. (5.9), we have:

$$\begin{aligned} \mathbf{H}_{RS} =& \mathbf{R}_i - \frac{\mathbf{t}_i\mathbf{n}_i^\top}{d_i} \\ =& \mathbf{R}_0 - \mathbf{R}_0[\boldsymbol{\omega}_1]_\times v_i + [\boldsymbol{\omega}_2]_\times \mathbf{R}_0 v'_i - \\ & \frac{\mathbf{t}_0\mathbf{n}_0^\top + \mathbf{d}_2\mathbf{n}_0^\top v'_i - \mathbf{R}_0\mathbf{d}_1\mathbf{n}_0^\top v_i - \mathbf{t}_0\mathbf{n}_0^\top[\boldsymbol{\omega}_1]_\times v_i}{d_0} \\ =& (\mathbf{R}_0 - \frac{\mathbf{t}_0\mathbf{n}_0^\top}{d}) + (-\mathbf{R}_0[\boldsymbol{\omega}_1]_\times + \frac{\mathbf{R}_0\mathbf{d}_1\mathbf{n}_0^\top}{d_0} + \frac{\mathbf{t}_0\mathbf{n}_0^\top[\boldsymbol{\omega}_1]_\times}{d_0})v_i + ([\boldsymbol{\omega}_2]_\times \mathbf{R}_0 - \frac{\mathbf{d}_2\mathbf{n}_0^\top}{d_0})v'_i \\ =& \mathbf{H}_{GS} + \mathbf{A}_1 v_i + \mathbf{A}_2 v'_i \end{aligned} \tag{5.18}$$

5.4.4 Comparisons of Full Model and Simplified Model

The GS homography and three RS homographies are summarized into table. 5.1. We propose to use simplified model instead of completed model for RS relative pose and instantaneous-motions estimation for the following reasons:

1. Both [Dai et al., 2016] and complete RS homography, which require 44 and 36 point matches respectively, are intractable to use in RANSAC estimation. In contrast, simplified RS model requires significantly less point matches (13 in total).

2. In practical applications, the simplified RS homography provides similar accuracy compared to the complete RS homography. A simulated experiment is conducted to justify the validity of the proposed simplified RS homography model. We simulated 60 feature points (within a planar object) and observed by two RS cameras positioned randomly on a sphere of radius of 100m. Then the mean projection errors (Euclidean distance) between $\mathbf{H}q_i$ and corresponding q'_i, where $\mathbf{H}$ are calculated by using GS-based homography $\mathbf{H}_{GS}$, complete RS homography $\mathbf{H}_RS$ (in Eq. (5.9)) and simplified $\mathbf{H}_{RS}$ (in Eq. (5.17)) respectively. We varied the norm value of camera rotational velocities from 0 to 30 deg/frame and translational velocities from 0 to 30 m/frame, which are both even too high to achieve in real applications. Results in Fig. 5.2 show that the simplified RS homography achieves similar accuracy to the completed RS homography model, while being much better compared to the GS one.

Base on the comparison above, we believe that simplified RS model is with higher practicality than the complete one.

5.5 RS Homography Estimation

5.5.1 4pt GS Homography Estimation

The homography matrix is usually computed using the Direct Linear Transform algorithm (DLT). From Eq. (5.1) we obtain $\mathbf{q}'_{v_i} \times \mathbf{H}_{GS}\mathbf{q}_i = \mathbf{0}$. This gives two linearly independent equations:

$$\begin{bmatrix} \mathbf{0}^\top & -\mathbf{q}_i^\top & v'_i\mathbf{q}_i^\top \\ \mathbf{q}_i^\top & \mathbf{0}^\top & -u'_i\mathbf{q}_i^\top \end{bmatrix} \begin{bmatrix} \mathbf{H}^\top_{GS,(1)} \\ \mathbf{H}^\top_{GS,(2)} \\ \mathbf{H}^\top_{GS,(3)} \end{bmatrix} = \mathbf{L}_i\mathbf{h}_{GS} = \mathbf{0} \tag{5.19}$$

where $\mathbf{0}^\top = [0,0,0]$ and $\mathbf{H}_{GS,(i)}$ is the i^{th} row of $\mathbf{H}_{GS}$. Given n point correspondences ($n \geqslant 4$), we obtain a system in the form $\mathbf{Lh}_{GS} = 0$ where $\mathbf{L}$ is a $2n \times 9$ matrix. The solution is then the singular vector associated to the smallest singular value of $\mathbf{L}$.

5.5.2 36pt Full RS Homography Matrix Estimation

By setting $\mathbf{H}_{GS,33} = 1$ and substituting Eq. (5.9) into $\mathbf{h}$, Eq. (5.19) can be rewritten as follows:

$$\mathbf{M}_{RS,i}\mathbf{h}_{RS} = \begin{bmatrix} \mathbf{0} & \mathbf{q}_i \\ -\mathbf{q}_i & \mathbf{0} \\ v'_i[u_i, v_i]^\top & -u'_i[u_i, v_i] \\ \mathbf{0} & v_i\mathbf{q}_i \\ -v_i\mathbf{q}_i & \mathbf{0} \\ v_iv'_i\mathbf{q}_i & -v_iu'_i\mathbf{q}_i \\ \mathbf{0} & v'_i\mathbf{q}_i \\ -v'_i\mathbf{q}_i & \mathbf{0} \\ {v'_i}^2\mathbf{q}_i & -u'_iv'_i\mathbf{q}_i \\ \mathbf{0} & v_iv'_i\mathbf{q}_i \\ -v_iv'_i\mathbf{q}_i & \mathbf{0} \\ v_i{v'_i}^2\mathbf{q} & -u'_iv_iv'_i\mathbf{q}_i \\ \mathbf{0} & {v_i}^2\mathbf{q}_i \\ -{v_i}^2\mathbf{q}_i & \mathbf{0} \\ {v_i}^2v'_i\mathbf{q}_i & -u'_i{v_i}^2\mathbf{q}_i \\ \mathbf{0} & {v_i}^2v'_i\mathbf{q} \\ -{v_i}^2v'_i\mathbf{q}_i & \mathbf{0} \\ {v_i}^2{v'_i}^2\mathbf{q}_i & -u'_i{v_i}^2v'_i\mathbf{q}_i \\ \mathbf{0} & {v_i}^3\mathbf{q}_i \\ -{v_i}^3\mathbf{q}_i & \mathbf{0} \\ {v_i}^3v'_i\mathbf{q} & -u'_i{v_i}^3\mathbf{q}_i \\ \mathbf{0} & {v_i}^3v'_i\mathbf{q}_i \\ -{v_i}^3v'_i\mathbf{q}_i & \mathbf{0} \\ {v_i}^3{v'_i}^2\mathbf{q}_i & -u'_i{v_i}^3v'_i\mathbf{q}_i \end{bmatrix}^\top \begin{bmatrix} {\mathbf{H}_{GS,(1)}}^\top \\ {\mathbf{H}_{GS,(2)}}^\top \\ H_{GS,31} \\ H_{GS,32} \\ {\mathbf{H}_{1,(1)}}^\top \\ {\mathbf{H}_{1,(2)}}^\top \\ {\mathbf{H}_{1,(3)}}^\top \\ {\mathbf{H}_{2,(1)}}^\top \\ {\mathbf{H}_{2,(2)}}^\top \\ {\mathbf{H}_{2,(3)}}^\top \\ {\mathbf{H}_{3,(1)}}^\top \\ {\mathbf{H}_{3,(2)}}^\top \\ {\mathbf{H}_{3,(3)}}^\top \\ {\mathbf{H}_{4,(1)}}^\top \\ {\mathbf{H}_{4,(2)}}^\top \\ {\mathbf{H}_{4,(3)}}^\top \\ {\mathbf{H}_{5,(1)}}^\top \\ {\mathbf{H}_{5,(2)}}^\top \\ {\mathbf{H}_{5,(3)}}^\top \\ {\mathbf{H}_{6,(1)}}^\top \\ {\mathbf{H}_{6,(2)}}^\top \\ {\mathbf{H}_{6,(3)}}^\top \\ {\mathbf{H}_{7,(1)}}^\top \\ {\mathbf{H}_{7,(2)}}^\top \\ {\mathbf{H}_{7,(3)}}^\top \end{bmatrix} = \mathbf{b}_i = \begin{bmatrix} -v_i \\ u_i \end{bmatrix} \tag{5.20}$$

where $\mathbf{M}_{RS,i}$ is a 2×71 matrix while $\mathbf{b}_i$ is a 2×1 vector. $\mathbf{h}_{RS}$ is a 71×1 vector. $\mathbf{0} = [0, 0, 0]^\top$, $\mathbf{H}_{(i)}$, $\mathbf{A}_{1,(i)}$ and $\mathbf{A}_{2,(i)}$ are the i^{th} rows of $\mathbf{H}_{GS}$, $\mathbf{A}_1$ and $\mathbf{A}_2$ respectively.

Each point correspondence gives two constraints in Eq. (5.20). Thus, we obtain a system in the form $\mathbf{M}_{RS}\mathbf{h}_{RS} = \mathbf{b}$ which is solved using SVD.

5.5.3 13pt Simplified RS Homography Matrix Estimation

In the simplified case of section. 5.4.3, by setting $\mathbf{H}_{GS,33} = 1$ and substituting Eq. (5.17) into $\mathbf{h}_{GS}$, Eq. (5.19) can be rewritten as:

$$\begin{bmatrix} \mathbf{0} & \mathbf{q}_i \\ -\mathbf{q}_i & \mathbf{0} \\ v'_i[u_i, v_i]^\top & -u'_i[u_i, v_i] \\ \mathbf{0} & v_i\mathbf{q}_i \\ -v_i\mathbf{q}_i & \mathbf{0} \\ v_iv'_i\mathbf{q}_i & -v_iu'_i\mathbf{q}_i \\ \mathbf{0} & v'_i\mathbf{q}_i \\ -v'_i\mathbf{q}_i & \mathbf{0} \\ {v'_i}^2\mathbf{q}_i & -u'_iv'_i\mathbf{q}_i \end{bmatrix}^\top \begin{bmatrix} {\mathbf{H}_{GS,(1)}}^\top \\ {\mathbf{H}_{GS,(2)}}^\top \\ H_{GS,31} \\ H_{GS,32} \\ {\mathbf{A}_{1,(1)}}^\top \\ {\mathbf{A}_{1,(2)}}^\top \\ {\mathbf{A}_{1,(3)}}^\top \\ {\mathbf{A}_{2,(1)}}^\top \\ {\mathbf{A}_{2,(2)}}^\top \\ {\mathbf{A}_{2,(3)}}^\top \end{bmatrix} = \mathbf{M}_{RS,i}\mathbf{h}_{RS} = \mathbf{b}_i = \begin{bmatrix} -v_i & u_i \end{bmatrix}^\top \tag{5.21}$$

where $\mathbf{M}_{RS,i}$ reduces to a 2×26 matrix and $\mathbf{b}_{RS,i}$ is a 26×1 vector. $\mathbf{h}$ is a 26×1 vector with 26 unknowns. Thus, with only 13 point correspondences, we can estimate $\mathbf{h}_{RS}$ linearly by using SVD. In order to obtain stable results, we perform a normalization of $\mathbf{M}_{RS}$ and $\mathbf{b}_{RS}$ in the way explained in [Hartley and Zisserman, 2003,Dai et al., 2016].

This 13pt minimal problem solver is used to extend the RANSAC pipeline [Fischler and Bolles, 1981] to the robust estimation of RS Homography with automatic matching.

Proof of the Eq. (5.21). By substituting Eq. (5.17) into $\mathbf{h}_{GS}$, the two linear independent equations in Eq. (5.19) can be rewritten as:

$$\mathbf{L}^{RS1^\top}\mathbf{h}^{RS1} = \begin{bmatrix} u_i v_i v'_i \\ u_i {v'_i}^2 \\ {v_i}^2 v'_i \\ v_i {v'_i}^2 \\ u_i v_i \\ u_i v'_i \\ v_i v'_i \\ {v_i}^2 \\ {v'_i}^2 \\ u_i \\ v_i \\ v'_i \\ 1 \end{bmatrix}^\top \begin{bmatrix} A_{1,31} \\ A_{2,31} \\ A_{1,32} \\ A_{2,32} \\ -A_{1,21} \\ -A_{2,21} + H_{GS,31} \\ -A_{2,22} + H_{GS,32} + A_{1,33} \\ -A_{1,22} \\ A_{2,33} \\ -H_{GS,21} \\ -H_{GS,22} - A_{1,23} \\ H_{GS,33} - A_{2,23} \\ -H_{GS,23} \end{bmatrix} = 0 \tag{5.22}$$

$$\mathbf{L}^{RS2^\top}\mathbf{h}^{RS2} = \begin{bmatrix} u_i u'_i v'_i \\ u'_i v_i v'_i \\ u_i u'_i v_i \\ u'_i {v_i}^2 \\ u_i v_i \\ u_i v'_i \\ u_i u'_i \\ u'_i v_i \\ u'_i v'_i \\ v_i v'_i \\ {v_i}^2 \\ u_i \\ u'_i \\ v_i \\ v'_i \\ 1 \end{bmatrix}^\top \begin{bmatrix} -A_{2,31} \\ -A_{2,32} \\ -A_{1,31} \\ -A_{1,32} \\ A_{1,11} \\ A_{2,11} \\ -H_{GS,31} \\ -H_{GS,32} - A_{1,33} \\ -A_{2,33} \\ A_{2,12} \\ A_{1,12} \\ H_{GS,11} \\ -H_{GS,33} \\ H_{GS,12} + A_{1,13} \\ A_{2,13} \\ H_{GS,13} \end{bmatrix} = 0 \tag{5.23}$$

where $M_{i,jk}$ is the component at j^{th} row and k^{th} column of the matrix $\mathbf{M}_i$. Without loss of generality, We can set $\mathbf{H}_{GS,33} = 1$ and rewrite Eq. (5.22) and (5.23) in a unified matrix form:

$$\underbrace{\begin{bmatrix} \mathbf{0} & \mathbf{q}_i \\ -\mathbf{q}_i & \mathbf{0} \\ v'_i\mathbf{q}_i & -u'_i\mathbf{q}_i \\ \mathbf{0} & v_i\mathbf{q}_i \\ -v_i\mathbf{q}_i & \mathbf{0} \\ v_iv'_i\mathbf{q}_i & -v_iu'_i\mathbf{q}_i \\ \mathbf{0} & v'_i\mathbf{q}_i \\ -v'_i\mathbf{q}_i & \mathbf{0} \\ v'^2_i\mathbf{q}_i & -u'_iv'_i\mathbf{q}_i \end{bmatrix}}_{2\times 27 \text{ matrix}}{}^{\top} \underbrace{\begin{bmatrix} \mathbf{H}_{(1)}{}^{\top} \\ \mathbf{H}_{(2)}{}^{\top} \\ H_{GS,31} \\ H_{GS,32} \\ 1 \\ \mathbf{A}_{1,(1)}{}^{\top} \\ \mathbf{A}_{1,(2)}{}^{\top} \\ \mathbf{A}_{1,(3)}{}^{\top} \\ \mathbf{A}_{2,(1)}{}^{\top} \\ \mathbf{A}_{2,(2)}{}^{\top} \\ \mathbf{A}_{2,(3)}{}^{\top} \end{bmatrix}}_{27\times 1 \text{ vector}} = \mathbf{0} \tag{5.24}$$

Eq. (5.24) can be solved directly using a DLT algorithm or rewritten in the non homogeneous form $\mathbf{Ax} = \mathbf{b}$ as follow:

$$\begin{bmatrix} \mathbf{0} & \mathbf{q}_i \\ -\mathbf{q}_i & \mathbf{0} \\ v'_i[u_i, v_i]^{\top} & -u'_i[u_i, v_i] \\ \mathbf{0} & v_i\mathbf{q}_i \\ -v_i\mathbf{q}_i & \mathbf{0} \\ v_iv'_i\mathbf{q}_i & -v_iu'_i\mathbf{q}_i \\ \mathbf{0} & v'_i\mathbf{q}_i \\ -v'_i\mathbf{q}_i & \mathbf{0} \\ v'^2_i\mathbf{q}_i & -u'_iv'_i\mathbf{q}_i \end{bmatrix}^{\top} \begin{bmatrix} \mathbf{H}_{GS,(1)}{}^{\top} \\ \mathbf{H}_{GS,(2)}{}^{\top} \\ H_{GS,31} \\ H_{GS,32} \\ \mathbf{A}_{1,(1)}{}^{\top} \\ \mathbf{A}_{1,(2)}{}^{\top} \\ \mathbf{A}_{1,(3)}{}^{\top} \\ \mathbf{A}_{2,(1)}{}^{\top} \\ \mathbf{A}_{2,(2)}{}^{\top} \\ \mathbf{A}_{2,(3)}{}^{\top} \end{bmatrix} = \begin{bmatrix} -v_i \\ u_i \end{bmatrix} \tag{5.25}$$

Thus, we obtain Eq. (5.21).

5.6 Plane-based RS Relative Pose and Instantaneous-motion Estimation

5.6.1 Relative pose

$[\mathbf{R}_0|\mathbf{t}_0]$ and plane normal vector $\mathbf{n}_0$: Once $\mathbf{H}_{GS}$ is known, it can be decomposed into $\mathbf{R}_0$, $\mathbf{t}_0$ and $\mathbf{n}_0$ by using SVD. $\mathbf{d}_0$ is set as 1 and absorbed by $\mathbf{t}_0$. Generally, this decomposition yields four solutions, where only one is physically meaningful considering the positive depth constraint [Ma et al., 2012, Malis and Vargas, 2007].

5.6.2 Instantaneous-motion:

We can further retrieve instantaneous-motion parameters of the cameras thanks to two linear equation systems derived from matrices $\mathbf{A}_1$ and $\mathbf{A}_2$:

1. First we compute $\boldsymbol{\omega}_1 = \{\omega^x_1, \omega^y_1, \omega^z_1\}$ and $\boldsymbol{d}_1 = \{d^x_1, d^y_1, d^z_1\}$ by using $[\mathbf{R}_0|\mathbf{t}_0]$ and $\mathbf{n}_0$ in matrix $\mathbf{A}_1$ (6 unknowns with 9 equations).

2. Then we extract $\boldsymbol{\omega}_2 = \{\omega^x_2, \omega^y_2, \omega^z_2\}$ and $\boldsymbol{d}_2 = \{d^x_2, d^y_2, d^z_2\}$ from $\mathbf{A}_2$ (6 unknowns with 9 equations).

Extracting ω_1 and d_1 from $\mathbf{A}_1$: Based on the definition of $\mathbf{A}_1$ in Eq. (5.17), we obtain the following linear system in $\boldsymbol{\omega}_1$ and $\boldsymbol{d}_1$:

$$\begin{bmatrix} 0 & -G_{0,11} & G_{0,12} & n_0^x \mathbf{R}_{0,(1)} \\ G_{0,13} & 0 & -G_{0,11} & n_0^y \mathbf{R}_{0,(1)} \\ -G_{0,12} & G_{0,11} & 0 & n_0^z \mathbf{R}_{0,(1)} \\ 0 & -G_{0,23} & G_{0,22} & n_0^x \mathbf{R}_{0,(2)} \\ G_{0,23} & 0 & -G_{0,21} & n_0^y \mathbf{R}_{0,(2)} \\ -G_{0,22} & G_{0,21} & 0 & n_0^z \mathbf{R}_{0,(2)} \\ 0 & -G_{0,33} & G_{0,32} & n_0^x \mathbf{R}_{0,(3)} \\ G_{0,33} & 0 & -G_{0,31} & n_0^y \mathbf{R}_{0,(3)} \\ -G_{0,22} & G_{0,31} & 0 & n_0^z \mathbf{R}_{0,(3)} \end{bmatrix} \begin{pmatrix} \omega_1^x \\ \omega_1^y \\ \omega_1^z \\ d_1^x \\ d_1^y \\ d_1^z \end{pmatrix} = \mathbf{0} \tag{5.26}$$

where the auxiliary matrix $\mathbf{G}$ is defined as $\mathbf{G} = \mathbf{R}_0 + \mathbf{t}_0 \mathbf{n}_0^\top$. As a result, 6 unknowns in $\boldsymbol{\omega}_1$ and $\boldsymbol{d}_1$ can be obtained by solving Eq. (5.26) linearly.

Extracting ω_2 and d_2 from $\mathbf{A}_2$: Based on the definition of $\mathbf{A}_2$ in Eq. (5.17), we obtain the following linear system in $\boldsymbol{\omega}_1$ and $\boldsymbol{d}_1$:

$$\begin{bmatrix} \mathbf{0}^\top & \mathbf{R}_{0,(3)}^\top & -\mathbf{R}_{0,(2)}^\top & -\mathbf{n}_0^\top & \mathbf{0}^\top & \mathbf{0}^\top \\ -\mathbf{R}_{0,(3)}^\top & \mathbf{0}^\top & \mathbf{R}_{0,(1)}^\top & \mathbf{0}^\top & -\mathbf{n}_0^\top & \mathbf{0}^\top \\ \mathbf{R}_{0,(2)}^\top & \mathbf{R}_{0,(1)}^\top & \mathbf{0}^\top & \mathbf{0}^\top & \mathbf{0}^\top & -\mathbf{n}_0^\top \end{bmatrix} \begin{bmatrix} \omega_2^x \\ \omega_2^y \\ \omega_2^z \\ d_2^x \\ d_2^y \\ d_2^z \end{bmatrix} = \mathbf{0} \tag{5.27}$$

Thus $\boldsymbol{\omega}_2$ and $\boldsymbol{d}_2$ can be obtained by solving Eq. (5.27) linearly.

5.6.3 Nonlinear Refinement.

The final step consists in a nonlinear refinement of pose and instantaneous-motion parameters with n pairs of point matches which are the inliers from the 13pt-RANSAC. This is achieved by minimizing the following cost function where the full homography matrix is now used:

$$\underset{\mathbf{R}_0,\mathbf{t}_0,\mathbf{n}_0,d_0,\boldsymbol{\omega}_1,\boldsymbol{\omega}_2,\mathbf{d}_1,\mathbf{d}_2}{\arg\min} = \sum_{i=1}^{n} (\mathbf{M}_{RS,i}\mathbf{h}_{RS} - \mathbf{b}_i)^2 \tag{5.28}$$

5.7 RS Image Stitching

5.7.1 Stitching Assumption

The goal of image stitching is to create a very wide angle image (or a panorama) from a set of images. After finding the homography matrix that aligns each pair of neighboring cameras, all the images are transformed so that they are mapped into the same projective space.

For that purpose, the cameras are assumed to have rotated about (approximately) the same centre of projection (Fig. 5.3). Thus, the RS homography matrix is further simplified by setting $\mathbf{t}_i$, $\mathbf{d}_1$ and $\mathbf{d}_2$ to 0, which leads to $\mathbf{H}_{GS} = \mathbf{R}_0$, $\mathbf{A}_1 = -\mathbf{R}_0[\boldsymbol{\omega}_1]_\times$ and $\mathbf{A}_2 = [\boldsymbol{\omega}_2]_\times \mathbf{R}_0$.

5.7.2 Stitching Pipeline

The proposed RS stitching pipeline consists of 4 steps:

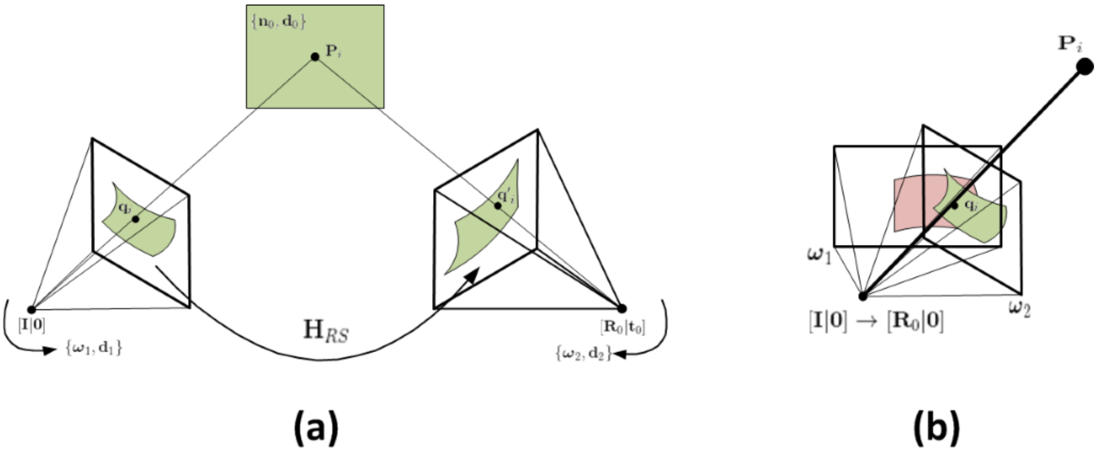

$\{\mathbf{n}_0, \mathbf{d}_0\}$
$\mathbf{P}_i$
$\mathbf{q}_i$
$\mathbf{q}'_i$
$[\mathbf{I}|\mathbf{0}]$
$\{\boldsymbol{\omega}_1, \mathbf{d}_1\}$
H_{RS}
$[\mathbf{R}_0|\mathbf{t}_0]$
$\{\boldsymbol{\omega}_2, \mathbf{d}_2\}$
(a)
$\mathbf{P}_i$
$\mathbf{q}_i$
ω_1
ω_2
$[\mathbf{I}|\mathbf{0}] \rightarrow [\mathbf{R}_0|\mathbf{0}]$
(b)

FIGURE 5.3: RS homographies in the general case (a) and in the pure-rotation case (b).

(1) RS homography estimation with 13pt solver: The RS homography matrix of coinciding optical centres has the same structure as the simplified RS homography matrix in Eq. (5.17). Thus, the 13pt method (section 5.5.3) is also feasible here.

(2) RS image alignment: When aligning two GS images all image points are directly mapped to new locations by applying $\mathbf{q}'_i = \mathbf{H}_{GS}\mathbf{q}_i$. Differently, in the RS case we have $\mathbf{q}'_i = (\mathbf{H}_{GS} + \mathbf{A}_1 v_i + \mathbf{A}_2 v'_i)\mathbf{q}_i$. Note that the row index v'_i is present in both sides of the equation. Thus, the point coordinates mapping equation becomes a second degree equation which is solved as follows:

$$\begin{cases} v'_i = \frac{-b\pm\sqrt{b^2-4ac}}{2a} \\ u'_i = \frac{(\mathbf{H}_{GS,(1)}+\mathbf{A}_{1,(1)}v_i+\mathbf{A}_{2,(1)}v'_i)\mathbf{q}_i}{(\mathbf{H}_{GS,(3)}+\mathbf{A}_{1,(3)}v_i+\mathbf{A}_{2,(3)}v'_i)\mathbf{q}_i} \end{cases}$$
$$\begin{aligned} \text{where,}\quad a &= \mathbf{A}_{2,(3)}q_i \\ b &= \mathbf{H}_{GS,(3)}q_i + \mathbf{A}_{1,(3)}q_i v_i - \mathbf{A}_{2,(2)}q_i \\ c &= -\mathbf{H}_{GS,(2)}q_i - \mathbf{A}_{1,(3)}q_i v_i \end{aligned} \tag{5.29}$$

There are two feasible solutions for each pixel in the original image. The solution with the smaller Euclidean distance to original point before warping is chosen as the correct one. This is usually the solution which maintains the consistency of image registration as we will show through experiments.

(3) Blending: To seamlessly blend the images, a multi-band blending strategy [Burt and Adelson, 1983, Brown et al., 2003] is used. In this procedure, we first decompose the images into a set of band-pass filtered component images. Then a corresponding bandpass mosaic are generated by assembling the component images in each spatial frequency hand. Finally, we sum these band-pass mosaic images to obtain the desired image mosaic.

(4) Correction of stitched RS images: After determining the instantaneous-motion parameters by means of the method described in section 5.6, an inverse mapping is applied to the aligned image points in order to remove RS distortions by compensating camera instantaneous-motion as follows [Lao and Ait-Aider, 2018]:

$$\mathbf{m}^{correct} = \mathbf{K}_2(\mathbf{I} - [\boldsymbol{\omega}_2]_\times)\mathbf{q}' \tag{5.30}$$

5.7.3 Uncalibrated Image Stitching

5.7.3.1 Direct RS Image Alignment

In image stitching applications, it is common that the input images are uncalibrated. Considering that points $\mathbf{m}_i$ and $\mathbf{m}'_i$ are the image measurements in pixels (i.e $[u_i, v_i, 1]$) instead of normalized points, Eq. (5.21) gives the uncalibrated $\mathbf{H}_{RS,i}^{image}$ which is defined as follows:

$$\begin{aligned}
\mathbf{H}_{GS}^{image} &= \mathbf{K}_2\mathbf{H}_{GS}\mathbf{K}_1{}^{-1} \\
\mathbf{A}_1^{image} &= \mathbf{K}_2\mathbf{A}_1\mathbf{K}_1{}^{-1} \\
\mathbf{A}_2^{image} &= \mathbf{K}_2\mathbf{A}_2\mathbf{K}_1{}^{-1}
\end{aligned} \tag{5.31}$$

where $\mathbf{K}_1$ and $\mathbf{K}_2$ are the calibration matrices of the first and second cameras respectively.

Thus the 13pt method (section 5.5.3) can be used to align RS image directly in image space without prior knowledge of the calibration matrices $\mathbf{K}_1$ and $\mathbf{K}_2$.

5.7.3.2 RS Correction of Uncalibrated Images

The determination of the instantaneous-motion parameters from $\mathbf{A}_1^{image}$ and $\mathbf{A}_2^{image}$ is different from the decomposition method of $\mathbf{A}_1$ and $\mathbf{A}_2$ described in section 5.6 since the calibration matrices are unknown.

Thus, basing on the pin-hole camera model, we assume that principle point is located in the centre of the image. Thus only the focal length f remains unknown. Now, the problem is to estimate the focal length f and instantaneous-motions $\boldsymbol{\omega}_1$ and $\boldsymbol{\omega}_2$ given $\mathbf{H}_{GS}^{image}$, $\mathbf{A}_1^{image}$ and $\mathbf{A}_2^{image}$. We first set the focal length as 0.9 times of the maximal dimension of each corresponding image [Purkait and Zach, 2017]. By using the direct relative pose and instantaneous-motion algorithm in section 5.6, we can roughly estimate the rotation between the two images and angular velocities. Finally, we perform an iterative refinement to estimate the focal lengths, the rotation between cameras and instantaneous-motions as follows:

$$\underset{f_1,f_2,\mathbf{R}_0,\mathbf{n}_0\boldsymbol{\omega}_1,\boldsymbol{\omega}_2}{\arg\min} = \sum_{i=1}^{n}(\mathbf{M}_{RS,i}\mathbf{h}_{RS} - \mathbf{b}_i)^2 \tag{5.32}$$

With the estimated parameters, an inverse mapping similar to Eq. (5.30) is applied to the stitched image directly, as follows:

$$\mathbf{m}^{correct} = \mathbf{K}_2(\mathbf{I} - [\boldsymbol{\omega}_2]_\times)\mathbf{K}_2{}^{-1}\mathbf{m}' \tag{5.33}$$

5.8 Experiments

Both the RS homography-based pose estimation (**RSH**) and image stitching method presented in this chapter were evaluated on synthetic and real data. The proposed methods were also compared to GS homography methods such as:

- Classical GS-based homograhy relative pose estimation method (**GSH**) and its stitching application **AutoStitch** [Brown and Lowe, 2007].

- Local multiple homography method: As-Projective-as-Possible stitching method (**APAP**) [Zaragoza et al., 2014].
- Local multiple homography method: Adaptive as-natural-as-possible image stitching method (**AANAP**) [Lin et al., 2015].
- Microsoft image stitching software: Image Composite Editor (**ICE**) [ICE,].
- Stitching function in Adobe PhotoShop: **Photoshop** [pho,].

Results are summarized below.

5.8.1 Relative pose estimation

5.8.1.1 Synthetic data experiments

Experiment setting: We generated a planar scene with 60 feature points, which was imaged by two RS cameras with 480 × 640 image resolution. We set the distance from the plane to the optical centre of the first camera as 1 unit, and located the second camera randomly on a sphere around the centre of the plane with 1 unit length radius. Since the ground truth of the relative pose is known, we calculated the rotation error as $e_{\text{rot}} = \arccos((\text{tr}(\mathbf{R}\mathbf{R}_{\text{GT}}^{\top}) - 1)/2)$ and the translation error as $e_{\text{trans}} = \arccos(\mathbf{t}^{\top}\mathbf{t}_{\text{GT}}/(\|\mathbf{t}\| \, \|\mathbf{t}_{\text{GT}}\|))$. We compared **GSG** and **RSH** by varying the noise in the image, the rotational and the translational speeds. The results are obtained after averaging the errors over 50 trials (the default setting is 1pixel noise, 10 degs/frame and 0.04 units/frame for the rotational and translational speed).

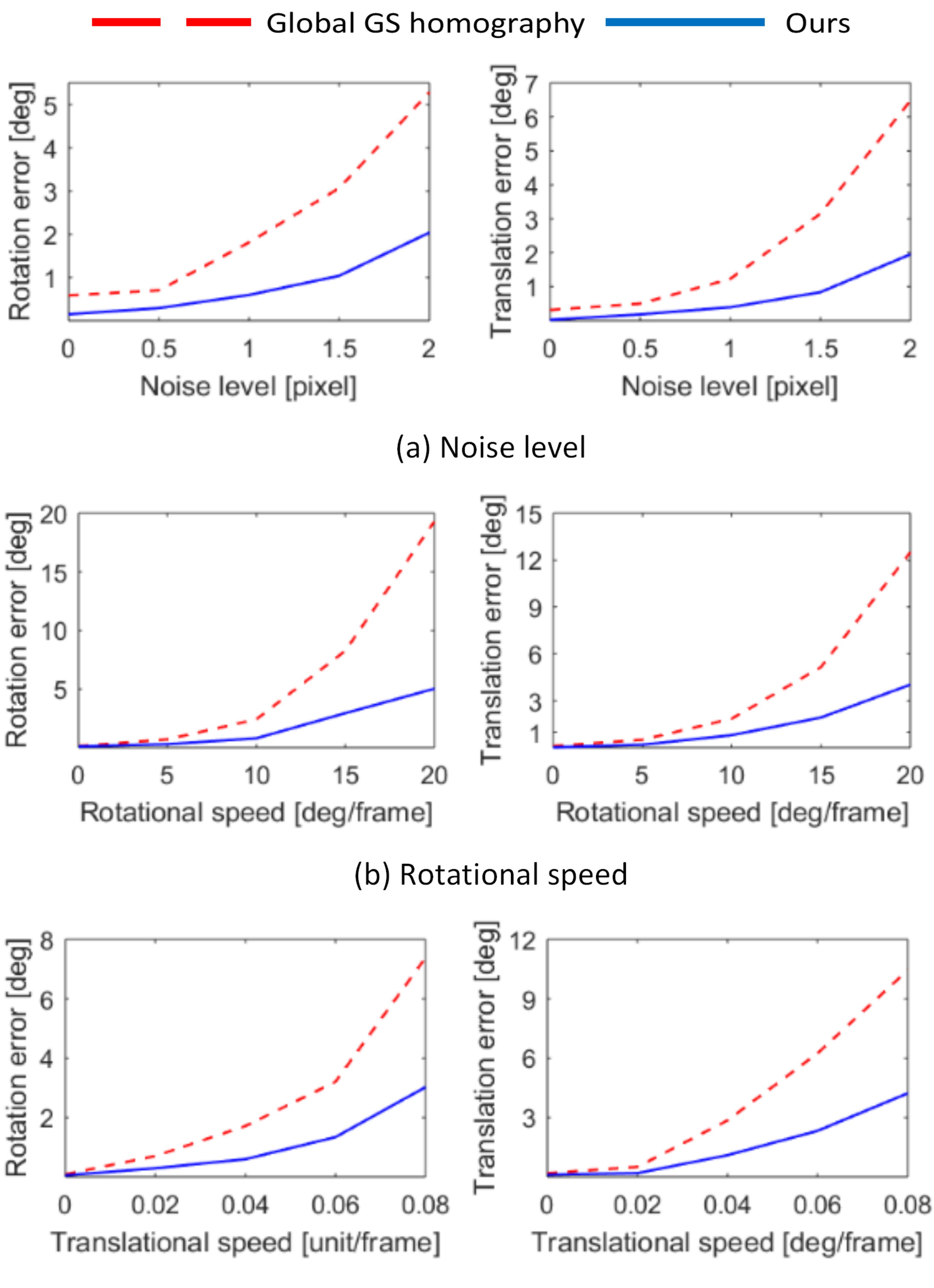
Relative pose estimation accuracy vs varying noise. We first tested the stability of the proposed method in the presence of image noise. Here we increased the random Gaussian image noise from 0 to 2pixels. Results in Fig. 5.4 show that the proposed method is more stable and achieves higher accuracy than **GSH**.

Relative pose estimation accuracy vs varying instantaneous-motion speed. We also evaluated the performances by increasing the rotational speed from 0 to 20 degs/frame and the translational speed from 0 to 0.08 units/frame receptively. As shown in Fig. 5.4, our method obtains obvious improvements comparing to **GSH**.

Number of detected inliers vs varying instantaneous-motion speed. Finally, we investigated the influence of the RS instantaneous-motion on RANSAC-based inlier selection. As shown in Fig. 5.5, with the increase of the camera instantaneous-motion, the inlier detection rate of **GSH**-RANSAC decreases dramatically. In contrast, the proposed **GSH** with RANSAC maintains its good performance.

5.8.1.2 Real data experiments

(1) Plane-based trajectory estimation
For this experiment we used sequence '01' and '22' from [Hedborg et al., 2012] which was captured by an iPhone4 camera. However, contrarily to [Hedborg et al., 2012] and [Zhuang et al., 2017] which require smooth motion input video, the proposed method can handle large baselines. Thus we just selected non-successive frames with 9 frames interval as an input sequence.



(a) Noise level

(b) Rotational speed

(c) Translational speed

FIGURE 5.4: Errors of relative pose estimation by using **GSH** and **RSH** with increasing image noise level (a), rotational speed (b) and translational speed(c).

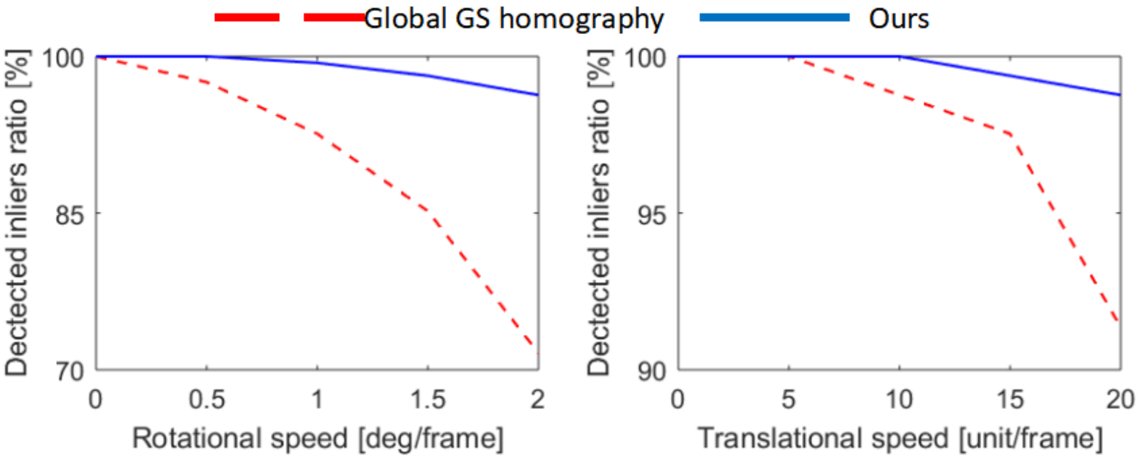


FIGURE 5.5: Rate of inliers by using **GSH** and **RSH** with increasing rotational speed and translational speed.

TABLE 5.2: Errors of the trajectory estimation for 'seq01' and 'seq02'. Errors of homography transformation and 3D reconstruction for seq 'trans' and seq 'rot'.

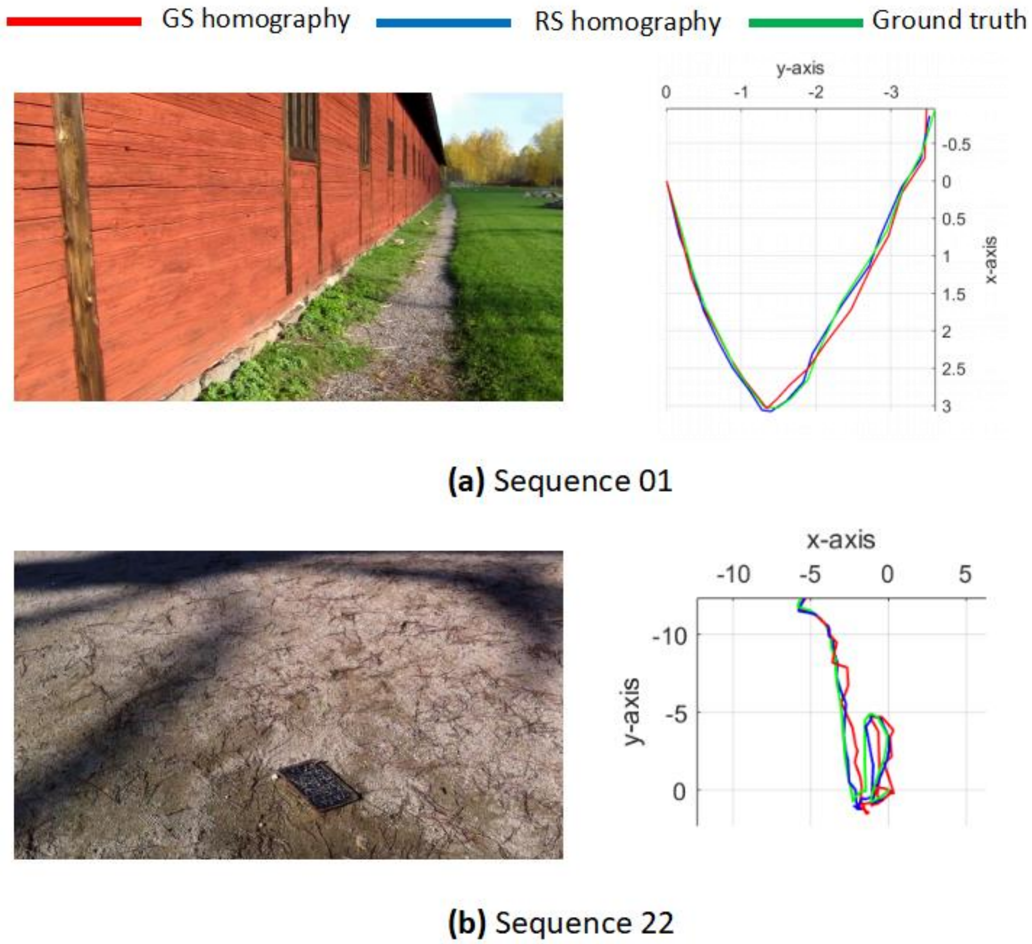
	GSH	RSH
Trajectory error on seq01 [units]	0.903	**0.199**
Trajectory error on seq22 [units]	3.620	**1.668**
Transform errors on seq 'trans' [pixels]	2.824	**1.668**
Transform errors on seq 'rot' [pixels]	4.131	**2.005**
Reconstruction errors on seq 'trans' [units]	1.833	**0.317**
Reconstruction errors on seq 'rot' [pixels]	2.519	**0.397**

We then performed **RSH** for the relative pose estimation and used camera-based RS bundle adjustment [Lao et al., 2018a] to refine the poses. Beside, we applied **GSH** to estimate the relative pose and run the GS plane-based bundle adjustment described but with known calibration matrix [Zhou et al., 2012].

We used the method described in [Hedborg et al., 2012] to calculate the trajectory error. The visual and quantitative evaluations summarized in Fig. 5.6 and Table 5.2 show that the proposed **RSH** method performs significantly better than **GSH** in both sequences.

(2) Plane-based SfM

We evaluated both **GSH** and **RSH** on two more challenging RS image sequences from [Lao et al., 2018b]: 'trans' and 'rot' which are taken with mainly translational and rotational velocities respectively. Two RS images of a chessboard (also present in the images) are chosen from the same sequence to estimate the relative pose with **GSH** and **RSH**.



(a) Sequence 01

(b) Sequence 22

FIGURE 5.6: Comparison of trajectory estimation (right sides) by using **GSH** and **RSH** on two RS image sequences (examples of input RS images are shown on the left side).

Then we perform a triangulation to reconstruct the chessboard (note that the pose of a row of RS image is obtained by using Eq. (6.8)).

Since the ground-truth of the poses are unknown, we use two methods to evaluate the accuracy of the relative pose estimation:

- Average homography transform errors of the feature points in the chessboard from the first image to the second.
- The reconstruction errors of the chessboard (each reconstructed 3D point is spatially aligned to the ground-truth, by minimizing the sum of all squared point-to-point distances).

The results presented in Fig. 5.7 and Table 5.2 show that **RSH** obtains significantly better results compared to **GSH** in both sequences.

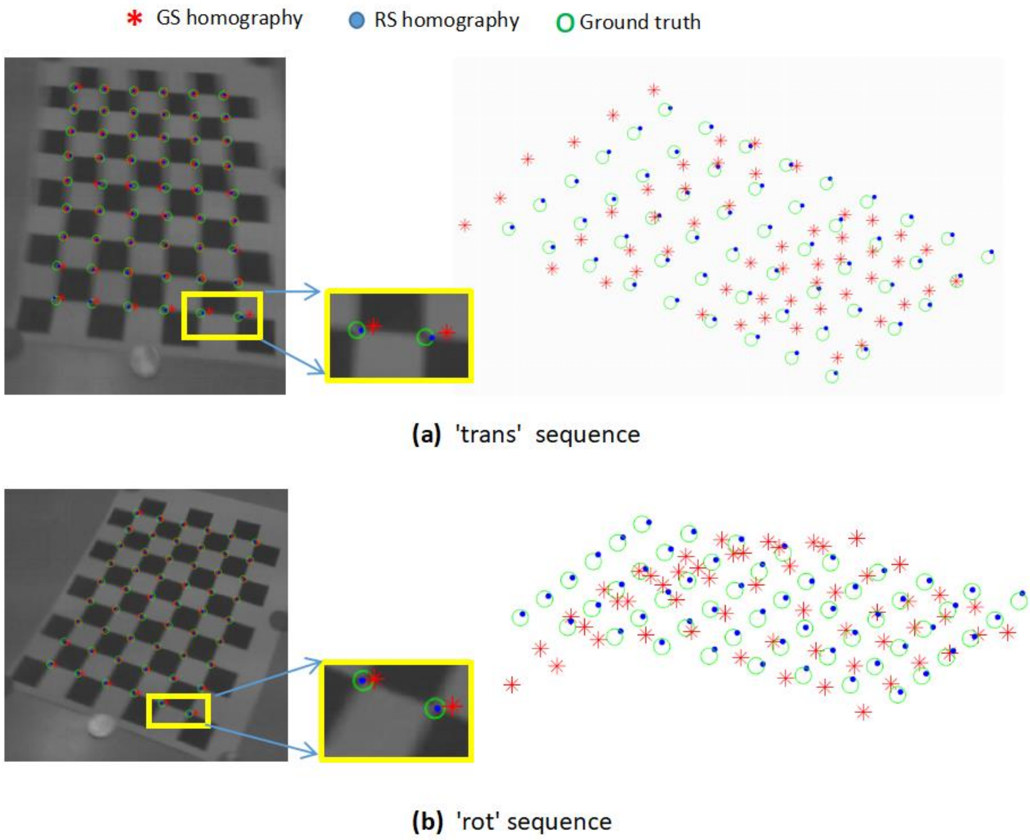


FIGURE 5.7: Mapping and 3D reconstruction errors by using **GSH** and **RSH** on sequence 'trans' and 'rot' respectively.

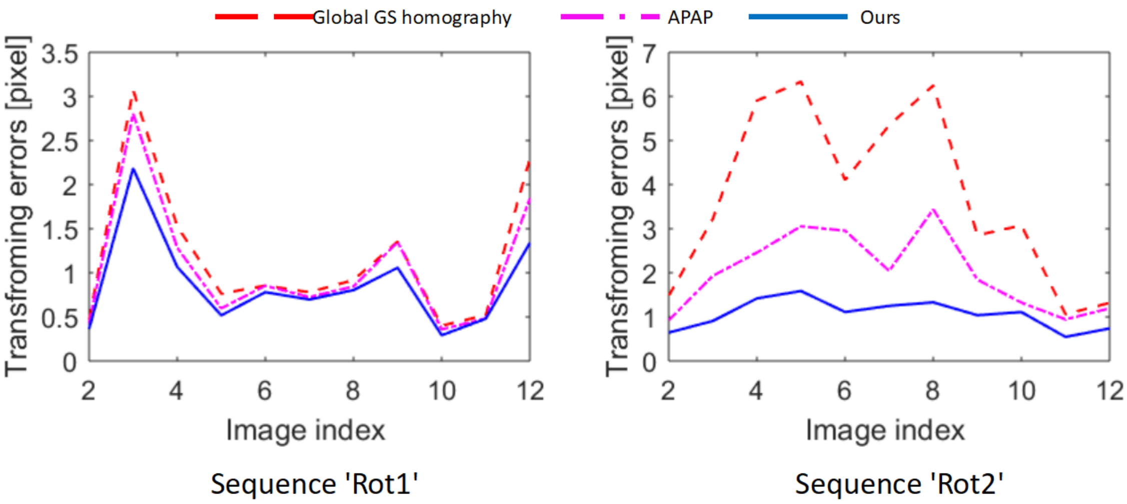


FIGURE 5.8: Mapping errors by using **GSH**, **APAP** and **RSH** on sequence 'rot1' and 'rot2' respectively.

TABLE 5.3: Stitching results on 'rot2' RS sequence. ✓ indicates successful stitching without obvious stitching errors.

	AutoStitch	**ICE**	**Photoshop**	**APAP**	**AANAP**	**RSH**
Fig. 5.9 frame2 to frame1	ghosting	✓	✓	✓	✓	✓
Fig. 5.9 frame3 to frame1	ghosting	✓	✓	✓	✓	✓
Fig. 5.9 frame4 to frame1	ghosting	misalignment	misalignment	ghosting + misalignment	ghosting + misalignment	✓
Fig. 5.10 frame5 to frame1	ghosting	misalignment	misalignment	ghosting + misalignment	ghosting + misalignment	✓
Fig. 5.10 frame6 to frame1	ghosting	misalignment	misalignment	ghosting + misalignment	ghosting + misalignment	✓
Fig. 5.10 frame7 to frame1	ghosting	misalignment	misalignment	ghosting + misalignment	ghosting + misalignment	✓
Fig. 5.11 frame8 to frame1	ghosting + misalignment	misalignment	misalignment	ghosting + misalignment	ghosting	✓
Fig. 5.11 frame9 to frame1	ghosting + misalignment	misalignment	misalignment	ghosting + misalignment	ghosting	✓
Fig. 5.11 frame10 to frame1	ghosting	misalignment	misalignment	misalignment	✓	✓
Fig. 5.12 frame11 to frame1	✓	✓	✓	✓	✓	✓
Fig. 5.12 frame12 to frame1	✓	✓	✓	✓	✓	✓
Success rate %	18.2	36.4	36.4	36.4	45.5	100

5.8.2 Image Stitching

5.8.2.1 Synthetic Dataset

We first compared the performances of **GSH**, **APAP**, **ANNAP** and **RSH** on two synthetic RS image sequences with pure rotation from [Forssén and Ringaby, 2010]: 'rot1'(camera aim changing) and 'rot2'(both changing camera aim, and in-plane rotation).

In order to evaluate the stability of all the methods with different instantaneous-motions, we chose the first frame of each sequence as a reference, then we transformed and stitched all the other frames to it. We kept the number of input feature matches the same for all the three methods and calculated the average transformation errors.

The results in Fig. 5.9, Fig. 5.10, Fig. 5.11 and Fig. 5.12 show the stitching results of frame 1 to frames 2,3,...,12 in 'rot2' video sequence. The results summarized in Table. 5.3 show that the proposed method **RSH** clearly outperforms all the other methods.

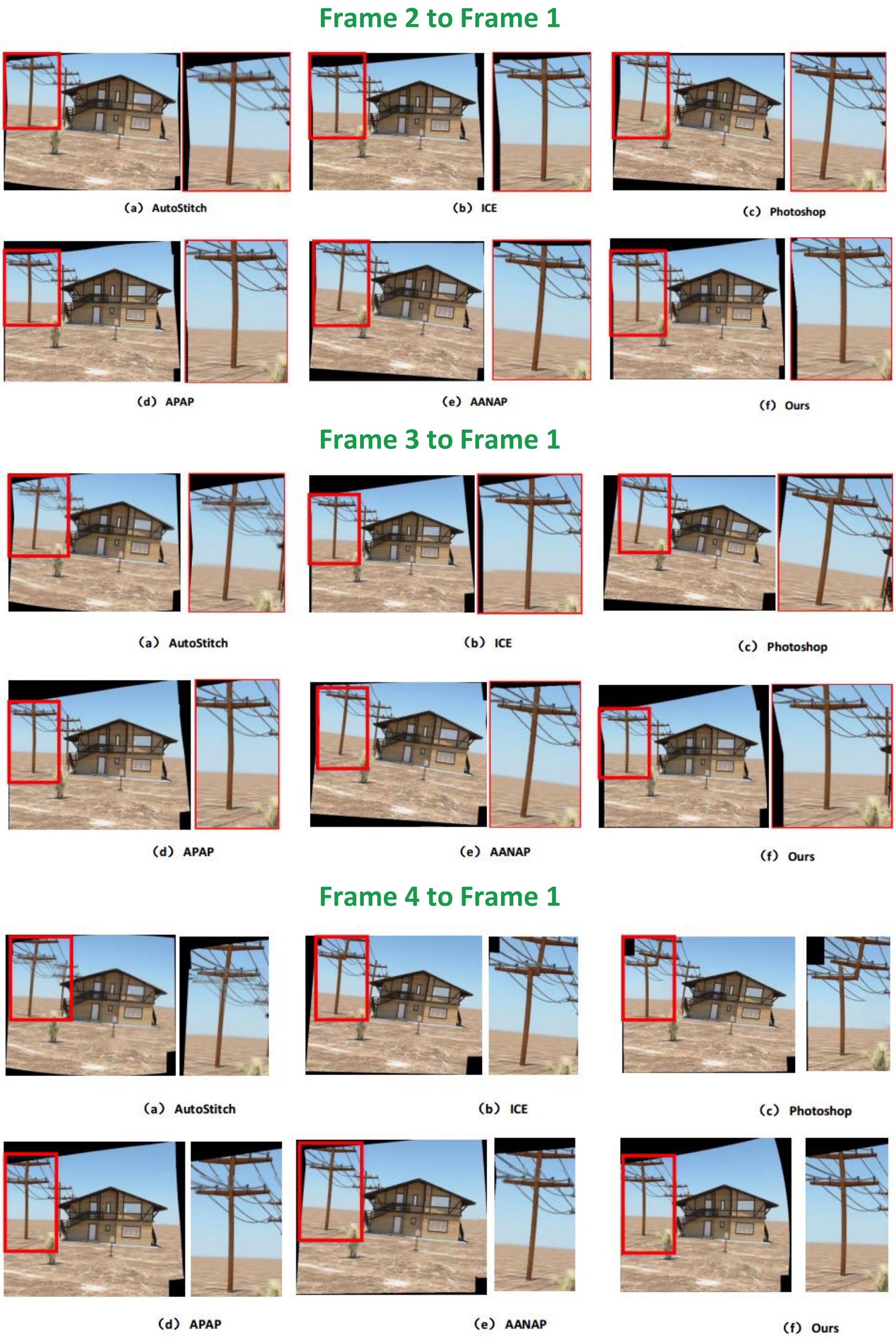
The results in Fig. 5.8 show that in the only aim changing sequence, all the three methods obtain similar performances while **RSH** is slightly better than **GSH** and **APAP** in all the groups. However, with in-plane rotation, the transformation errors of **GSH** and **APAP** increase dramatically, while **RSH** performs obviously better.

5.8.2.2 Real Images

(1) Images from [Forssén and Ringaby, 2010]. The first input image pair is from a RS image sequence 'indicator' [Forssén and Ringaby, 2010] taken by an iPhone4. The second input is from a self-capture dataset 'facade' with strong RS effects. In Fig. 5.13, we can observe that **AutoStitch** produces blur on stitched images while the results from **APAP** are slightly better. The result of **ICE** in 'indicator' dataset is visually acceptable although significant misalignments can be observed along the pole. For the 'facade' dataset, **ICE** gives a dramatically mismatched result. **Photoshop** performances are visually good in both datasets, however, some wrong alignments are present such as the pole in the 'indicator' dataset and the eave in the 'facade' dataset. In contrast, **HRS** achieves the best results. After stitching, our method can remove the distortions and offers a much more visually pleasant stitching image as final outputs.

(2) Images from 'facade' sequence. In this experiment, we evaluate the stitching quality with varying number of point-matches by decreasing the number of point correspondences. We conduct this evaluations on 'facade' dataset. Results in Fig. 5.14 show that the multiple local homographies (spatially-varying warping) methods such as **APAP** and **AANAP** are sensitive to the number of input point-matches. With the decreasing of input matches, the quality of stitching results with **APAP** and **AANAP** declines dramatically. In contrast, the global methods **GSH** and **RSH** show a relative high stability.

(3) Images from [Jia and Evans, 2012]. In this experiment, we evaluate all the stitching methods on a real RS images sequence from [Jia and Evans, 2012] which captured a urban scene under fast rotation. The results in Fig. 5.15 and 5.16 show that **AutoStitch** fails in image alignment. **APAP** and **AANAP** provide blur stitching in regions with lack of point-matches. **ICE** and **Photoshop** provide visually pleasant results, however, geometry inconsistencies or 'object's fracture' are present along the stitching seams. In contrast, the proposed method **RSH** obtains the best results. Note that **RSH** can not align the non-rigid objects such as moving cars or pedestrians, nevertheless, **APAP** and **AANAP** are not able to handle it neither.



Frame 2 to Frame 1

(a) AutoStitch (b) ICE (c) Photoshop

(d) APAP (e) AANAP (f) Ours

Frame 3 to Frame 1

(a) AutoStitch (b) ICE (c) Photoshop

(d) APAP (e) AANAP (f) Ours

Frame 4 to Frame 1

(a) AutoStitch (b) ICE (c) Photoshop

(d) APAP (e) AANAP (f) Ours

FIGURE 5.9: Stitching frame02, frame03 and frame04 to frame01 of 'rot2' sequence.

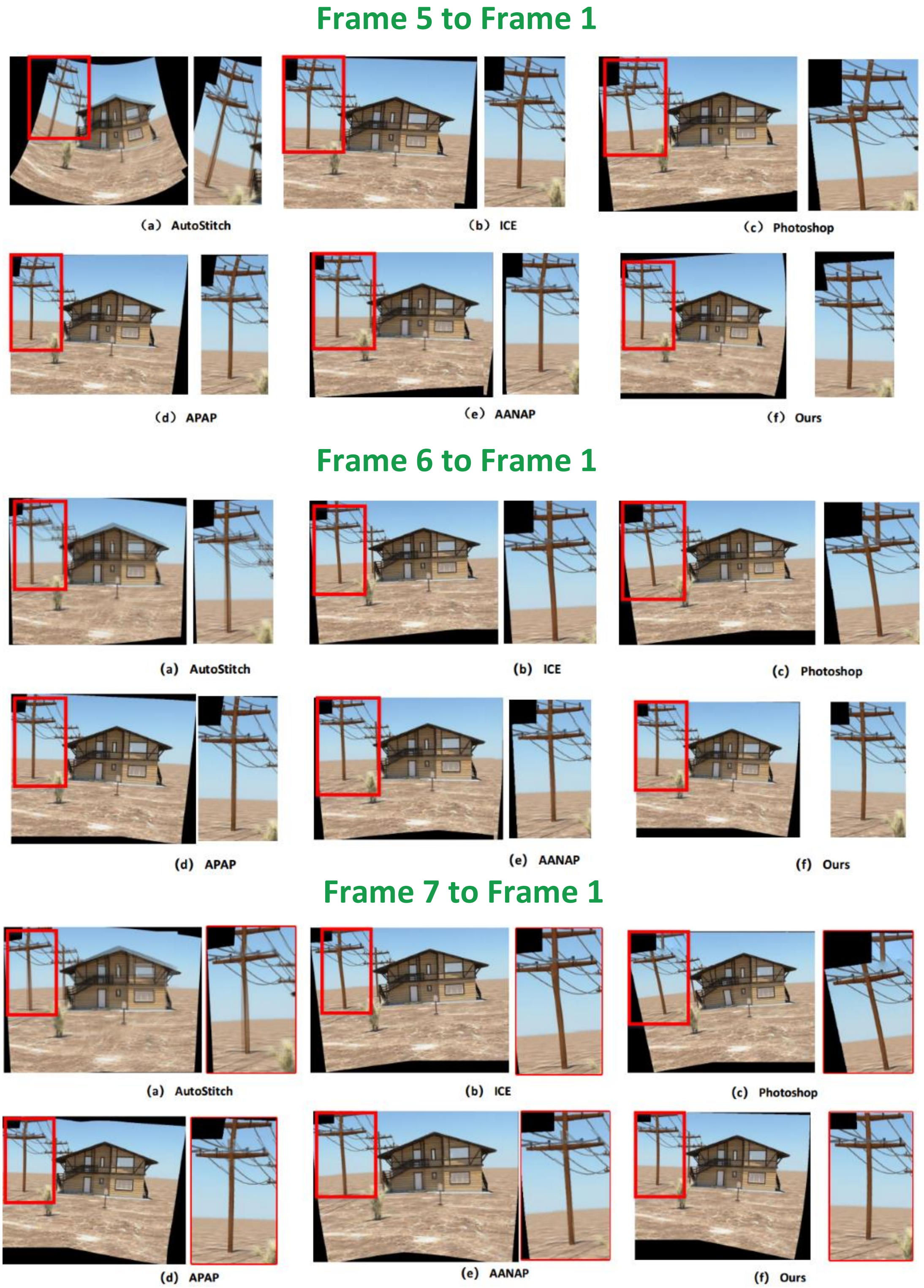
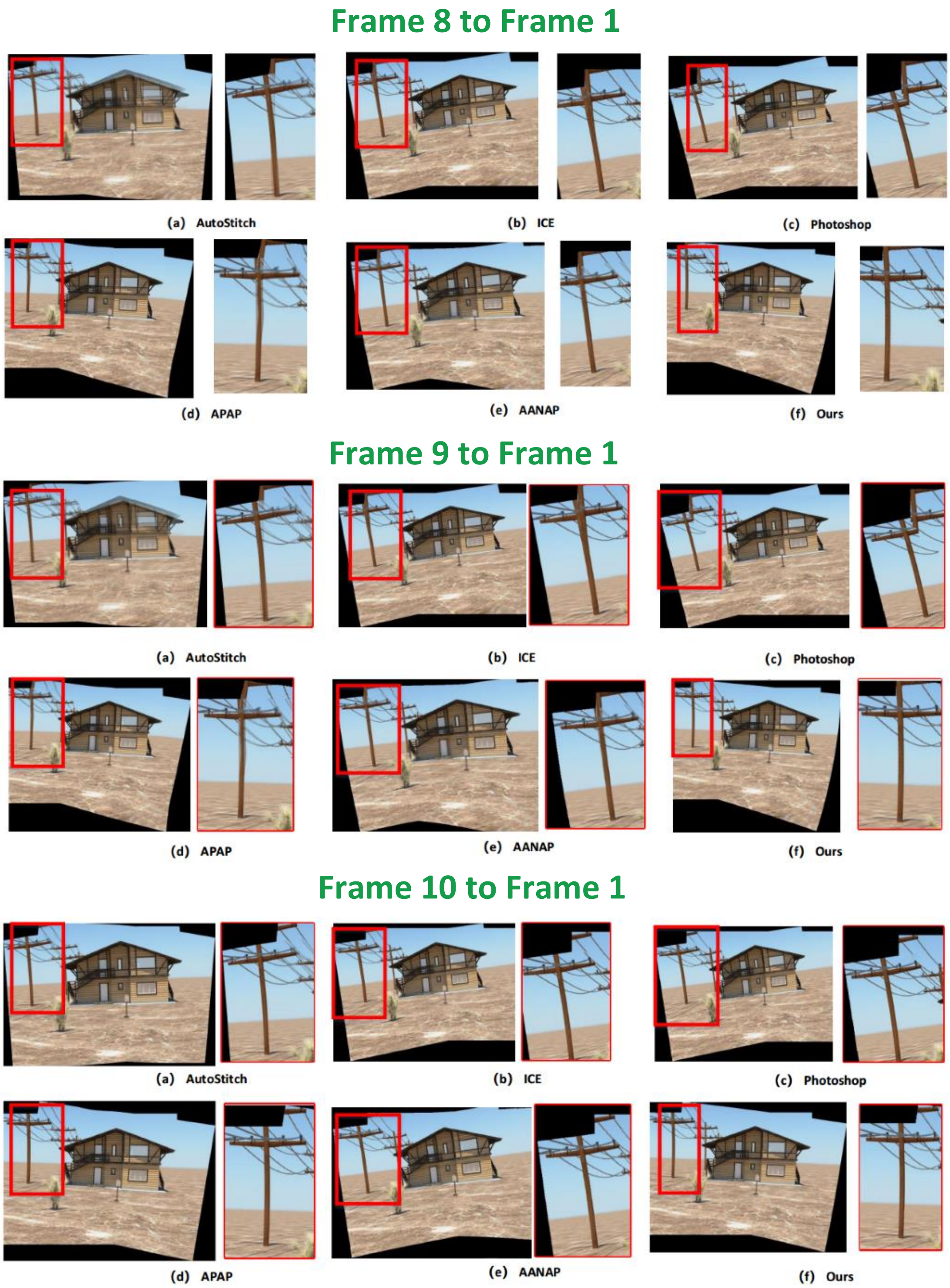


FIGURE 5.10: Stitching frame05, frame06 and frame07 to frame01 of 'rot2' sequence.



Frame 8 to Frame 1

(a) AutoStitch (b) ICE (c) Photoshop

(d) APAP (e) AANAP (f) Ours

Frame 9 to Frame 1

(a) AutoStitch (b) ICE (c) Photoshop

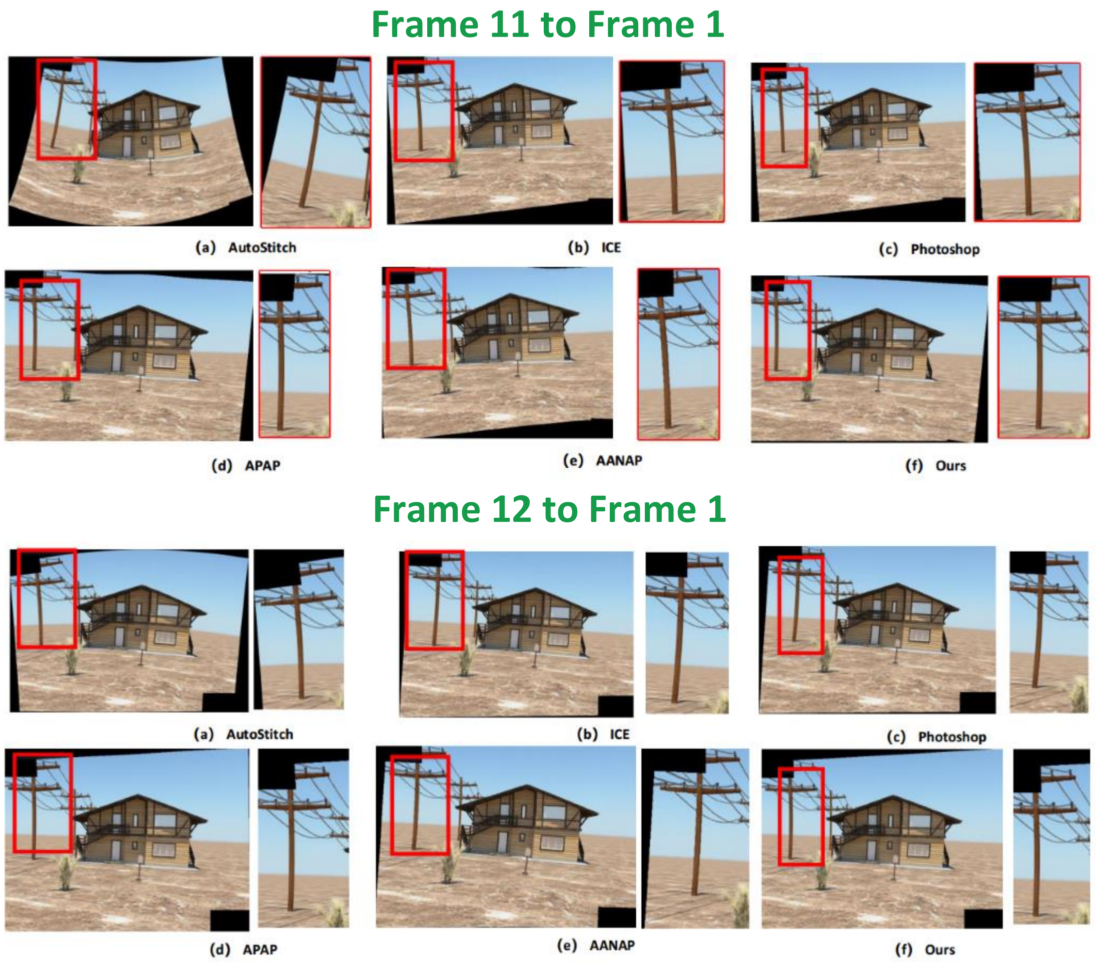
(d) APAP (e) AANAP (f) Ours

Frame 10 to Frame 1

(a) AutoStitch (b) ICE (c) Photoshop

(d) APAP (e) AANAP (f) Ours

FIGURE 5.11: Stitching frame08, frame09 and frame10 to frame01 of 'rot2' sequence.



Frame 11 to Frame 1

(a) AutoStitch (b) ICE (c) Photoshop

(d) APAP (e) AANAP (f) Ours

Frame 12 to Frame 1

(a) AutoStitch (b) ICE (c) Photoshop

(d) APAP (e) AANAP (f) Ours

FIGURE 5.12: Stitching frame11 and frame12 to frame01 of 'rot2' sequence.

(4) Images from [Zhuang et al., 2017]. In this experiment, we evaluate all the stitching methods by using two real RS image from [Zhuang et al., 2017], which have a large overlap region. As shown in Fig. 5.17, the inliers between the two images distribute densely in the right part of the two images while being limited to the white facade on the left part. The stitching results in Fig.5.18 show that **APAP** and **AANAP** suffer from this unbalanced point-matches distribution and provide significant distortions on the stitched regions of the 'white facade'. Slight geometrical inconsistencies are present along the stitching seams of **ICE** and **Photoshop**'s results. In contrast, the proposed method **RSH** obtains the best result.

5.8.3 Running Time

The experiments were conducted on an i5 CPU at 2.8GHz with 4G RAM. On average, it took around 3.35s for **GSH**, 13.5s for **APAP** and 6.5s for RSH(0.05s for 13pt solver running per time, 0.16s for the non-linear refinement, 5.9s for the image warping, blending and RS correction). Since the proposed method was implemented in MATLAB, a significant improvement can be expected when using C++.

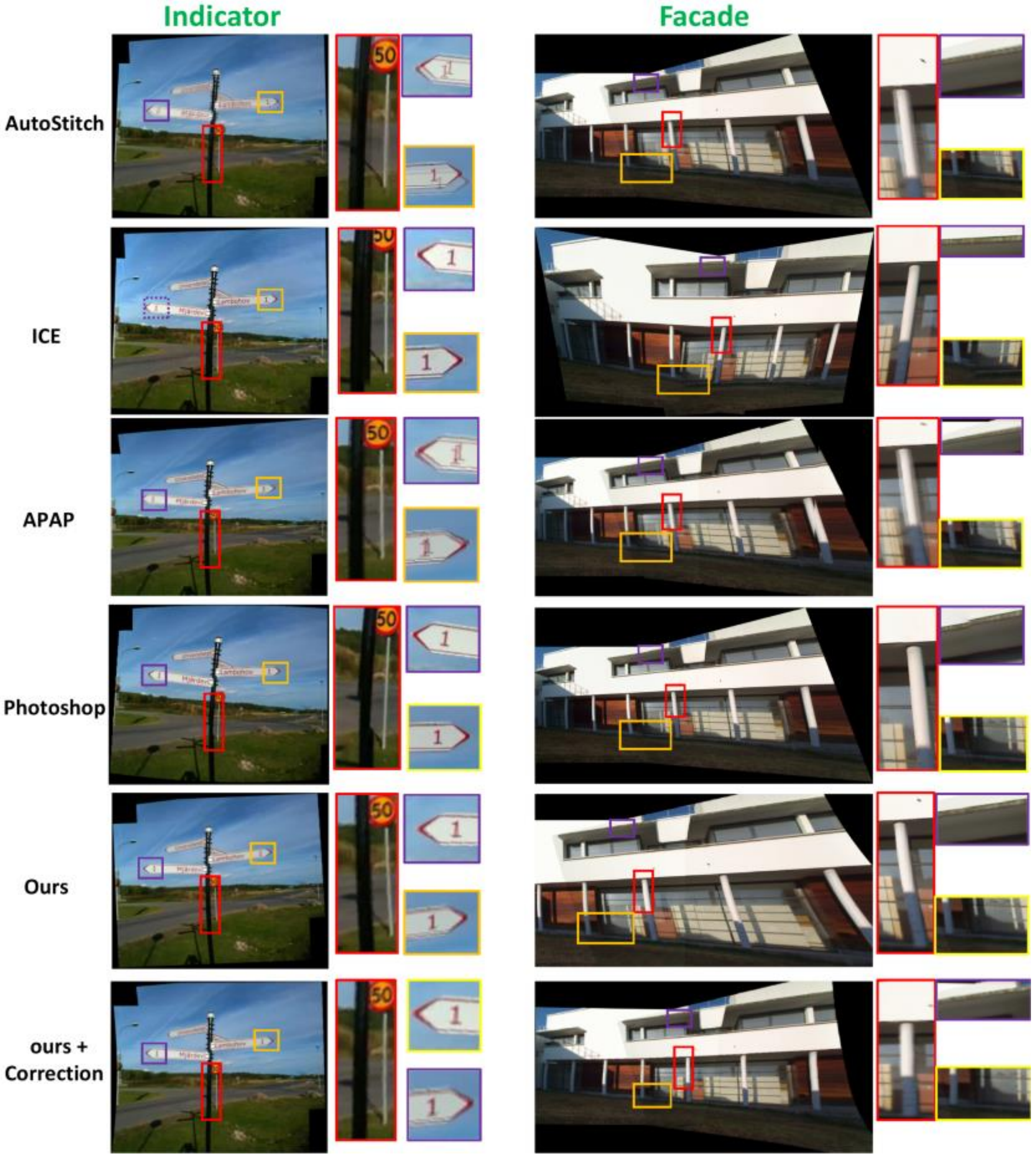


FIGURE 5.13: Results of real uncalibrated RS images stitching with different methods.

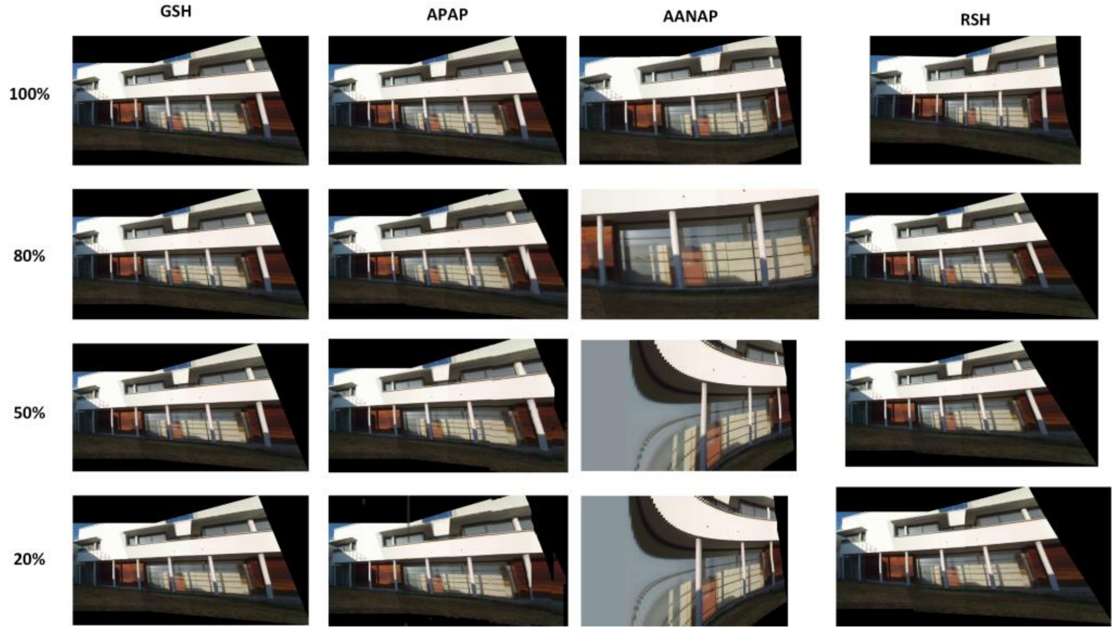
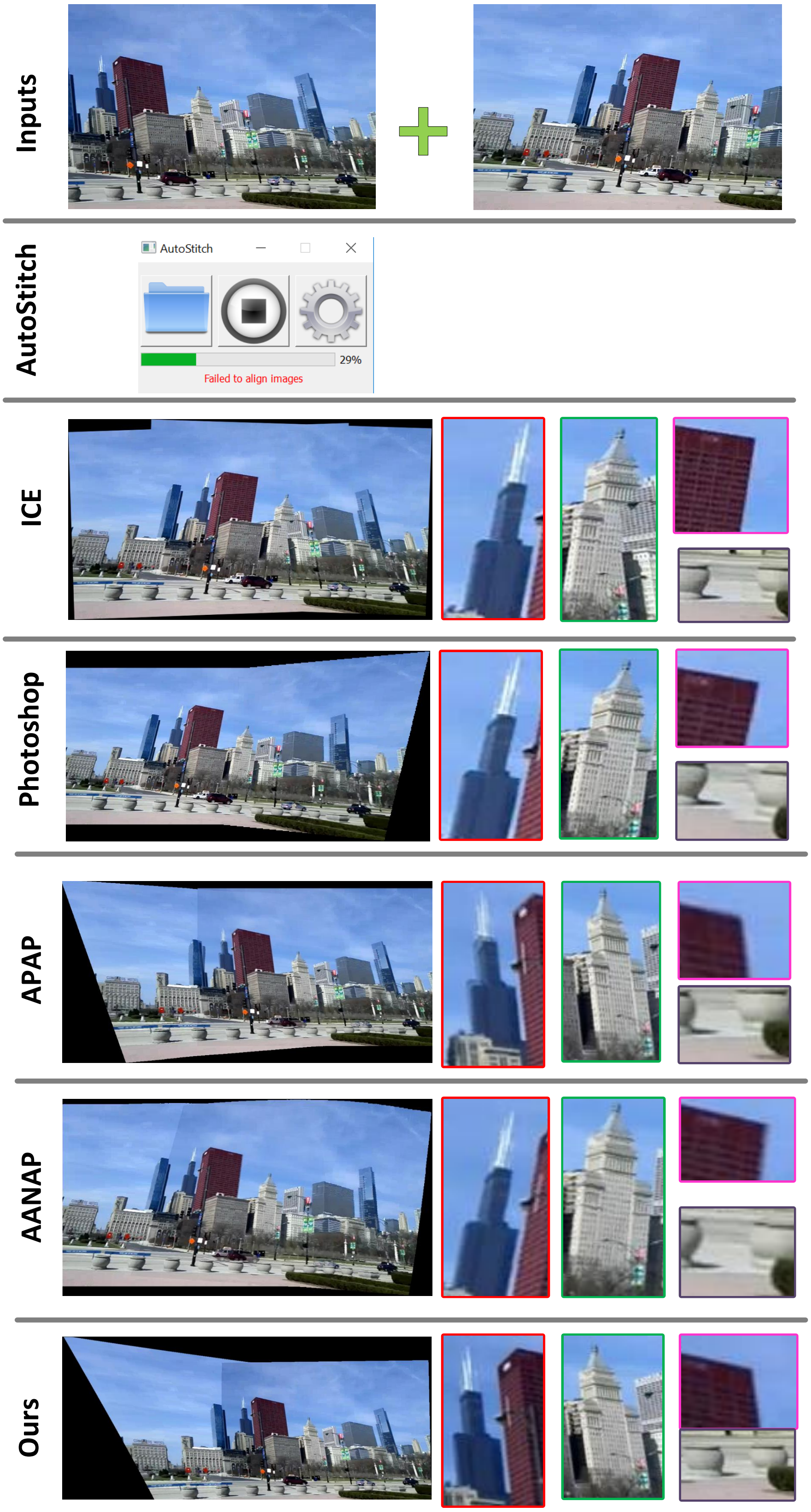


FIGURE 5.14: The evaluations of stitching qualities by varying the number of input point-matches.

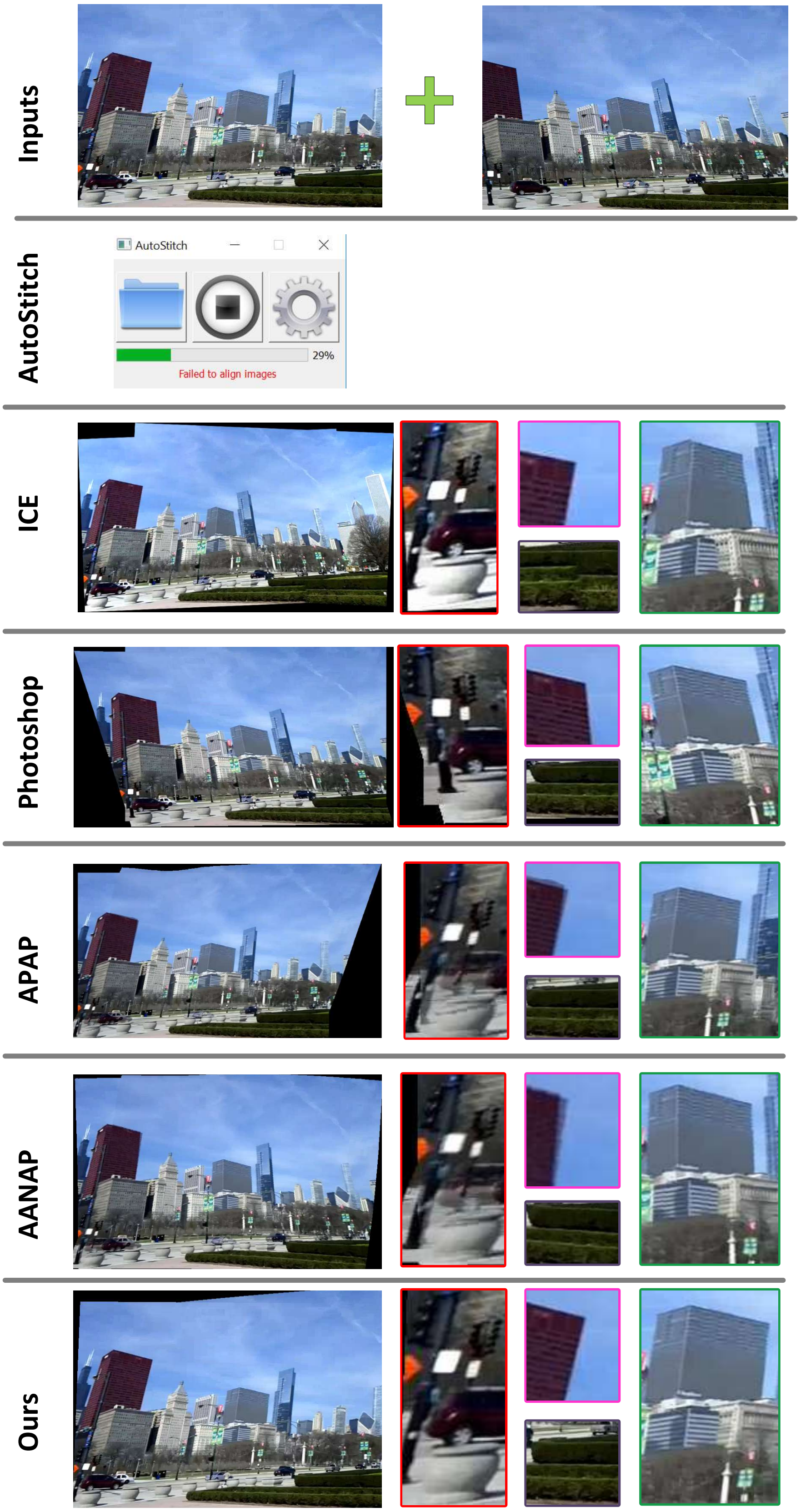
5.9 Discussion and Conclusion

The present work in this chapter is the first to address the homography for the RS case. We first defined a theoretical RS Homography matrix and proposed a 36pt solver to retrieve it from an image pair. Then we derived a simplified homography matrix and the associated 13pt minimal solver which is more suited for RANSAC based applications. The RS homography was successfully used in two well-known homography-based applications: relative pose estimation and image stitching. The experiment results show that the proposed method is superior to the state-of-the-art techniques and some well-known commercial image editing applications.



Failed to align images

FIGURE 5.15: An example of stitching real RS images from [Jia and Evans, 2012].



Failed to align images

FIGURE 5.16: An example of stitching real RS images from [Jia and Evans, 2012].

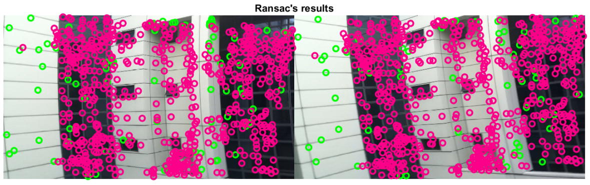


FIGURE 5.17: RANSAC results (inliers in pink, outliers in green) for a pair of real RS images from [Zhuang et al., 2017].

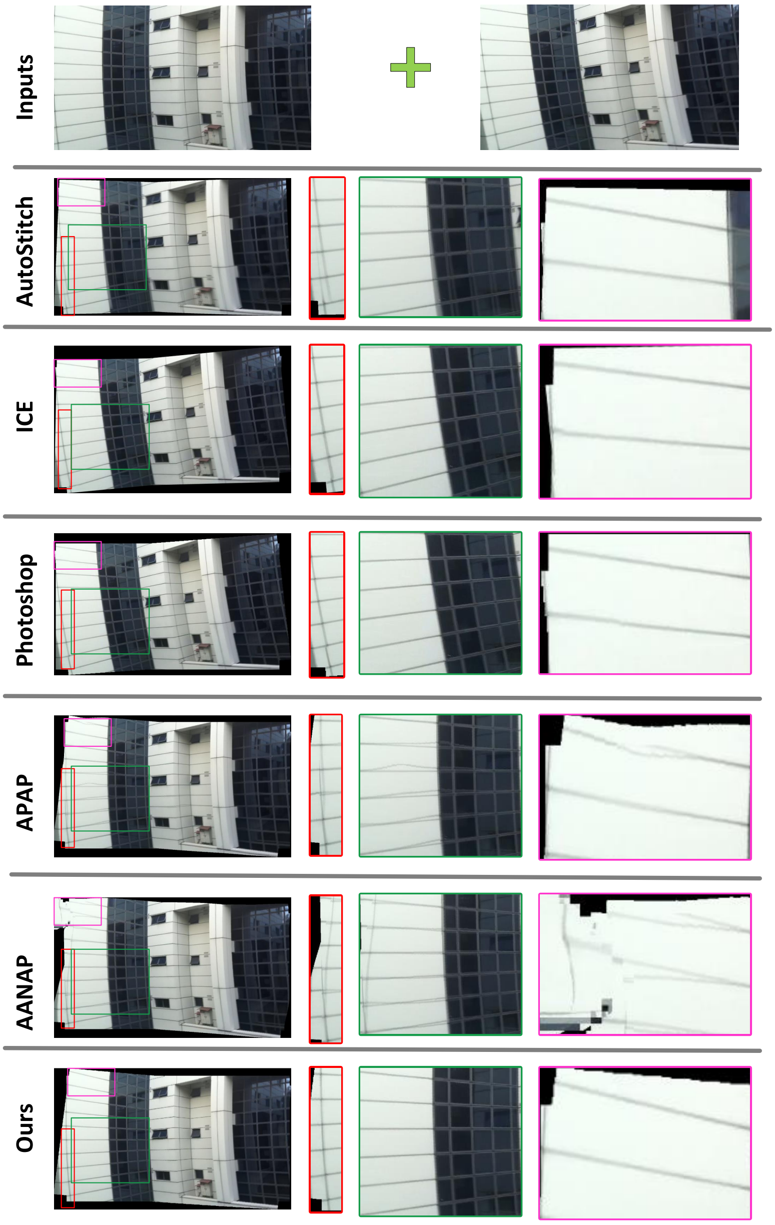


FIGURE 5.18: An example of stitching real RS images from [Zhuang et al., 2017].

Chapter 6

Analogies between RS and Non-rigid Vision

6.1 Introduction

When looking at an image which is strongly distorted by RS effects, one immediately imagines that the scene is deforming as if it was non-rigid. Thus, it seemed us natural to investigate the possibility to take advantage from theoretical background of non-rigid vision methods to solve RS vision problems.

In this chapter, We propose new methods for the absolute camera pose problem (PnP) and structure from motion (SfM) which handles Rolling Shutter (RS) effects by drawing **analogies** with non-rigid vision (Fig. 6.1). Unlike all existing methods which perform 3D-2D registration after augmenting the Global Shutter (GS) projection model with the velocity parameters under various kinematic models, we propose to use local differential constraints.

Estimating RS camera pose and instantaneous motion. The main idea of estimating RS camera pose and instantaneous motion simultaneously (RS-PInP) consists in considering that RS distortions due to camera instantaneous-motion during image acquisition can be interpreted as virtual deformations of a template captured by a GS camera.

Once the virtual deformations have been recovered using Shape-from-Template (SfT), the camera pose and instantaneous-motion are computed by registering the deformed scene on the original template. This 3D-3D registration involves a 3D cost function based on the Euclidean point distance, more physically meaningful than the re-projection error or the algebraic distance based cost functions used in previous work. By transforming the RS PnP problem into a 3D-3D registration problem, we show that our RS-PInP solution is more robust and stable than existing works [Albl et al., 2015] because the constraints to be minimized are more physically meaningful and are all expressed in the same metric dimension.

Results on both synthetic and real data show that the proposed method outperforms existing RS pose estimation techniques in terms of accuracy and stability of performance in various configurations.

RSSfM. We also propose a solution to the SfM problem for Rolling Shutter images (RSSfM) using an analogy with Non-Rigid SfM (NRSfM). This is an extension of the template-based analogy from a single view case to the multiple views and template-free case.

We propose the following two-step method for at least three RS images of an unknown scene. In the first step, RS distortions are used to compute one-to-one relative 3D deformations of points on the scene surface based only on their 2D measurements

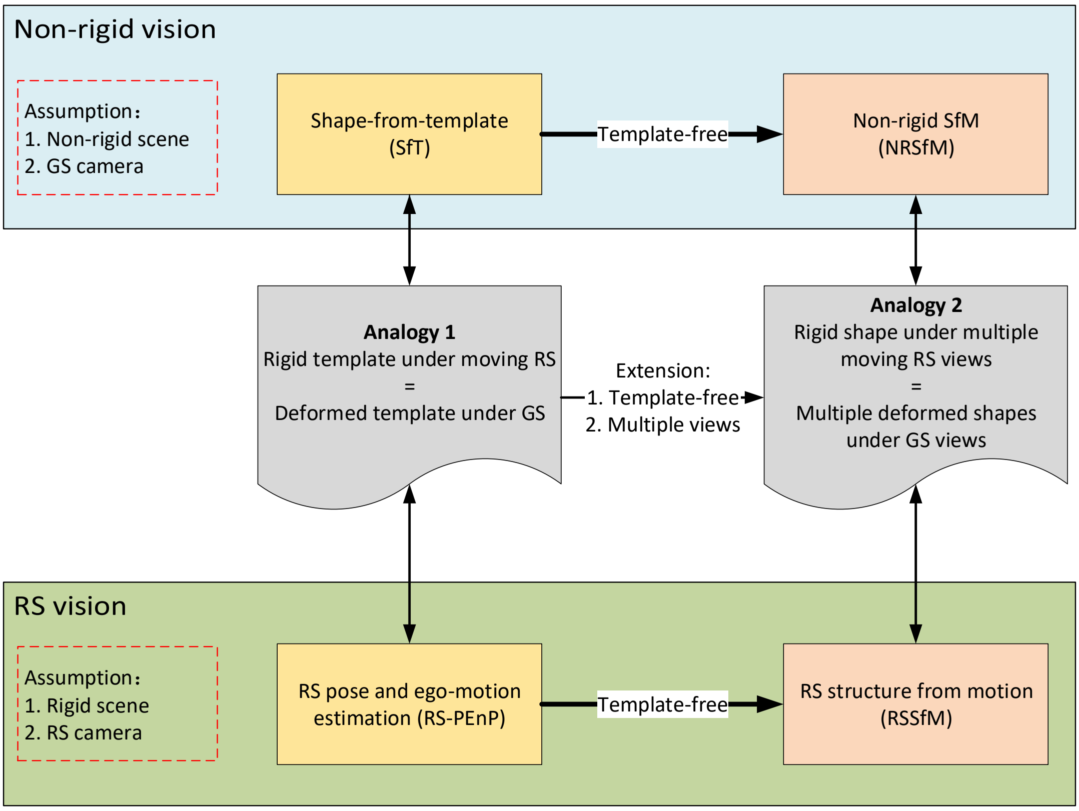

Non-rigid vision
Assumption:
1. Non-rigid scene
2. GS camera
Shape-from-template (SfT)
Template-free
Non-rigid SfM (NRSfM)
Analogy 1
Rigid template under moving RS
=
Deformed template under GS
Extension:
1. Template-free
2. Multiple views
Analogy 2
Rigid shape under multiple moving RS views
=
Multiple deformed shapes under GS views
RS vision
Assumption:
1. Rigid scene
2. RS camera
RS pose and ego-motion estimation (RS-PEnP)
Template-free
RS structure from motion (RSSfM)

FIGURE 6.1: Overview of the proposed RS-PInP and RSSfM method using analogies with non-rigid vision.

in at least three images. This is achieved thanks to the NRSfM method [Parashar et al., 2018]. This step consists in relaxing the RS constraints (kinematic and projection models) and treating the distorted images as GS ones obtained under 3D deformations. Then, a new algorithm is proposed to simultaneously retrieve the actual structure, the camera poses and instantaneous-motions that correctly explain image distortions using the RS projection model and a constant velocity instantaneous-motion assumption. This second step, which reintroduces the RS constraint, is carried out by minimizing a non-linear cost function summing squared distances between 3D points. A strategy is proposed for the initialization of shape and motion parameters based on Generalized Procrustes Analysis (GPA) [Bartoli et al., 2013].

Specifically, we show that this two step approach enables us to handle common singular configurations of classical RSSfM such as similar readout directions [Albl et al., 2016b]. We show experimentally that the proposed method outperforms state-of-art techniques.

Chapter outline. The rest of this chapter is organized as:

- We first review the related works about RS-PInP and RSSfM in section 6.2.
- We briefly introduce the two main techniques of non-rigid vision (section 6.3), namely SfT (section 6.3.1) and NRSfM (section 6.3.2).
- Then we show two analogies which build the links between RS-PInP problem and SfT (section 6.4.1), and between RSSfM and NRSfM (section 6.4.2).

- The details of the proposed RS-PInP method using SfT will be presented in section 6.5.
- The details of the proposed novel RSSfM method using NRSfM is introduced in section 6.6.
- Experiments are shown in section. 6.7 and section. 6.8.

6.2 Related Work

6.2.1 RS-PInP

One of the key issues in solving RS geometric problems is incorporating feasible camera instantaneous-motion into projection models. We give bellow the state-of-the-art works of RS-PInP:

- Saurer et al. [Saurer et al., 2015] propose a minimal solver to estimate RS camera pose based on the translation-only model with at least 5 3D-2D correspondences. However, this solution is limited to specific scenarios, such as a forward moving vehicle. It is not feasible for the majority of applications such as a hand-held camera, a drone or a moving robot, where instantaneous-rotation contributes significantly to RS effects [Hedborg et al., 2012, Duchamp et al., 2015].
- Albl et al. [Albl et al., 2016a] propose another minimal solver, which also requires at least 5 3D-2D matches. It is based on a uniform instantaneous-motion model. Nevertheless, it also requires the assistance of inertial measurement units (IMUs), which makes the algorithm dependent on additional sensors. Albl et al. also propose a minimal and non-iterative solution to the RS-PInP problem called R6P [Albl et al., 2015, Albl et al., 2019], which can achieve higher accuracy than the standard P3P [Haralick et al., 1991] by using an approximate doubly-linearized model. The approximation requires that the rotation between camera and world frames is small. Therefore, all 3D points need to be rotated first to satisfy the double-linearization assumption based on a rough estimation from IMU measurements or P3P. This pre-processing step makes R6P suffer from dependencies on additional sensors or the risk that P3P gives a non satisfactory rough estimate. Besides, R6P gives up to 20 feasible solutions and no flawless recipe is provided to choose the right one, which may lead to unstable performances.Recently, [Kukelova et al., 2018] presents new efficient solutions to the RS camera absolute pose problem. Unlike R6P, we approach the problem using simple and fast linear solvers in an iterative scheme.
- Magerand et al. [Magerand et al., 2012] present a polynomial projection model for RS cameras and propose the constrained global optimization of its parameters by means of a semidefinite programming problem obtained from the generalized problem of moments method. Contrarily to other methods, their optimization does not require an initialization and can be considered for automatic feature matching in a RANSAC framework. Unfortunately, the resolution is left to an automatic but computationally expensive solver.

In summary, a new efficient and stable solution to estimate the pose and instantaneous-motion of an RS camera under general motion, without the need for other sensors, is still absent from the literature. Such a solution is highly required for many potential applications.

6.2.2 RSSfM

The state-of-the-art works of RS-PInP are:

- The methods in [Hedborg et al., 2011, Zhuang et al., 2017, Im et al., 2018] use an RS video sequence to solve RSSfM by assuming smooth camera motion between every consecutive frames. This imposes a high acquisition framerate which results in high computational efficiency requirements. Unordered images with large baseline are not handled.

- The methods in [Ito and Okatani,] attempts to solve RSSfM by establishing an equivalence with self-calibrated SfM. Nevertheless, the method has strong constraints, namely a purely rotational motion, an affine camera and the availability of one image without RS effects.

- The degeneracies of RSSfM were pointed out in [Albl et al., 2016b]. It is explained that when the images are taken at positions with similar readout directions, bundle adjustment (BA) with the RS model fails to recover structure and motion. The proposed solution is simply to avoid these degenerate configurations, by taking images at positions with close to perpendicular readout directions.

In summary, a new robust and stable solution to solve RSSfM with unordered images and without overly restrictive assumptions on camera motion, readout direction or projection model is still absent from the literature. Such a solution would be an important step in the potential widespread deployment of 3D vision with RS imaging systems.

6.3 Non-rigid Vision

There are two main methods in deformable 3D reconstruction from images (Fig. 6.2). These go by the names of Shape-from-Template (SfT), Non-Rigid Structure from Motion (NRSfM).

6.3.1 Shape-from-Template

SfT refers to the task of template-based monocular 3D reconstruction, which estimates the 3D shape of a deformable surface by using different physics-based deformation rules [Salzmann and Fua, 2011, Bartoli et al., 2015]. Fig. 6.3 illustrates the geometric modeling of SfT. A 3D template $\tau \subset \mathbb{R}^3$ transforms to the deformed shape $S \subset \mathbb{R}^3$ by a 3D deformation $\Psi \in C^1(\tau, \mathbb{R}^3)$. If $\Omega \subset \mathbb{R}^2$ is a 2D space obtained by flattening a 3D template τ, thus, an unknown deformed embedding $\varphi \subset C^1(\Omega, \mathbb{R}^3)$ maps a 2D point $\mathbf{p} \in \Omega$ to $\mathbf{Q} \in S$. Finally, $\mathbf{Q}$ can be projected onto an image point $\mathbf{q} \in I$ by a known GS-based projection function Π^{GS}. The known transformation between Ω and I is denoted as η. It is obtained from 3D-2D point correspondences using Bsplines as in [Brunet et al., 2014]. The goal of SfT is to obtain the deformed surface S given $\mathbf{p}$, $\mathbf{q}$ and the first order derivatives of the optical flow at $\mathbf{p}$, namely $\frac{\partial \eta}{\partial p}(\mathbf{p})$.

The deformation constraints used to solve SfT can be categorized as:

Isometric deformation. The geodesic distances are preserved by the deformation [Bartoli et al., 2015, Salzmann and Fua, 2011, Brunet et al., 2014, Collins and Bartoli, 2015, Chhatkuli et al., 2017]. This assumption commonly holds for paper, cloth and volumetric objects.

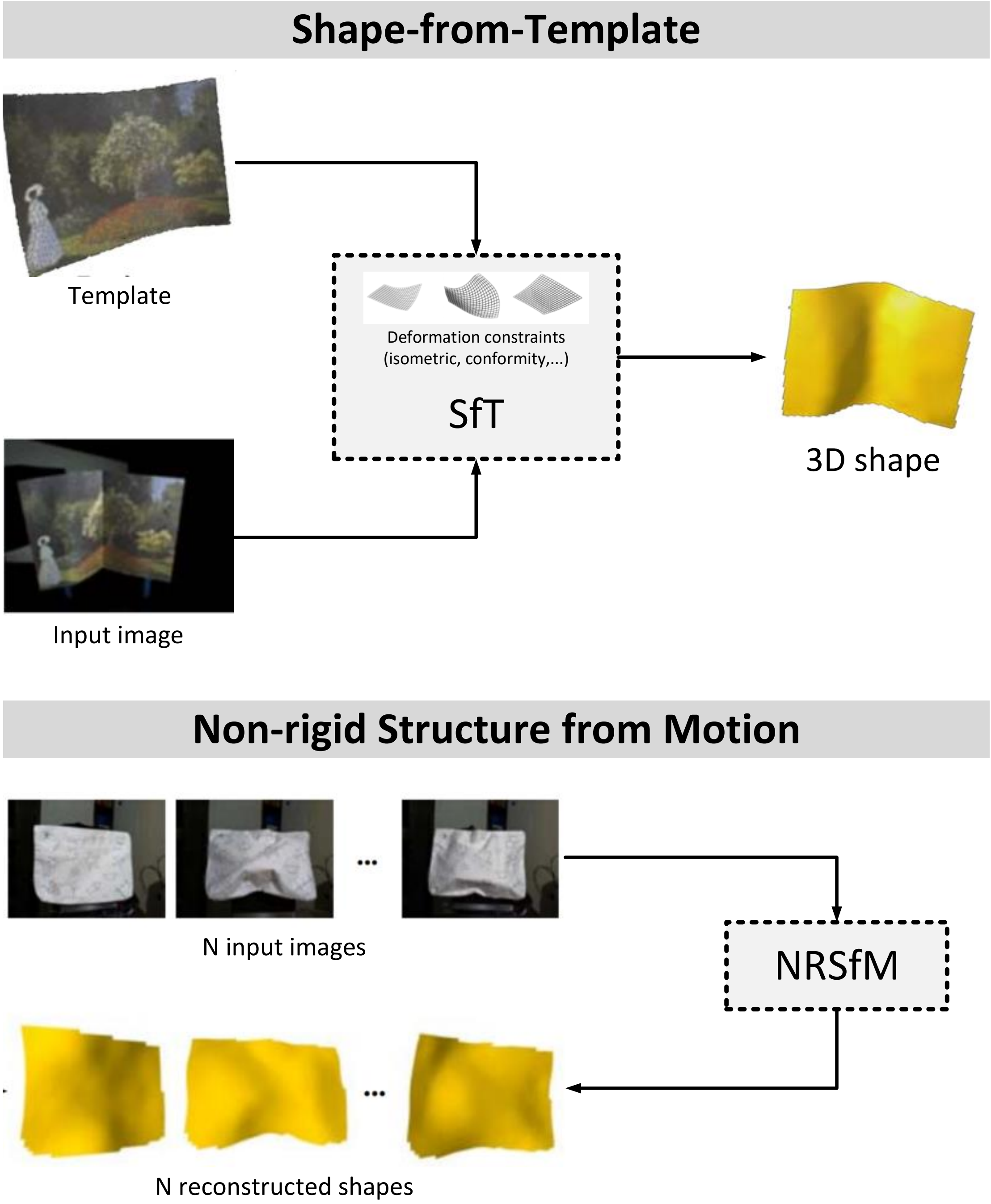


FIGURE 6.2: Two main monocular none-rigid scene 3D reconstruction techniques: Shape-from-Template (SfT) and Non-rigid structure from motion (NRSfM). Result extracted from [Chhatkuli et al., 2017].

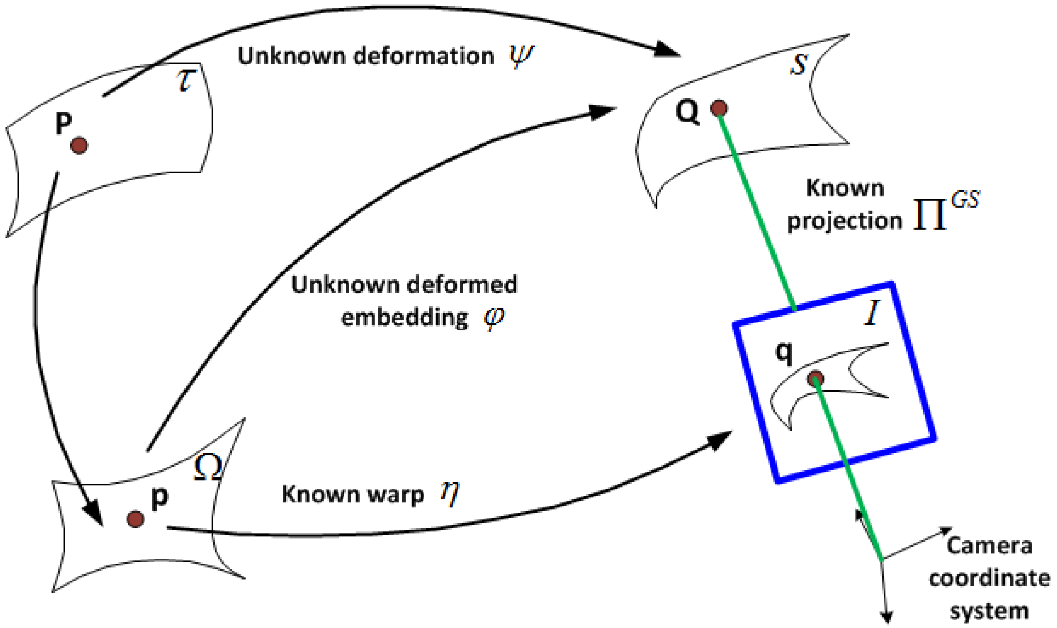

Unknown deformation ψ
τ
P
S
Q
Known projection Π^{GS}
Unknown deformed embedding φ
I
q
Ω
p
Known warp η
Camera coordinate system

FIGURE 6.3: Geometric modeling of SfT.

Conformal deformation. The isometric constraint can be relaxed to conformal deformation, which preserves angles since local scaling varies isotropically, and possibly handles isotropic extensible surfaces such as a balloon [Bartoli et al., 2015].

Elastic deformation. Linear [Malti et al., 2015,Malti and Herzet, 2017] or non-linear [Haouchine et al., 2014] elastic deformations are used to constrain extensible surfaces. Elastic SfT does not have local solution, in contrast to isometric SfT, and requires boundary condition to be available, as a set of known 3D surface points.

6.3.2 Non-rigid SfM

Conventional SfM allows the computation of a rigid object's 3D structure by given images of the object from different views. However, the rigidity constraints of SfM do not hold in many applications since real-world objects are more complex containing not only rigid motions but also non-rigid deformations, as well as their combination. Non-Rigid Structure-from-Motion (NRSfM) which uses multiple images of a deforming object to reconstruct its 3D, can solve such reconstruction problems.

The extension from SfM to the non-rigid case is by allowing the 3D points $\mathbf{P}_i$, to vary from frame to frame as follows:

$$\mathbf{P}_i^j = \begin{bmatrix} \mathbf{P}_i^1 & \mathbf{P}_i^2 & \dots & \mathbf{P}_i^m \end{bmatrix} \tag{6.1}$$

Where $\mathbf{P}_i^j$ is the location of i^{th} point at j^{th} frame. In order to reduce the ill-posedness of the NRSfM problem, a prior or regularization is often employed such as: *1)* the deformable model used (statistical [Bregler et al., 2000, Akhter et al., 2009, Gotardo and Martinez, 2011a] or physical [Varol et al., 2009, Taylor et al., 2010, Chhatkuli et al., 2018, Parashar et al., 2018]). *2)* the camera model (weak [Gotardo and Martinez, 2011a, Gotardo and

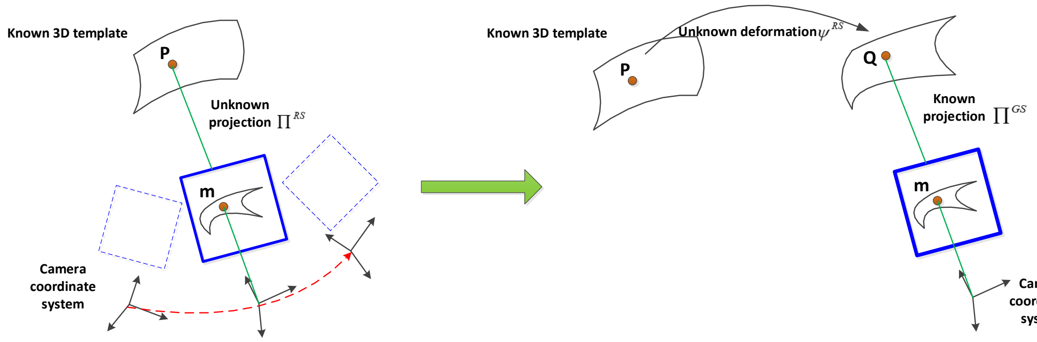

Known 3D template
P
Unknown projection Π^{RS}
m
Camera coordinate system
Known 3D template
Unknown deformation ψ^{RS}
P
Q
Known projection Π^{GS}
m
Camera coordinate system

FIGURE 6.4: **Analogy 1**: Equivalence between the RS projection of a rigid object (left) and a GS projection of a virtually deformed template (right).

Martinez, 2011c] or full perspective [Hartley and Vidal, 2008]). The fitness of the prior to deformation is a crucial element in successfully solving the NRSfM problem.

6.4 An Equivalence between RS Projection and Surface Deformation

In this section, we introduce two analogies between non-rigid vision and RS vision. The first analogy is between RS-PInP and SfT. The second is between RSSfM and NRSfM.

6.4.1 Analogy between RS-PInP and SfT

The main idea here is that distortions in RS images caused by camera instantaneous-motion can be expressed as the virtual deformation of a 3D shape captured by a GS camera. We first model the GS projection of a known 3D shape after a deformation Ψ:

$$\mathbf{m}_i = \Pi^{GS}(\mathbf{K}\Psi(\mathbf{P}_i)) \tag{6.2}$$

If we define the deformation as $\Psi^{RS}(\mathbf{P}_i) = \mathbf{R}(v_i)\mathbf{P}_i + \mathbf{t}(v_i)$, Eq. (6.2) becomes similar to Eq. (2.11):

$$\mathbf{m}_i = \Pi^{GS}(\mathbf{K}\Psi^{RS}(\mathbf{P}_i)) = \Pi^{GS}(\mathbf{K}(\mathbf{R}(v_i)\mathbf{P}_i + \mathbf{t}(v_i))) = \Pi^{RS}(\mathbf{K}\mathbf{Q}_i) \tag{6.3}$$

Analogy 1: Eq. (6.3) and Fig. 6.4 show that 3D shapes observed by a moving an RS camera are equivalent to corresponding deformed 3D shapes filmed by a GS camera.

We name this virtual corresponding deformation Ψ^{RS} as the ***virtual deformation*** and the virtually equivalent deformed shape $\Psi^{RS}(\mathbf{P}_i)$ as the ***virtual deformed shape***.

6.4.2 Equivalence between RSSfM and NRSfM

Here, we extend the previously defined analogy to the case with multiple RS images of an unknown shape. Namely, we consider different images of the same rigid surface taken by a moving RS camera as GS snapshots of a deforming 3D surface.

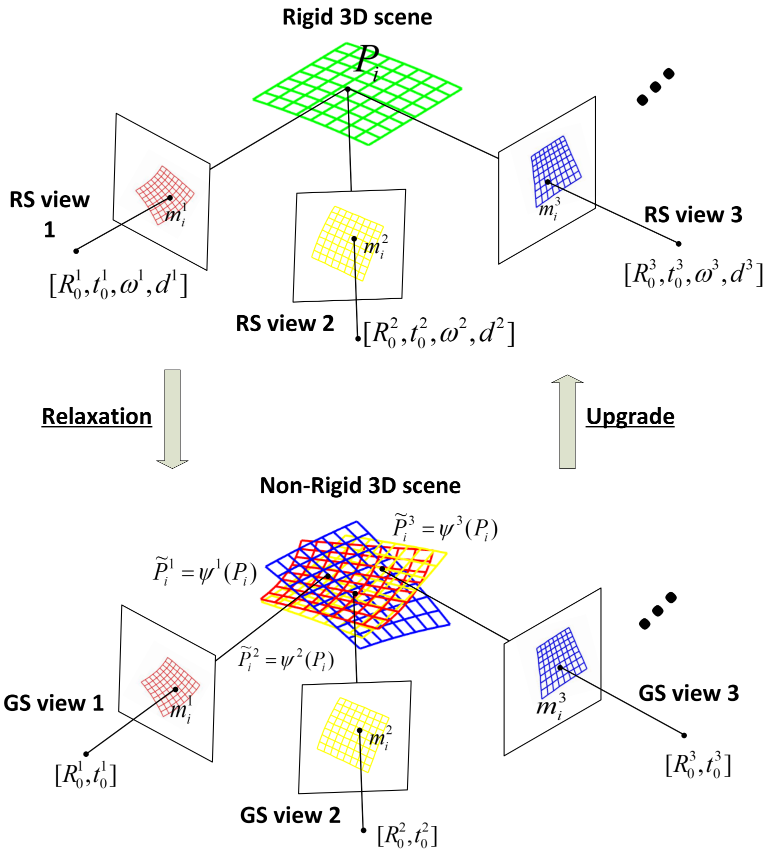


FIGURE 6.5: **Analogy 2**: Equivalence between a rigid 3D scene filmed by multiple RS cameras and a non-rigid scene filmed by multiple GS cameras.

We define ψ^j as a deformation that maps the original 3D structure $\mathbf{P}_i$ from world coordinates into camera coordinates directly. Then the RS projection described in Eq. (2.11) may be re-written as:

$$\mathbf{m}_i^j = \Pi^{RS}(\mathbf{P}_i^j) = (\Pi^{GS} \circ \psi^j)(\mathbf{P}_i) \tag{6.4}$$

Analogy 2: Eq. (6.4) and Fig. 6.5 show that a set of RS images $\mathbf{m}_i^j$ of a rigid scene may also be interpreted by the same scene under deformations ψ^j captured by multiple GS cameras.

Since the deformations are virtual, the 3D scene does not actually deform in the real world. Therefore, we called the original 3D shape $\mathbf{P}_i$ as ***actual structure***, the deformations ψ^j as the ***virtual deformations***, and the virtually deformed shape $\tilde{\mathbf{P}}_i^j = \psi^j(\mathbf{P}_i)$ as the ***virtual deformed shape***.

6.5 RS Pose and Instantaneous-motion Estimation using SfT

In this section, we introduce the proposed novel RS-PInP method, illustrated in Fig. 6.6, which first recovers the virtual template deformation using SfT and then computes the pose and velocity parameters using 3D-3D registration.

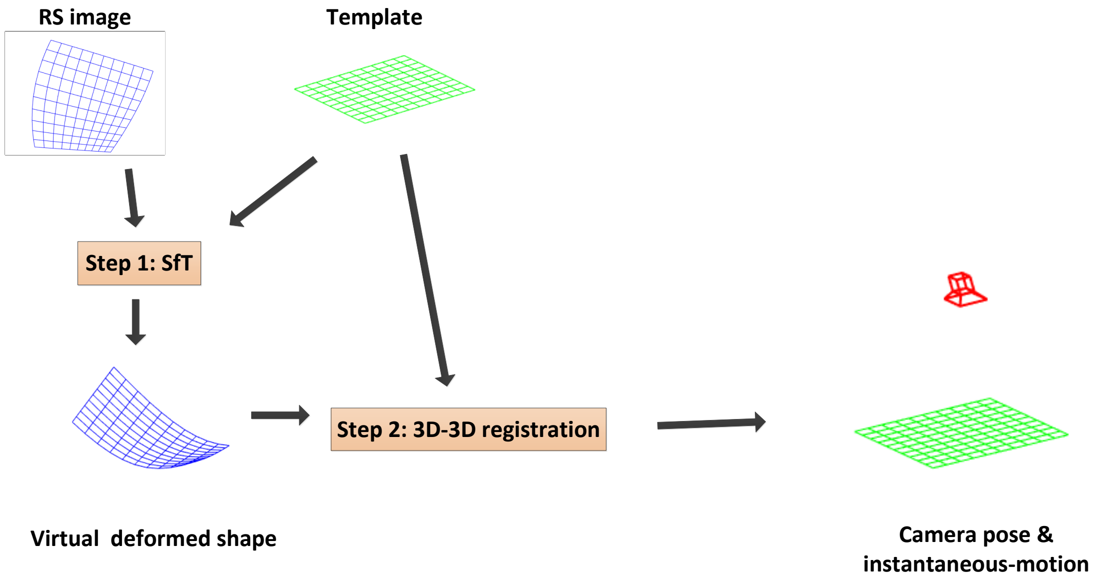


FIGURE 6.6: **An overview of the proposed pose and instantaneous-motion estimation method:** *Step 1:* Given an RS image and a known 3D template, we reconstruct the virtually deformed shape using SfT. *Step 2:* By performing 3D-3D registration between the virtually deformed shape and the template, RS camera pose and instantaneous-motion are obtained simultaneously.

6.5.1 Step 1: Reconstruction of the Virtual RS Deformed Shape

After showing the link between the RS-PInP and SfT problems, we focus on how to reconstruct the virtual deformed shape by using SfT. Since the assumption on the physical properties of the template plays a crucial role in solving the SfT problem we should determine which one of the deformation constraints can best describe the virtual deformation.

6.5.1.1 Equivalent RS Deformation under Different Instantaneous-motion Types

Any RS instantaneous-motion can be regarded as a combination of six atomic instantaneous-motions: translations along the X ($\mathbf{d}_x$), Y ($\mathbf{d}_y$), Z ($\mathbf{d}_z$) axes and rotations about the X ($\boldsymbol{\omega}_x$), Y ($\boldsymbol{\omega}_y$), Z ($\boldsymbol{\omega}_z$) axes. Fig. 6.7 shows RS images and virtual deformed shapes produced by different types of RS instantaneous-motions. Albl et al. [Albl et al., 2016a] and Rengarajan et al. [Rengarajan et al., 2017] illustrated four different types of RS effects (2D deformations) produced by camera instantaneous-motion. Differently, we focus on virtual 3D deformations instead. Fig. 6.7 also shows that the corresponding virtual deformations caused by different camera instantaneous-motions can be summarized into three types, by assuming a vertical scanning direction of the 3D template:

- *(i) Horizontal wobble:* Translation along the x-axis, rotation along the y-axis and z-axis creates surface wobble along the horizontal direction (perpendicular to the scan direction). In such cases, the distances are preserved only along the horizontal direction while the angles change during the deformation.
- *(ii) Vertical shrinking/extension:* Translation along the y-axis or rotation along the x-axis produce a similar effect, which shrinks or extends the 3D shape along the scan direction (vertical). This deformation preserves the distances along the horizontal direction but changes the angles. Thus, unlike an elastic deformation, stretching the

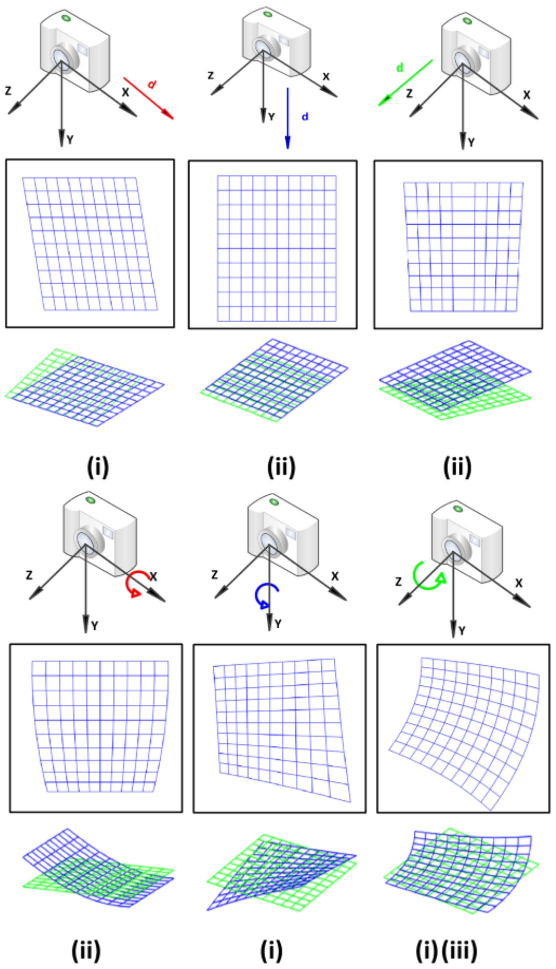


FIGURE 6.7: The 3D template shapes (green) captured by a RS camera under different atomic instantaneous-motions (first and third row) yield distorted RS images (second and fourth row). The exact same images are also obtained as the projection of the corresponding virtually deformed 3D shapes (blue) into a GS camera (third row). The type of corresponding virtual deformations are also given, see main text for details.

surface in the vertical direction will not introduce a compression in the horizontal direction.

- *(iii) Vertical wobble:* Beside horizontal wobble, rotation along the z-axis also leads to wobble in the vertical direction. The distances along the horizontal direction remain unchanged while the angles vary dynamically.

6.5.1.2 Choosing the Appropriate Physic Prior of SfT

It is important to notice that the virtual deformations do not follow any classical physics-based SfT surface models such as isometry, conformity or elasticity: isometric surface deformation preserves the distances along all directions while the virtual distortion only preserves the distances along the horizontal direction. The conformal deformation is a relaxation of the isometric model, which allows local isotropic scaling and preserves the angles during deformation. However, it cannot describe how the virtual deformation

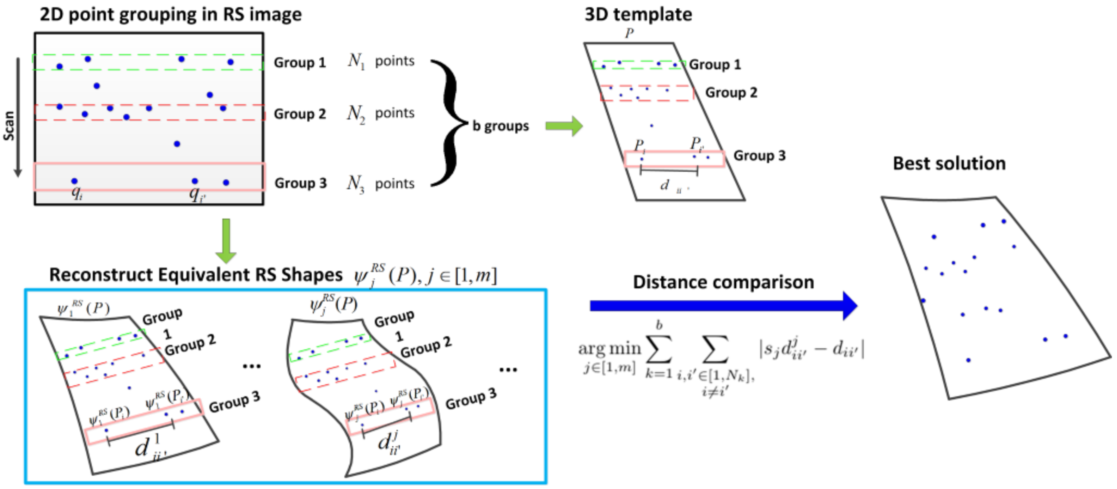


FIGURE 6.8: Choosing the best virtual shape from conformal SfT. .

angles change. The elastic surface stretches in one direction and generally produces extension in the orthogonal direction. In contrast, no shrinking or extension occurs along the horizontal direction during the virtual deformation.

We focus on reconstructing the virtual deformed shape based on the isometric and conformal deformations for the following reasons:

- The isometric constraint holds along the horizontal direction on the 3D shapes. Since the image acquisition time is commonly short, the effects of extension and compression of the 3D shape are limited, which makes the isometric model work in practice. Alternatively, the conformal model can reconstruct extensible 3D shapes [Bartoli et al., 2015]. Thus, the conformal model as a relaxation of the isometric model can be theoretically considered a better approximation to the virtual deformation.

- A complex virtual deformed shape will be produced if an RS camera is under general instantaneous-motion, composed of the six types of atomic motions. Therefore, different surface patches on the shape could be under different 3D deformations. Importantly, the isometric and conformal SfT solutions we used from [Bartoli et al., 2015] exploit **local** differential constraints and recover the local deformation around each point on the shape independently. The assumption we implicitly make is thus that the camera projection is GS in each neighbourhood. This turns out to be a very mild and valid assumption in practice.

- Analytical solutions to SfT using the isometric and conformal models are reported in [Bartoli et al., 2015], which are therefore faster and show the potential to form real-time applications [Collins and Bartoli, 2015]. In contrast, the existing solutions to the elastic model are made slower [Malti et al., 2015, Malti and Herzet, 2017] and require boundary conditions unavailable in RS-PInP.

Isometric deformation. Bartoli et al. showed that only one solution exists to isometric surface reconstitution from a single view and proposed the first analytical algorithm [Bartoli et al., 2015]. A stable solution framework for isometric SfT has been proposed later [Chhatkuli et al., 2017]. Thanks to the existing isometric algorithms, we can stably and efficiently obtain a single reconstruction of virtual deformed shape $\Psi^{RS}(\mathbf{P}_i)$.

Conformal deformation. Contrarily to the isometric case, conformal-based SfT theoretically yields a small, discrete set of solutions (at least two) and a global scale ambiguity [Bartoli et al., 2015]. Thus, we obtain multiple reconstructed virtual deformed shapes by using the analytical SfT method under the conformal constraint. However, only one reconstruction is close to the real virtual deformed shape $\Psi^{RS}(\mathbf{P}_i)$. Therefore, we pick up the most practically reasonable reconstruction based on distance preservation along the horizontal direction.

We assume that a total of m reconstructed shapes $\left\{\Psi_j^{RS}(\mathbf{P}), \quad j = 1, 2..., m\right\}$ are obtained. As shown in Fig. 6.8 the 2D points located close to each other in the scanning direction in the image are segmented into b groups $\mathbb{G}_k, k \in [1, b]$ of N_k points. In the experiments, we group two 2D points into the same group if their difference of row index is lower than a threshold (experimentally set as 5 pixels). Then, we calculate a global scale factor s_j of each reconstructed virtual deformed shape to the template by using $s_j = \frac{2}{n(n-1)} \sum_{i,i' \in [1,n], i \neq i'} d_{ii'} / d^j_{ii'}$, where $d_{ii'}$ is the euclidean distance between 3D points $\mathbf{P}_i$ and $\mathbf{P}_{i'}$ and $d^j_{ii'}$ is the euclidean distance of the corresponding reconstructed 3D points $\Psi_j^{RS}(\mathbf{P}_i)$ and $\Psi_j^{RS}(\mathbf{P}_{i'})$. We choose $i, i' \in [1, n]$ randomly and calculate the average value. Finally, we choose the reconstruction $\Psi_j^{RS}(\mathbf{P})$ with the smallest sum of distance differences along the horizontal direction between each virtual deformed shapes ${}^x d^j_{ii'}$ and known 3D template ${}^x d_{ii'}$ as the best solution:

$$\underset{j \in [1,m]}{\arg\min} \sum_{k=1}^{b} \sum_{\substack{i,i' \in [1,N_k], \\ i \neq i'}} |s_j {}^x d^j_{ii'} - {}^x d_{ii'}| \tag{6.5}$$

6.5.2 Step 2: Camera Pose and Instantaneous-motion Computation

6.5.2.1 Instantaneous-motion model

Since the acquisition time of a frame is commonly short, one can generally assume a uniform kinematic model (with constant translational and rotational velocities). Moreover, by considering small rotation angles, we obtain the so-called linearized model, which has been used in many applications [Magerand et al., 2012, Albl et al., 2015, Dai et al., 2016, Albl et al., 2016b] as shown in Eq. (2.14).

6.5.2.2 3D-3D Registration

After obtaining the virtual shape $\Psi^{RS}(\mathbf{P})$, we register the virtually deformed shape to the known 3D template $\mathbf{P}$ using the RS instantaneous-motion model. By iteratively minimizing the distance errors between the known 3D template and the reconstructed virtual shape, we obtain the camera pose and instantaneous-motion parameters simultaneously:

$$\underset{\mathbf{R}_0, \mathbf{t}_0, \boldsymbol{\omega}, \mathbf{d}}{\arg\min} \sum_{i=1}^{n} \left\| \mathbf{R}(v_i)\mathbf{P}_i + \mathbf{t}(v_i) - \Psi^{RS}(\mathbf{P}_i) \right\| \tag{6.6}$$

Actually, we slightly abused the term 'registration' to mean that the 3D points of the virtually deformed surface are fitted with the corresponding 3D points of the template. This can be seen as a registration where the recovered parameters are not a mere rigid transformation but a local motion with constant velocity.

Initialization: we initialize the parameters in Eq. (6.6) as follows:

- $\mathbf{R}_0$ and $\mathbf{t}_0$ are initialized using a classical PnP method [Haralick et al., 1991].
- The instantaneous-motion parameters ($\boldsymbol{\omega}$,$\mathbf{d}$) are initialized by the following two steps: (1) Group image points into sets of vertically close points (so that the RS effect can be neglected) and run P3P for each set. (2) Initialize $\mathbf{d}$ and $\boldsymbol{\omega}$ by computing the relative translation and rotation between groups and dividing by the scan time. Alternatively, we can operate in the same procedure by grouping the points of the deformed surface into subsets of close 3D points, which are registered by 3D point could transformations [Horn et al., 1988].

However, in many practical situations, it is more convenient and more efficient to set the initial values of $\mathbf{d}$ and $\boldsymbol{\omega}$ to 0, which in our experiments always allowed convergence toward the correct solution.

6.6 Solving RSSfM Using NRSfM

In this section, we introduce the proposed RSSfM method, illustrated in Fig. 6.9, which first recovers the virtual deformed structure for each input RS image using NRSfM and then computes the actual structure, camera pose and instantaneous-motion using 3D-3D RS registration.

6.6.1 Step 1: Reconstruction of the one-to-one virtual deformed shapes

Choosing the Appropriate NRSfM Method. NRSfM aims to recover the 3D shapes of an object under deformation from a set of 2D GS images. Thus, it allows us to reconstruct the virtual deformed shapes $\tilde{\mathbf{P}}_i^j$ for every RS image. Various NR-SfM methods have been presented over the last two decades. For example, [Hu et al., 2013] requires no missing data while [Agudo and Moreno-Noguer, 2015,Agudo et al., 2016] require rigid motion at the beginning of the sequence. [Akhter et al., 2009,Gotardo and Martinez, 2011b] require smooth video sequences. These assumptions do not hold with unordered RS image sets. Besides, some piece-wise methods [Varol et al., 2009,Taylor et al., 2010,Russell et al., 2014] require a segmentation of the image domain into regions, which may be costly with large input datasets or unavailable.

However, as our discussion in section 6.3.2, not all of them are suitable for RSSfM. In this paper, we use isometric NRSfM (Iso-NRSfM) [Parashar et al., 2018] to reconstruct the virtual deformed shapes for the following reasons:

1. Isometry is a good constraint to model the virtual deformation [Lao et al., 2018b].
2. It handles missing data due to occlusions and unordered input images. *3)* It requires $m \geq 3$ views with linear complexity in the number of views and points, and thus combines the use of minimal data with higher efficiency than the other NRSfM methods.

General isometric NRSfM. The Iso-NRSfM method models the object's 3D shape for each image by a Riemannian manifold and deformations as isometric mappings. Each manifold is parameterized by embedding the corresponding retinal plane. This modeling allows one to reason on the metric tensor and Christoffel Symbols, directly in retinal coordinates, and in relationship to the inter-image warps, which can be computed from point-matches between images. Based on the theorem that the metric tensor and Christoffel Symbols may be transferred between views using only the warps, a system of

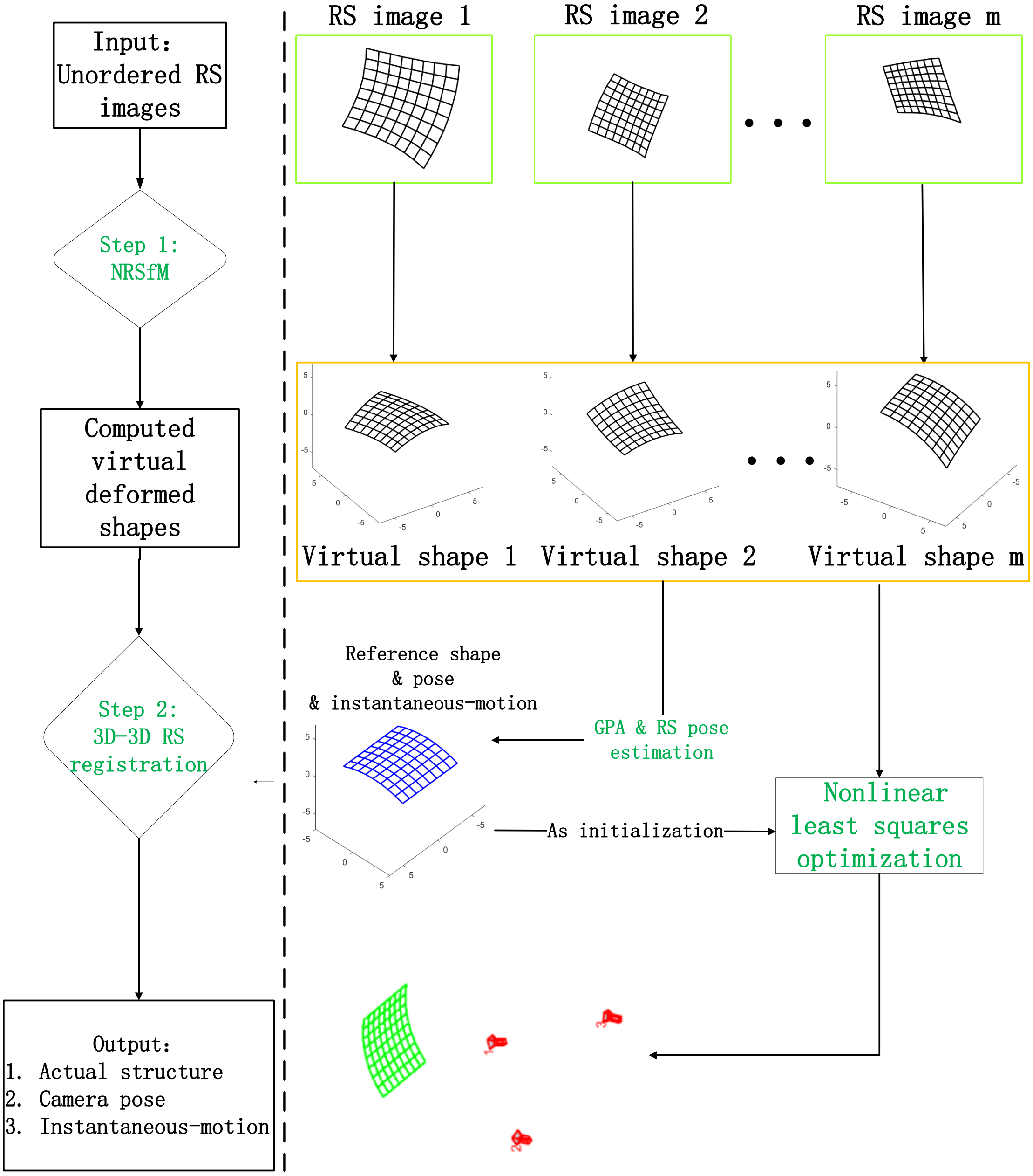


FIGURE 6.9: **Overview of the proposed RSSfM method.** *Step 1:* Given multiple RS images, the virtually deformed shape is reconstructed for each image using NRSfM. *Step 2:* By performing an iterative 3D-3D RS registration using Generalized Procrustes Analysis and RS pose estimation as initialization, the actual structure, camera pose and instantaneous-motion are obtained simultaneously.

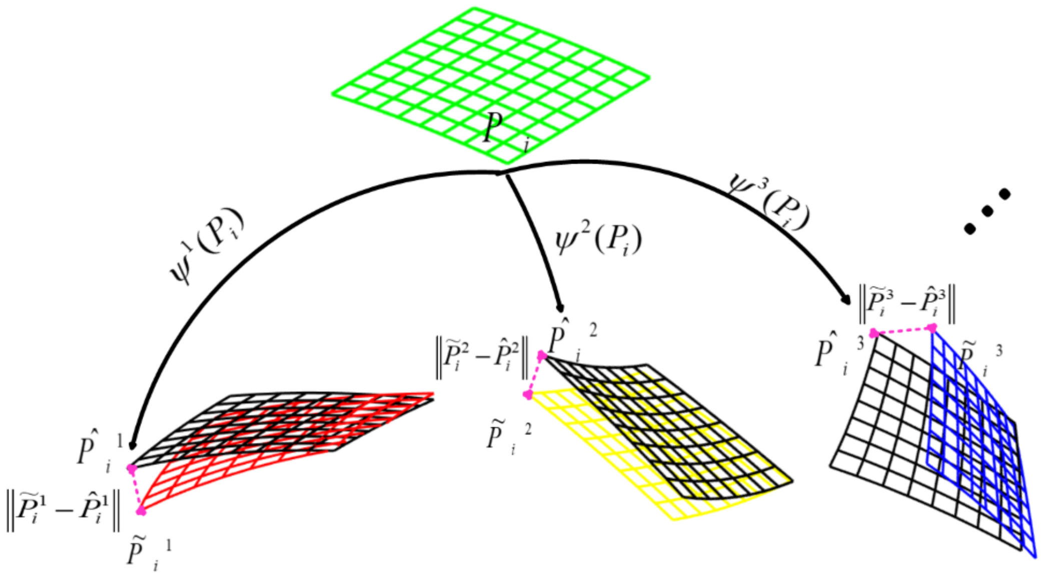


FIGURE 6.10: The 3D-3D registration, recovers the actual shape $\mathbf{P}_i$ (green) by minimizing the sum of squares of the distance differences between re-deformed shapes (black) $\hat{\mathbf{P}}_i^j$ and the virtual deformed shapes $\tilde{\mathbf{P}}_i^j$ (red, yellow and blue) recovered by the NRSfM.

two quartics in two variables that involves up to second order derivatives of the warps can be created for an infinitesimally planar surface at each point. The solution of this system are normals of the surface in all views. The shapes can finally be recovered by integrating the normal fields for each view.

Isometric NRSfM with the infinitesimal planarity assumption. In infinitesimal planarity, one assumes that a surface is at each point locally planar. Thus the surface is globally curved and represented infinitesimally by a set of planes. Since we assume the linearized model for RS instantaneous-motion, the virtual deformations are quasi continuous and smooth in the case of wobble, shrinking and extension [Lao et al., 2018b], which can thus be interpreted by infinitesimal planarity. The general solution for Iso-NRSfM uses the solution with infinitesimal planarity as initialization. However, infinitesimal planarity (InfP-NRSfM) also alone gives good results while being much faster than the general algorithm. Therefore, we compare the use of both Iso-NRSfM and InfP-NRSfM to reconstruct the virtual deformed shapes in the experiments.

6.6.2 Step 2: Recovering the Actual Shape and Cameras

6.6.2.1 3D-3D Registration

After obtaining virtual deformed shapes $\tilde{\mathbf{P}}_i^j$ for each view, we have to estimate the actual shape and camera pose and instantaneous-motion. However, the transformations from the actual shape to the virtual deformed shapes are non-rigid. Therefore, as shown in Fig. 6.10, in order to estimate an accurate actual shape, pose and instantaneous-motion of each view, we design a specific 3D-3D registration by minimizing the sum of squares of the distance difference between re-deformed shapes $\hat{\mathbf{P}}_i^j$ and the virtual deformed shapes $\tilde{\mathbf{P}}_i^j$ recovered by NRSfM:

$$\begin{aligned} &\arg\min_{\beta} = \sum_{j=1}^{m}\sum_{i=1}^{n} V_i^j \left\|\tilde{\mathbf{P}}_i^j - \hat{\mathbf{P}}_i^j\right\|^2 \\ \text{with} \quad &\beta = \left\{\{\mathbf{P}_i\}, \left\{\mathbf{R}_0^j\right\}, \left\{\mathbf{t}_0^j\right\}, \left\{\omega^j\right\}, \left\{d^j\right\}\right\} \\ &\hat{\mathbf{P}}_i^j = \psi^j(\mathbf{P}_i), \end{aligned} \tag{6.7}$$

where $V_i^j \in [0,1]$ indicates if a 3D point $\mathbf{P}_i$ is visible in the j^{th} image or not. The deformation function ψ^j is constrained by the RS instantaneous-motion model. Various models have been presented in existing work such as SLERP [Hedborg et al., 2012], Rodrigues formula [Ait-Aider et al., 2006] for the rotation and the constant accelerated translation [Zhuang et al., 2017]. The proposed 3D-3D RS registration can easily equip with different RS instantaneous-motion models. Here, we use a constant velocity model (Eq. (2.14) which is a good compromise between accuracy and complexity and is widely used in previous RS works [Magerand et al., 2012, Dai et al., 2016, Albl et al., 2015, Albl et al., 2016b, Lao et al., 2018b]:

$$\psi^j(\mathbf{P}_i) = (\mathbf{I} + [\boldsymbol{\omega}^j]_\times v_i^j)\mathbf{R}_0^j\mathbf{P}_i + \mathbf{t}_0^j + \mathbf{d}^j v_i^j \tag{6.8}$$

The cost function in Eq. (6.7) being non-linear, the availability of a good initial guess for the actual surface points, camera pose and instantaneous-motion is crucial to ensure convergence toward the solution. This is addressed in the next section.

6.6.2.2 Shape, Pose and Instantaneous-Motion Initialization

We propose to use Generalized Procrustes Analysis (GPA) and RS pose estimation. GPA solves the problem of registering more than two observed shape data [Dryden et al., 1998]. In this problem, a reference shape which should be as similar as possible to all observed shapes and one global transformation per observed shape has to be computed. In RSSfM, we assume that the deformations of a given actual point $\mathbf{P}_i$ in all images can be approximated by a random noise. Thus the actual scene could be close to the 'average' shape among all the virtual deformed shapes. We can then roughly estimate the actual scene $\{\mathbf{P}_i\}$ by performing GPA using the virtual deformed shapes $\tilde{\mathbf{P}}_i^j$ as observed shapes. Then using RS pose computation [Lao et al., 2018b] from this rough computed actual scene and the RS images, we find the global camera pose $\left\{\mathbf{R}_0^j\right\}$, $\left\{\mathbf{t}_0^j\right\}$ and instantaneous-motion $\{\boldsymbol{\omega}^j\}$, $\{\mathbf{d}^j\}$ to initialize the optimization in Eq. (6.7).

6.6.2.3 Planar Degeneracy

The combination of NRSfM and the RS constraints makes the proposed two-step method work in the common degenerate configurations of RSSfM. An intuitive explanation to this desirable property is as follows. First, NRSfM reconstructs consistent virtually deformed shapes by considering that the viewed surface is locally smooth and differentiable. This is a convenient prior on the scene structure which, though widely applicable, is not used by any other RSSfM methods. Once the 3D surfaces are reconstructed for each image, the RS assumption serves to constrain the pose and instantaneous-motion parameters to be compatible with these while the degeneracy was already resolved at the first step.

We explain how using the 3D-3D error to recover the scene structure and camera motion instead of the reprojection error enables us to fix the degenerate configuration uncovered in [Albl et al., 2016b]. It is stated that any number of RS images with parallel readout directions can be explained by a planar scene. Bundle adjustment with the linearized RS model (RSBA) always converges toward this trivial solution. However this case is not degenerate for the proposed 3D-3D method.

We assume without loss of generality that an RS camera has the pose $\mathbf{R}_0 = \mathbf{I}$ and $\mathbf{t}_0 = [0,0,0]^\top$, while the ground-truth of the instantaneous-motion is $\boldsymbol{\omega}^{\mathrm{GT}}$ and $\mathbf{d}^{\mathrm{GT}}$. According to Eq. (2.11) and (6.8), a 3D point $\mathbf{P}_i^{\mathrm{GT}} = [X,Y,Z]^\top$ projects as $\mathbf{m}_i = [u_i, v_i]^\top = \Pi^{GS}((\mathbf{I} + [\boldsymbol{\omega}]_\times v_i)\mathbf{P}_i + \mathbf{d}v_i)$. RSBA minimizes the sum of squares of the re-projection errors:

$$\mathbf{e}_i = \mathbf{m}_i - \Pi^{GS}((\mathbf{I} + [\boldsymbol{\omega}]_\times v_i)\mathbf{P}_i + \mathbf{d}v_i) \tag{6.9}$$

In our method however, the first step using NRSfM does not have degeneracies [Parashar et al., 2018]. After obtaining the equivalent deformed shape $\tilde{\mathbf{P}}_i^j = (\mathbf{I} + [\boldsymbol{\omega}^{\mathrm{GT}}]_\times v_i)\mathbf{P}_i^{\mathrm{GT}} + \mathbf{d}^{\mathrm{GT}} v_i$, the second step uses the 3D-3D re-deformation error instead:

$$\mathbf{e}_i = \tilde{\mathbf{P}}_i - \hat{\mathbf{P}}_i = \tilde{\mathbf{P}}_i - ((\mathbf{I} + [\boldsymbol{\omega}]_\times v_i)\mathbf{P}_i + \mathbf{d}v_i) \tag{6.10}$$

Obviously, both Eq. (6.9) and (6.10) vanish for the correct configuration $\{\mathbf{P}_i = \mathbf{P}_i^{\mathrm{GT}}, \boldsymbol{\omega} = \boldsymbol{\omega}^{\mathrm{GT}}, \mathbf{d} = \mathbf{d}^{\mathrm{GT}}\}$. However, if we alter the 3D scene and camera to the configuration $\{\mathbf{P}_i = [X,0,Z]^\top, \boldsymbol{\omega} = [-1;0;0]^\top, \mathbf{d} = [0,0,0]^\top\}$, Eq. (6.9) still vanish, while Eq. (6.10) does not. This means that the RS images could be explained by projecting the 3D scene to the plane $Y = 0$ with the specific instantaneous-motion ($\boldsymbol{\omega} = [-1;0;0]^\top$). However, this ambiguity does not occur for the proposed 3D-3D RS registration.

6.7 Experiments of RS-PInP

In our experiments, the analytical isometric solution and analytical conformal solution are used to reconstruct the virtual deformed shape from RS images of both synthetic and real planar and non planar templates under isometric and conformal constraints respectively. The Levenberg-Marquardt algorithm is used in the non-linear pose and instantaneous-motion estimation from Eq. (6.6).

We compare the proposed methods to two state-of-the-art camera pose approaches:

- **AnIRS:** The analytical isometric solution [Chhatkuli et al., 2017][1].
- **AnCRS:** The analytical conformal solution (**AnCRS**) [Bartoli et al., 2015][4].
- **GS-PnP:** GS PnP solution [2] [Gao et al., 2003].
- **RS-PnP:** The RS-PInP solution[3] which uses R6P [Albl et al., 2015].

6.7.1 Synthetic Data

We simulated a calibrated pin-hole camera with 640 × 480 px resolution and 320 px focal length. The camera was located randomly on a sphere with a radius of 20 units and was pointing to a simulated surface (10 × 10 units) with varying average scanning angles from 0 to 90 deg. We drew n points on the surface to form the 3D template. Random Gaussian noise with standard deviation σ was also added to the 2D projected points $\mathbf{m}$.

[1] http://igt.ip.uca.fr/∼ab/Research

[2] estimateWorldCameraPose function in MATLAB

[3] http://cmp.felk.cvut.cz/∼alblcene/r6p

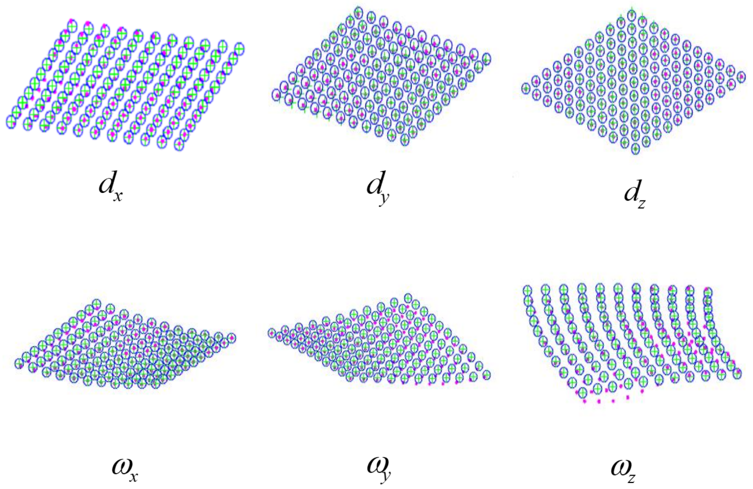


FIGURE 6.11: Reconstructed virtual deformed shapes by **AnIRS** (magenta points) and **AnCRS** (green crosses) compared to ground truth structure (blue circles) under six types of camera instantaneous-motion.

TABLE 6.1: Mean ($|e_I|$, $|e_C|$) and standard deviation (σ_I, σ_C) of reconstruction errors (expressed in units) of the virtual deformed shape by **AnIRS** and **AnCRS** under six types of camera instantaneous-motions.

	d_x	d_y	d_z	ω_x	ω_y	ω_z
$\lvert e_I \rvert$	0.0130283	0.0113629	**0.0001183**	0.0023273	0.0020031	0.1338190
$\lvert e_C \rvert$	**0.0040963**	**0.0052104**	0.0009037	**0.0000921**	**0.0008493**	**0.0008417**
σ_I	0.0001810	0.0000943	**0.0000014**	0.0000834	0.0007209	0.0393570
σ_C	**0.0000318**	**0.0000529**	0.0000310	**0.0000206**	**0.0003639**	**0.0001201**

6.7.1.1 Recovering the virtual deformed shape.

We first evaluate the reconstruction accuracy of **AnIRS** and **AnCRS** on the virtual deformed shape. Since the types of deformation depend on the type of RS instantaneous-motion, we measure the mean and standard deviation of distances between the reconstructed 3D points and the corresponding points on the 3D template under six atomic instantaneous-motion types (section 6.5.1.1). For each motion type, we run 200 trials to obtain statistics. We varied the number of 3D-2D matches from 10 to 121 and used a noise level $\sigma = 1$ px. At each trial, the instantaneous-motion speed was randomly set as follows: translational speed varying from 0 to 3 units/frame and rotational speed varying from 0 to 20 deg/frame.

The results in Fig. 6.11 show that both **AnIRS** and **AnCRS** provide stable and high accuracy results for the virtual deformed shape reconstruction. The quantitative evaluation in Table 6.1 demonstrates that **AnCRS** generally performs better than **AnIRS**. Specifically, it indicates that the advantages of **AnCRS** are significant in the cases of instantaneous-rotation along the x or z-axis. The only exception is in translation along the z-axis, where the virtual deformation is with relatively smaller extension/shrinking than other types. Thus, **AnIRS** gives better results than **AnCRS**. However, all observations confirm our analysis in section 6.5.1.1 that conformal surfaces can better model the extensibility of virtual deformation generally.

6.7.1.2 Pose estimation.

We compared **AnIRS** and **AnCRS** in camera pose estimation with both **GS-PnP** and **RS-PnP**. Since the ground truth of camera poses are known, we measured the absolute error of rotation (deg) and translation (units).

Accuracy vs instantaneous-motion speed. We fixed the number of 3D-2D matches to 60 and noise level to $\sigma = 1$ px. We increased the translational speed and rotational speed from 0 to 3 units/frame and 20 deg/frame gradually. At each configuration, we run 100 trials with random velocity directions and measured the average pose errors. The results in Fig. 6.12(a,b) show that both **AnIRS** and **AnCRS** provide significantly more accurate estimates of camera orientation and translation with all instantaneous-rotation configurations (ω_x, ω_y and ω_z) compared to **GS-PnP** and **RS-PnP**. Under three instantaneous-translations, **AnIRS** and **AnCRS** show an obvious superiority in camera rotation estimation, and perform slightly better in camera translation estimation than **RS-PnP**. As expected, GS-based **GS-PnP** fails in pose estimation once the instantaneous-motion is strong. In contrast, **RS-PnP** achieves better results in translation than **GS-PnP**, but both of them provide an inaccurate estimate for camera rotation to the same extent.

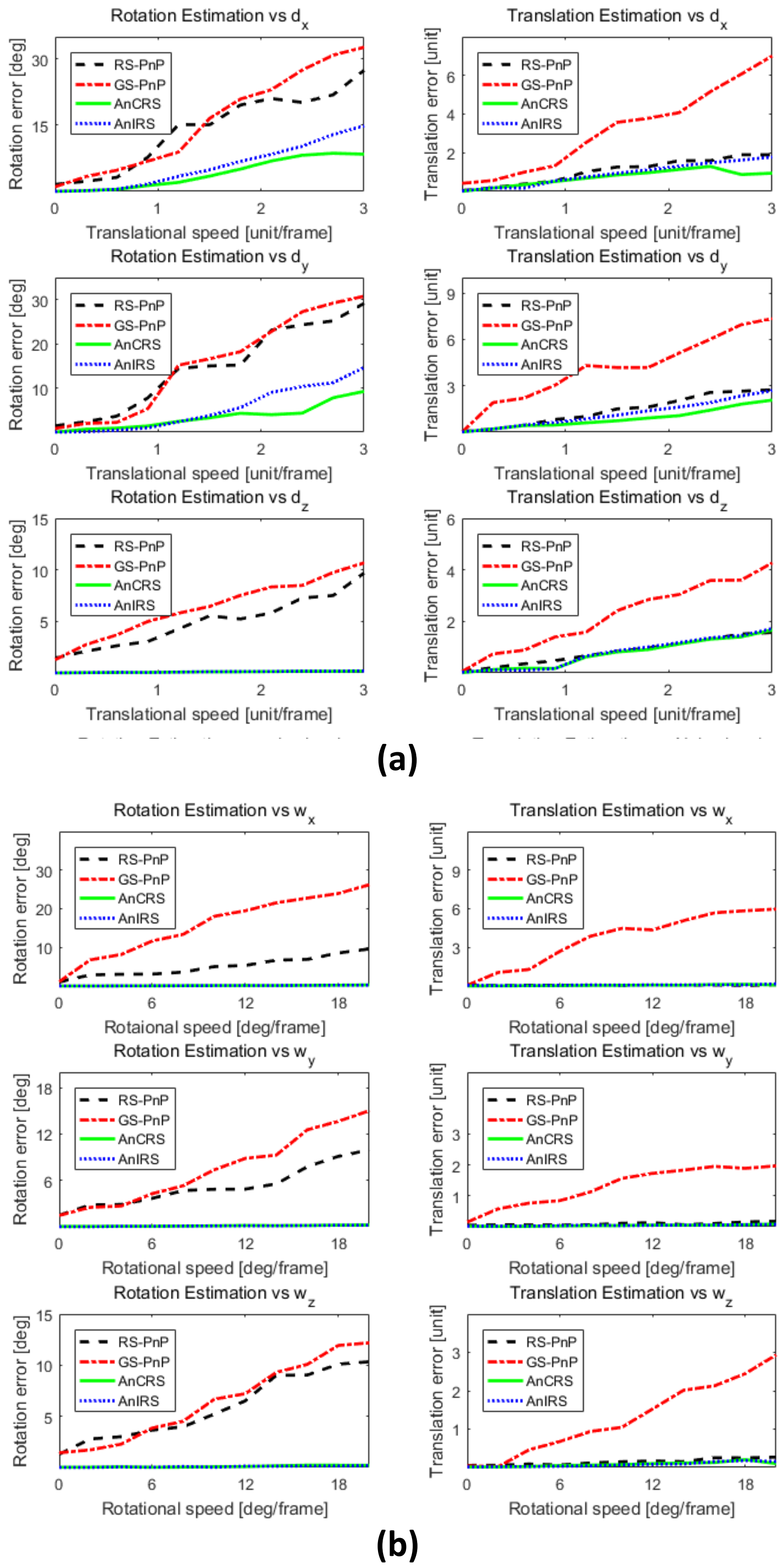
Accuracy vs image noise. In this experiment, we evaluated the robustness of the four solutions against different noise levels. Thus, we fixed the camera with translational and rotational speed at 1 unit/frame and 5 deg/frame. Random noise with levels varying from 0 to 2 px were added to the 60 image points. The results in Fig. 6.13(c) show that both **AnIRS** and **AnCRS** are robust to the increasing image noise. In contrast, **GS-PnP** and **RS-PnP** are relatively sensitive to image noise.

Accuracy vs number of matches. The number of 3D-2D matches has a great impact on the PnP problem. Therefore, we evaluated the performance of the proposed method with different numbers of 3D-2D matches. The camera was fixed with translational and rotational speed at 1 unit/frame and 5 deg/frame. The image noise level was set to 1 px. Then we increased the number of matches from 10 to 120. The results in Fig. 6.13(d) show that the estimation accuracy of all four methods increases with the number of matches. However **AnIRS** and **AnCRS** provide better results in both rotation and translation estimation in comparison to **GS-PnP** and **RS-PnP**.

6.7.2 Real Data

6.7.2.1 Augmented Reality with an RS Video

The four methods have been further evaluated by using real RS images. A planar marker providing 64 3D-2D matches was captured by a hand-held logitech webcam. Strong RS effects are present on the recorded video due to the quick arbitrary camera instantaneous-motion. This scenario can occur in many AR applications. After obtaining the camera pose and instantaneous-motion, the boundaries of the calibration board were reprojected into the RS image. As shown in Fig. 6.15, if the poses and instantaneous-motions are accurately recovered, the reprojected matrix boundaries can perfectly fit the planar marker. In addition to visual checking, the mean value of reprojection errors of 3D marker points of each frame were used as a quantitative measurement.



(a)

(b)

FIGURE 6.12: Pose estimation errors for **AnIRS**, **AnCRS**, **GS-PnP** and **RS-PnP** under different instantaneous-translations (a) and instantaneous-rotations (b).

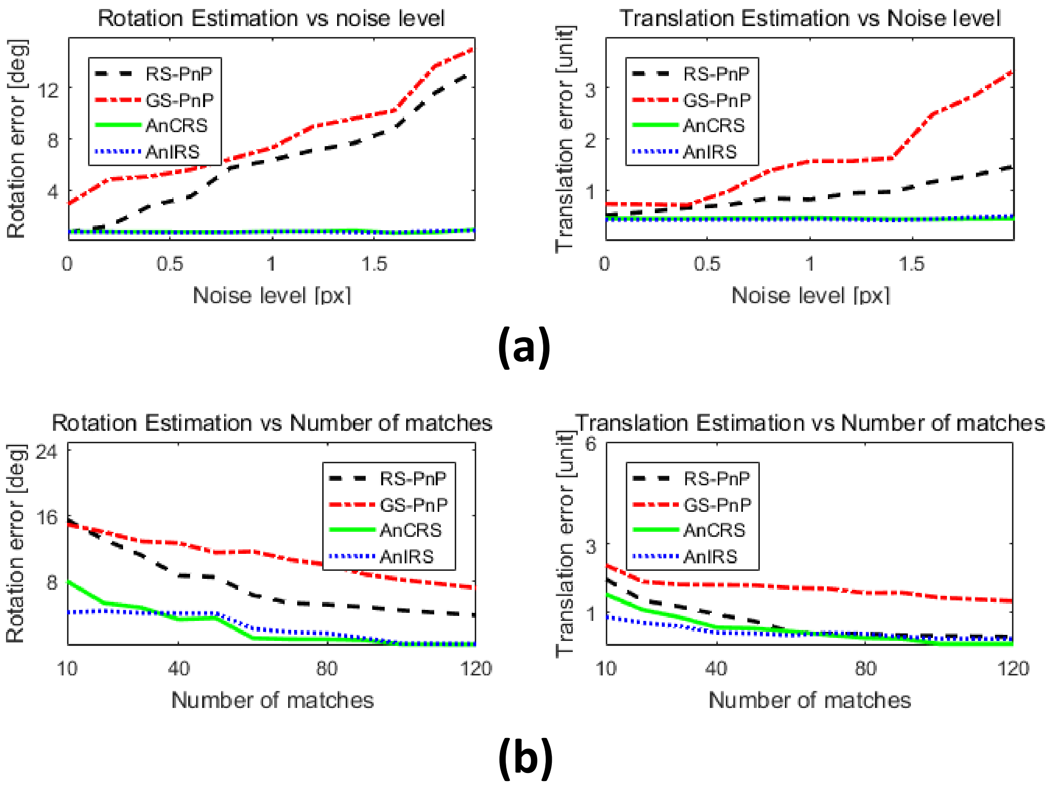


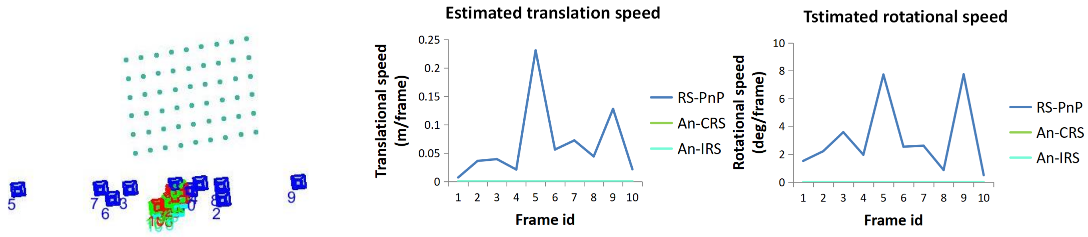
FIGURE 6.13: Pose estimation errors for **AnIRS**, **AnCRS**, **GS-PnP** and **RS-PnP** under different image noise levels (a) and number of matches (b).

In the first 10 frames, all four methods obtained acceptable reprojected matrix boundaries due to the small RS effects. However, finding more inliers does not ensure retrieving the true pose and instantaneous-motion, as **RS-PnP** yields 20 geometrically feasible solutions and it is challenging to pick the 'true' one. For example, Fig. 6.14(a) shows the estimated pose in our AR dataset, where only static camera frames (without instantaneous-motion) were picked. Fig. 6.14(b) shows that R6P gives distributed locations and huge instantaneous-motion up to 5m/frame, while P3P and our method give similar poses.

In the second frame, with the camera quickly moving, **RS-PnP** and The GS-based method **GS-PnP** provide unstable estimates of camera pose. In contrast, both proposed methods **AnIRS** and **AnCRS** significantly outperform **GS-PnP** and **RS-PnP**. It is noteworthy that **AnCRS** achieves slightly smaller reprojection errors than **AnIRS**. This coincides with the observations made in the synthetic experiments and confirms the theoretical analysis of section 6.5.1.1 that the conformal constraint is more suitable to explain the virtual deformations.

6.7.2.2 Pose Registration with Real RS Video

We tested the four methods for pose registration of an SfM reconstruction. The public dataset [Hedborg et al., 2012] was used, which was captured by both RS and GS cameras installed on a rig. The 3D points were obtained by performing SfM with the GS images.



(a) (b)

FIGURE 6.14: Pose and instantaneous-motion estimation by different methods with 10 frames with small RS effects.

3D-2D correspondences can be obtained by matching RS images to GS images. The results are presented in Fig. 6.16. The proposed methods **AnIRS** and **AnCRS** give clearly more accurate estimates than **GS-PnP** and **RS-PnP** for most of the frames.

6.7.3 Running Time

The experiments were conducted on an i5 CPU at 2.8GHz with 4G RAM. On average, it took around 2.8s per frame for **AnIRS** (0.1s for isometric reconstruction and 2.7s for 3D-3D registration) and 14.6s for **AnCRS** (10.6s for conformal reconstruction and 4s for 3D-3D registration). Since the proposed method was implemented in MATLAB, an improvement can be expected when using C++ and GPU acceleration, as shown in [Collins and Bartoli, 2015].

6.8 Experiments of RSSfM

In our experiments, the Iso-NRSfM and InfP-NRSfM [Parashar et al., 2018][4] are both used to reconstruct the virtual deformed shapes. Then we use the stratified GPA method [Bartoli et al., 2013][5] to initialize the optimization described by Eq. (6.7) using the Levenberg-Marquardt algorithm. The proposed method was compared to four state-of-the-art techniques:

- **SfM:** An SfM method close to [Wu, 2011][6].
- **RSBA [Albl et al., 2016b]:** SfM followed by R6P [Albl et al., 2015] to initialize camera pose and instantaneous-motions, and refinement by RSBA.
- **Iso-RSSfM:** The proposed method with Iso-NRSfM.
- **InfP-RSSfM:** The proposed method with InfP-NRSfM.

6.8.1 Synthetic Data

We simulated RS cameras located randomly on a sphere with a radius of 20 units and pointing to a cylindrical surface consisting of 81 points. The size of the surface is 8 units × 8 units with a varying radius. The RS image size is 640p × 480p and the focal length 320p. We compared all methods by varying the instantaneous-motion speed, the noise

[4] http://igt.ip.uca.fr/~ab/Research/Local-Iso-NRSfM_v1p1.zip
[5] http://igt.ip.uca.fr/~ab/Research/SGPA_v1p0.tar.gz
[6] http://mathworks.com/help/vision/examples/structure-from-motion-from-multiple-views.html

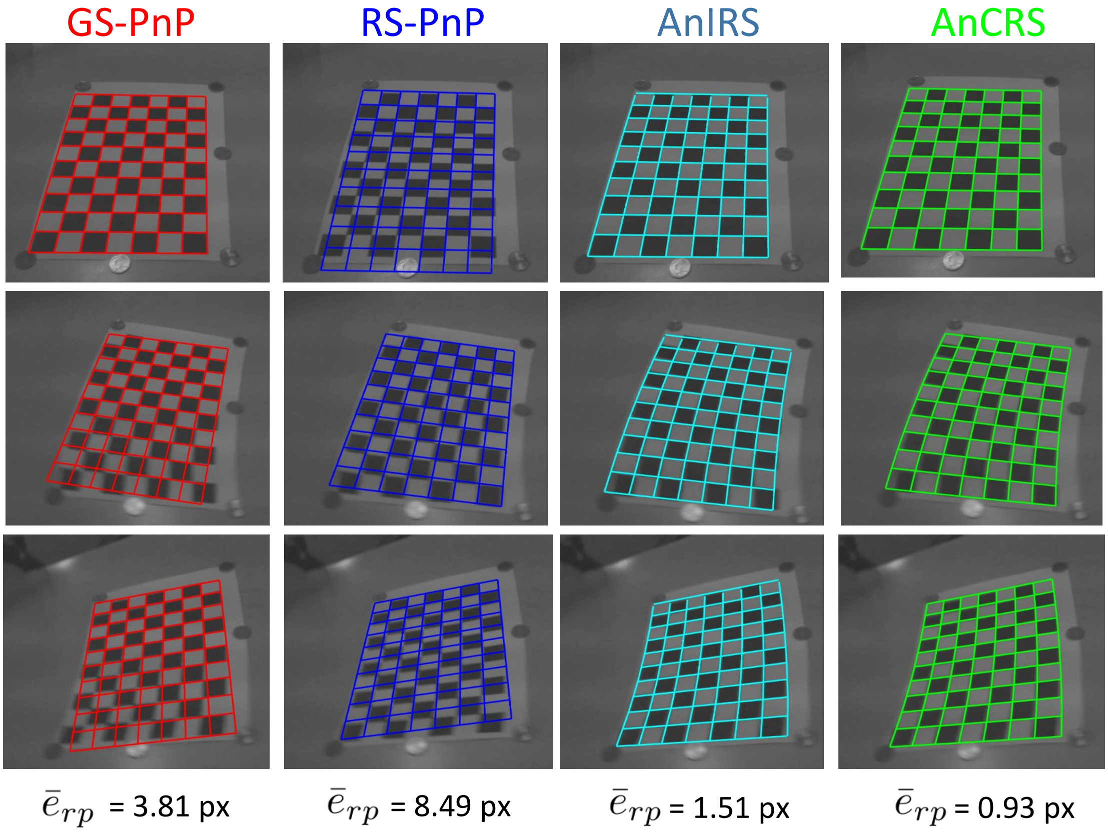


FIGURE 6.15: Visual comparison of reprojected object boundaries by different camera pose and instantaneous-motion estimates. e_{rp} is the reprojection error of the 3D marker points.

on image measurements, the number of views, the surface curvature and the readout direction. The results are obtained after averaging the errors over 50 trials. The default setting is 15 degs/frame and 0.5 units/frame for rotational and translational speed, 1p noise, 6 views, 15 units radius (inverse curvature).

	d_x	d_y	d_z	ω_x	ω_y	ω_z
$\lvert e_{\text{InfP}}\rvert$	0.067	0.065	0.063	0.115	0.120	0.122
$\lvert e_{\text{Iso}}\rvert$	0.067	0.065	**0.062**	**0.110**	0.120	**0.121**

TABLE 6.2: Mean values ($\lvert e_{\text{InfP}}\rvert$, $\lvert e_{\text{Iso}}\rvert$) of reconstruction errors (expressed in units) of the virtual deformed shape by **InfP-RSSfM** and **Iso-RSSfM** under six types of camera instantaneous-motion.

(1) Reconstructing the virtual deformed shapes. We first evaluate the ability of **InfP-RSSfM** and **Iso-RSSfM** to reconstruct the virtual deformed shapes. Since the types of deformation depend on the type of RS instantaneous-motion, we measure the mean distance between the reconstructed 3D points and the corresponding ground truth 3D points computed by Eq. (6.4) and (6.8). The results in Fig. 6.17 and table 6.2 show that the two proposed methods accurately reconstruct the non-rigid shapes under different

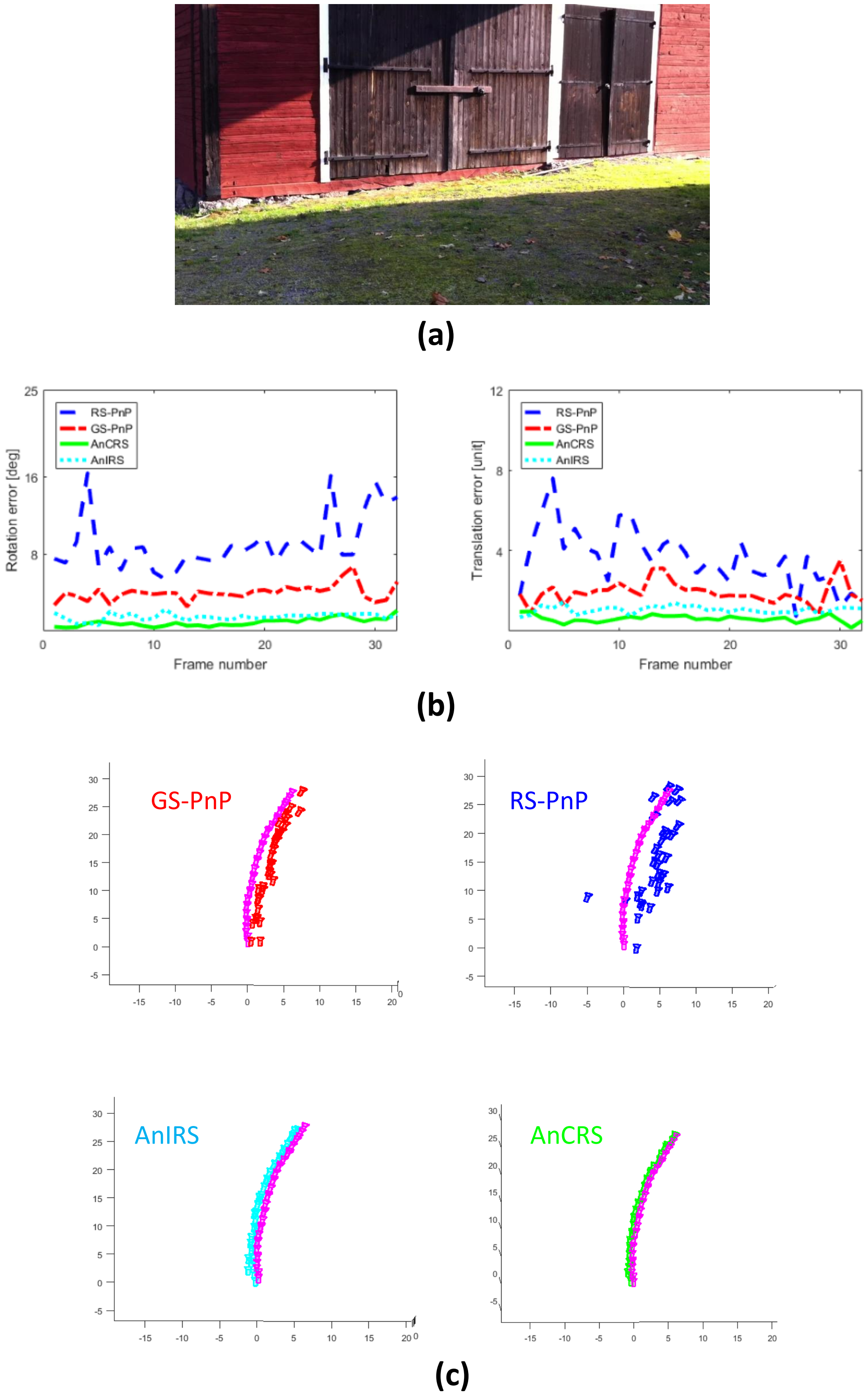


FIGURE 6.16: Results of pose registration with real RS video: **(a)** An example of input RS image. **(b)** Rotation and translation errors of each frame. **(c)** Estimated trajectories by **GS-PnP**, **RS-PnP**, **AnIRS** and **AnCRS** compared to ground truth.

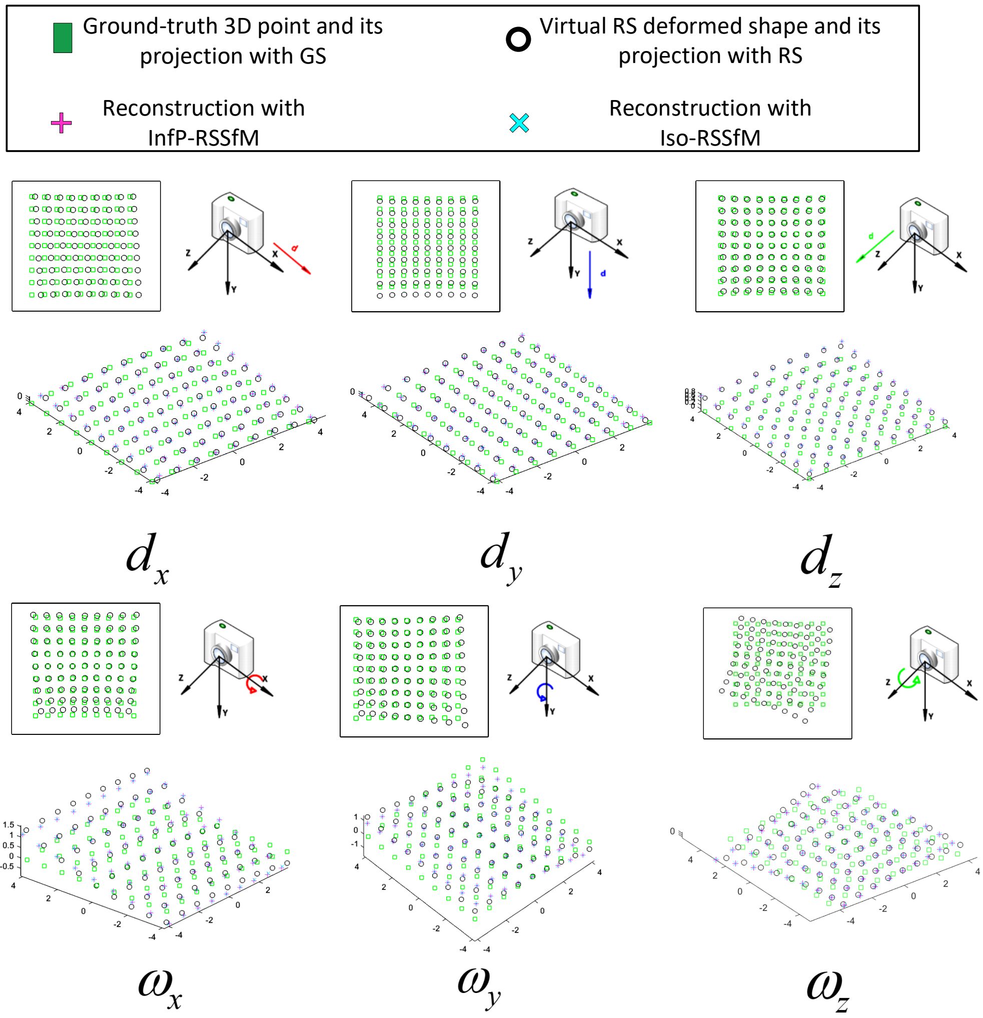


FIGURE 6.17: Deformed shapes reconstructed by **InfP-RSSfM** and **Iso-RSSfM** in comparison to ground truth under six types of camera instantaneous-motion.

instantaneous-motion types. Although **Iso-RSSfM** achieves slightly better reconstruction for $\mathbf{d}_z$, $\boldsymbol{\omega}_x$ and $\boldsymbol{\omega}_z$ than **InfP-RSSfM**, no significant visual differences can be observed.

(2) Varying instantaneous-motion speed. We evaluated the robustness of the four methods against increasing rotational and translational speed from 0 to 30 degs/frame and 1 units/frame gradually, but with random directions. We measure the reconstruction errors (mean difference between computed and ground truth 3D points in units) and pose errors (mean difference between the computed and ground truth rotation $e_{\text{rot}} = \arccos((\text{tr}(\mathbf{R}\mathbf{R}_{\text{GT}}^\top) - 1)/2)$ and translation $e_{\text{trans}} = \arccos(\mathbf{t}^\top \mathbf{t}_{\text{GT}}/(\|\mathbf{t}\| \, \|\mathbf{t}_{\text{GT}}\|))$ of each camera in deg). The results in Fig. 6.18 show that the estimated errors of **SfM** grow with speed. **RSBA** achieves better results with slow instantaneous-motion, while its errors grow dramatically beyond 15 degs/frame. In contrast, both **InfP-RSSfM** and **Iso-RSSfM** provide the best results under all configurations.

(3) Varying noise. In Fig. 6.19, we observe that the errors for all methods increase linearly when noise varies from 0 to 3 pixels. However, **SfM** shows a better tolerance to

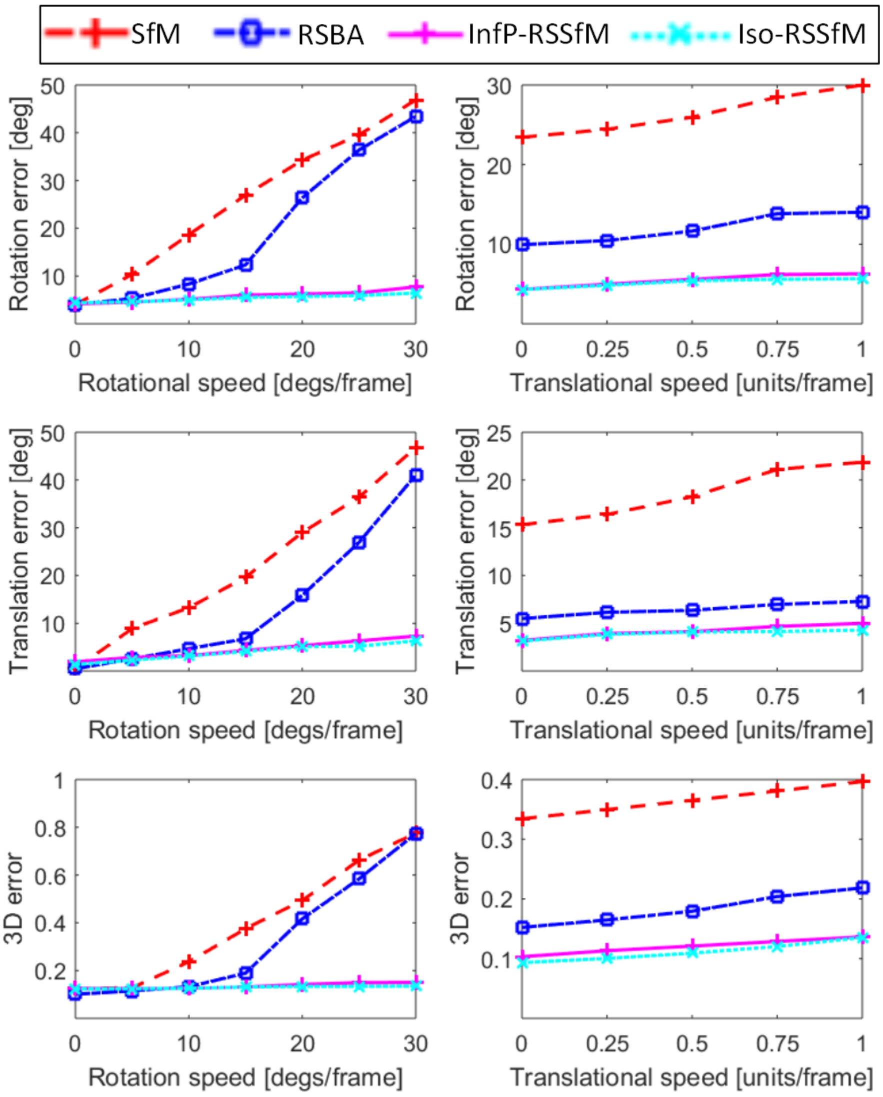


FIGURE 6.18: Reconstruction and pose estimation errors for **SfM**, **RSBA**, **InfP-RSSfM** and **Iso-RSSfM** with increasing rotational and translational speed.

noise than **RSBA** even though its global performance is lower. Both proposed methods achieve the best performance with all noise levels.

(4) Varying number of views. Fig. 6.19 shows that all the four methods give descending errors from 3 to 12 views. **InfP-RSSfM** and **Iso-RSSfM** provide similar results, and better than **SfM** and **RSBA**.

(5) Varying curvature. In this experiment, we vary the radius of the surface (inverse of the curvature) from 5 to 30 units. The results in Fig. 6.19 show that all the four methods perform better with smaller curvature. The performance of **InfP-RSSfM** and **Iso-RSSfM** are the best among the compared methods. However, as expected **Iso-RSSfM** provides slightly better results than **InfP-RSSfM** when the curvature is large.

(6) Varying readout direction angle. We evaluate the robustness of the four methods with an RS critical motion sequence. We vary the readout directions of the cameras from

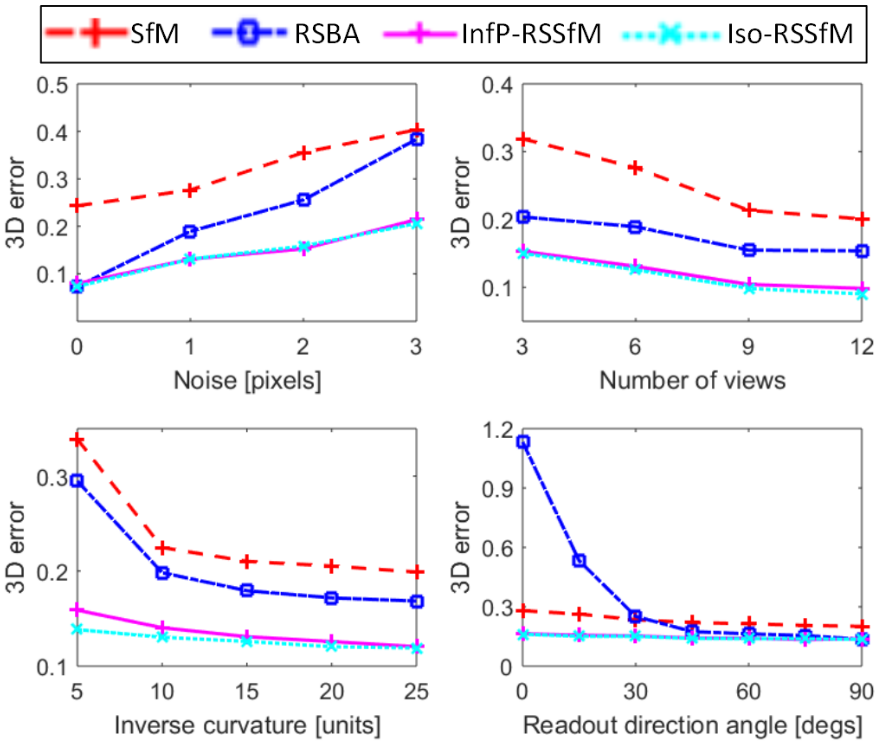


FIGURE 6.19: Reconstruction errors for **SfM**, **RSBA**, **InfP-RSSfM** and **Iso-RSSfM** under different noise levels in image, number of views, curvature and readout direction angle.

parallel to perpendicular by increasing the mean angles between them from 0 deg to 90 degs (degenerate to stable). In Fig. 6.19, we observe that **RSBA** provides better results than **SfM** with at least 30 deg readout direction angle. While with smaller angle, the reconstruction error of **RSBA** grows dramatically, which means that it collapses into the planar degenerate solution. As expected from the analysis in section 6.6.2.3, **InfP-RSSfM** and **Iso-RSSfM** provides stable results under all settings.

(7) Data from public benchmark. We tested the four methods on synthetic RS image datasets from [Forssén and Ringaby, 2010]. We generated unordered image sets by randomly selecting 2 image triplets. In Fig. 6.20, we observe that quantitatively our methods work best in pose estimation and that qualitatively **SfM** obtains deformed reconstruction, while **RSBA** performs even worse and provides extremely deformed reconstruction. In contrast, **InfP-RSSfM** and **Iso-RSSfM** provide reconstructions close to ground truth.]

6.8.2 Real Data

6.8.2.1 Planar Marker Dataset

We use the RS video dataset from [Lao et al., 2018b] which captures a chessboard with strong RS effects. First, the frames from the video sequence were manually categorized into vertical and horizontal readout direction. Then we designed two kinds of experiments: *1)* We randomly chose 3 images from the 'vertical' group and 'horizontal' group respectively. *2)* We randomly chose 6 images from the 'vertical' group only. Since the rigid 3D shape is known, we measured the mean distance difference between the computed and ground truth 3D points. The results in Fig. 6.21 show that **SfM** provides deformed reconstructions in both experiments. **RSBA** obtains better results than **SfM** in the

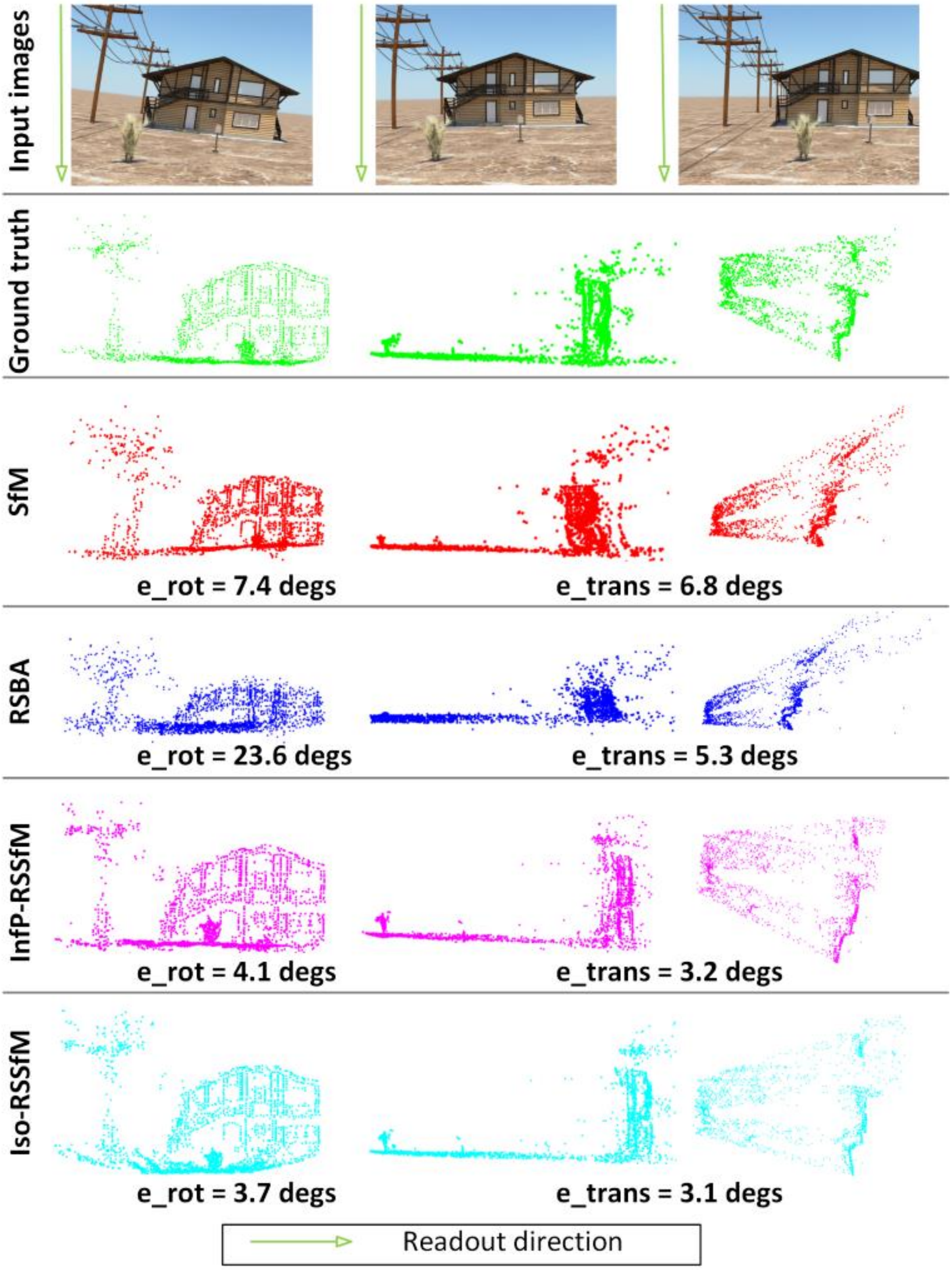


FIGURE 6.20: Reconstruction results and pose estimation errors of **SfM**, **RSBA**, **InfP-RSSfM** and **Iso-RSSfM** for synthetic RS images dataset.

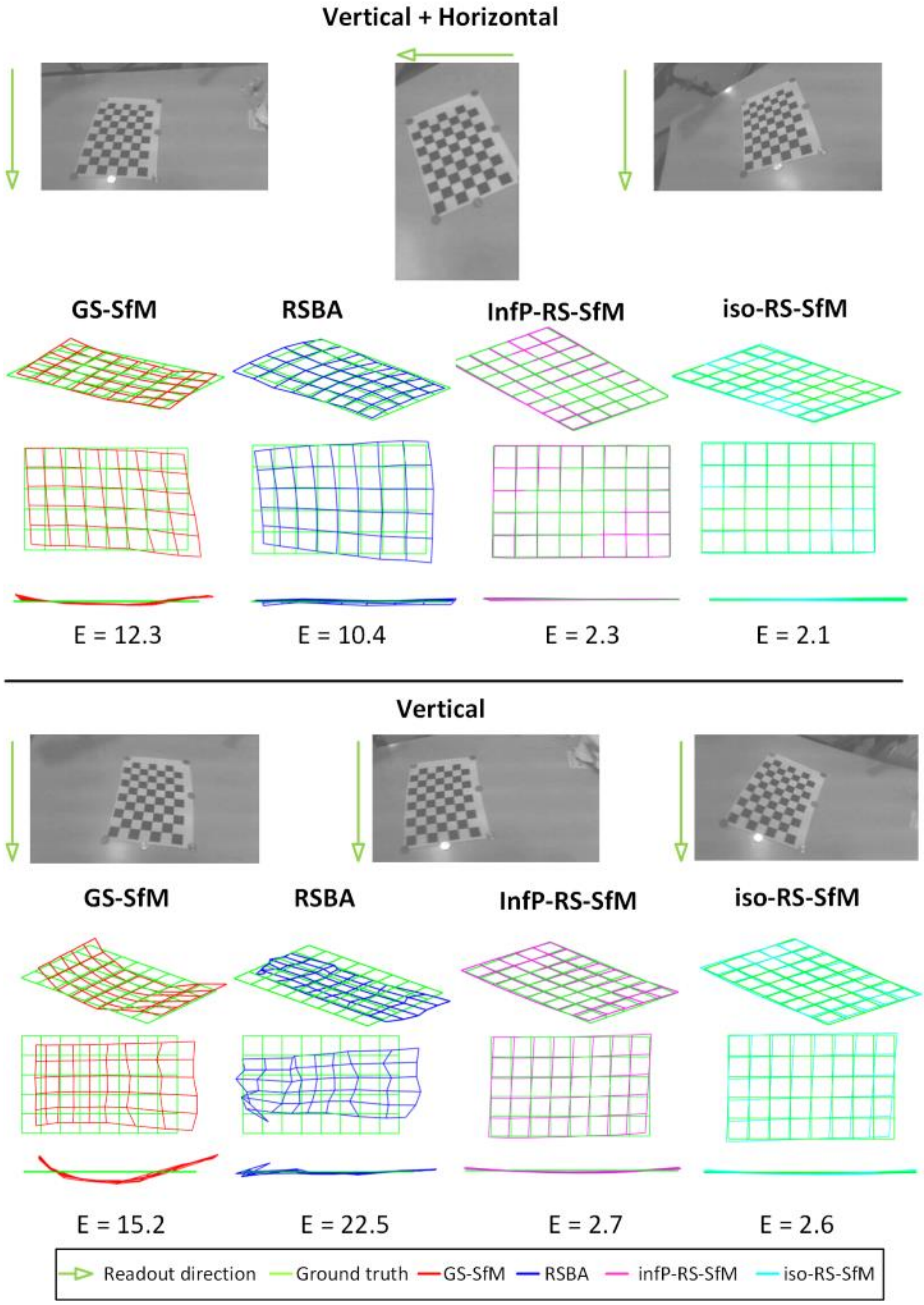


FIGURE 6.21: Reconstructed shapes and mean of reconstruction errors E (expressed in units) of **SfM**, **RSBA**, **InfP-RSSfM** and **Iso-RSSfM** with 'vertical+horizontal' and 'vertical' as inputs respectively for the planar marker dataset.

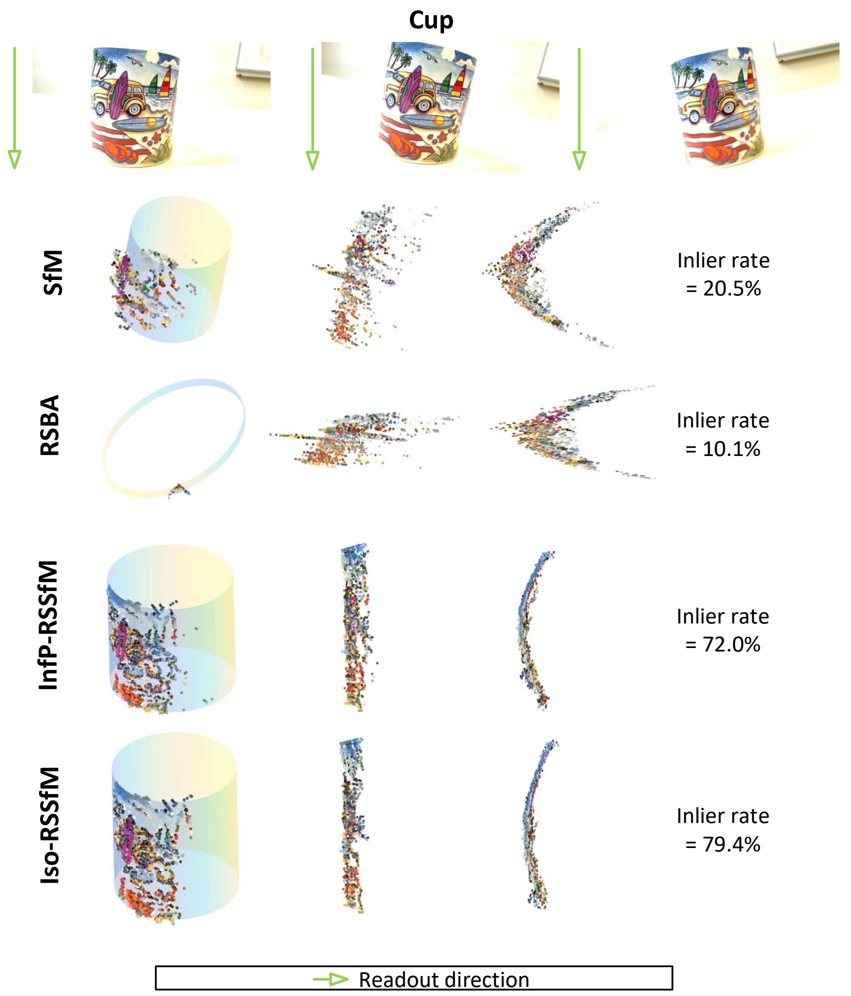


FIGURE 6.22: Visual checking and quantitative evaluations of **SfM**, **RSBA**, **InfP-RSSfM** and **Iso-RSSfM** for the cup dataset.

'vertical+horizontal' experiment, while it suffers from the planar degeneracy and gives a strongly deformed shape in the 'vertical-only' experiment. In contrast, **InfP-RSSfM** and **Iso-RSSfM** provide a correct reconstruction in both experiments.

6.8.2.2 Cup and Box Datasets

A cylinder cup and a cubic box were captured by a hand-held Logitech webcam with strong RS effects. The videos were with close readout direction during the acquisition. Again, we randomly chose 6 frames from each video sequence. The ground-truth is now not available. Thus, we use two methods to evaluate the reconstruction results: *1)* Visual checking. *2)* For the cup dataset, we fitted the computed shapes with cylinders by using the *'pcfitcylinder'* function in MATLAB and measured the fitting errors. For the box dataset, we segmented and fitted the computed scenes with three planes respectively in CloudCompare[7]. Thus, the mean value of fitting errors and angle between normal vector of the three planes (supposed to be 90 degs) are used as quantitative evaluation criteria. We can observe in Fig. 6.22 and Fig. 6.23 that **SfM** fails in handling the RS effects and

[7] https://www.danielgm.net/cc/

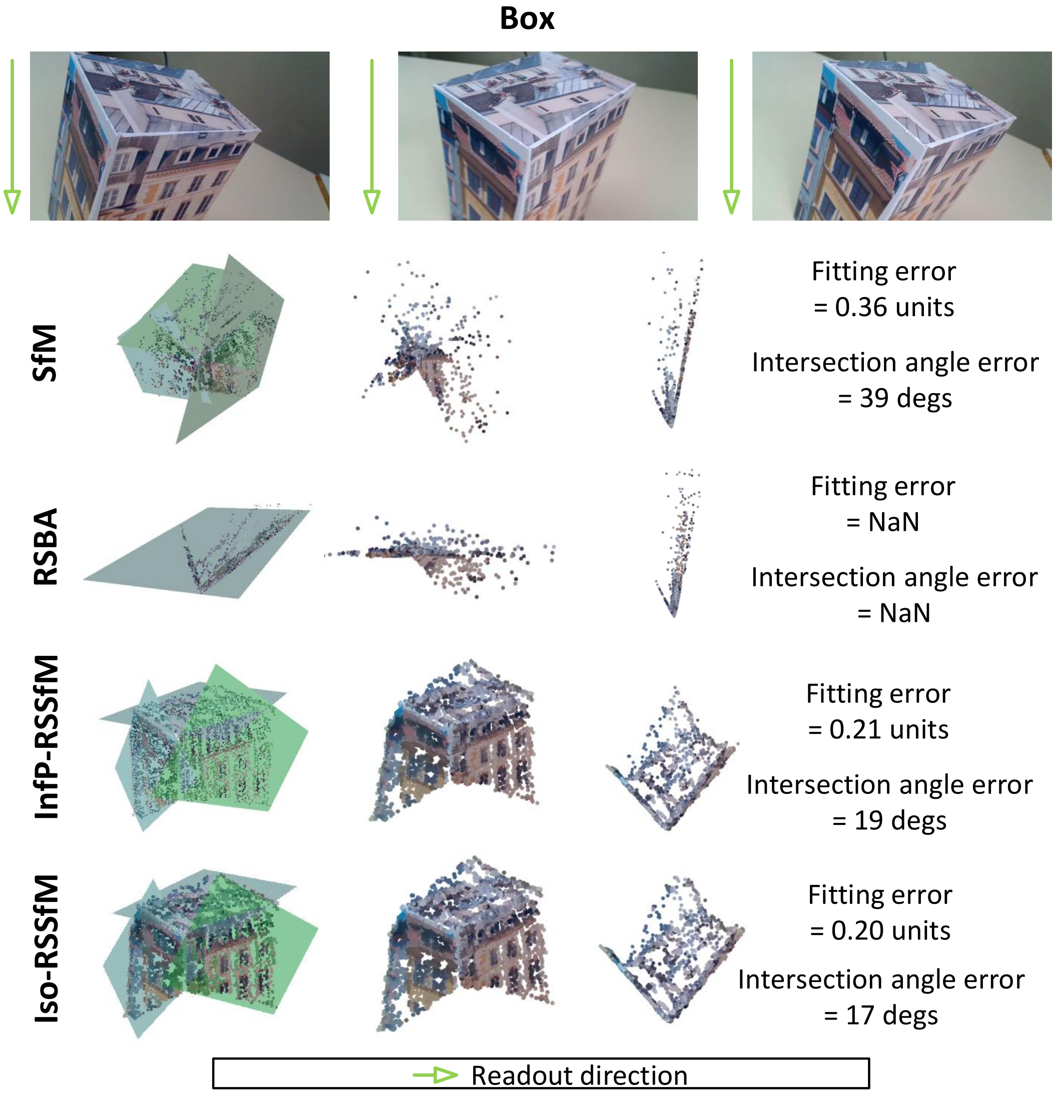


FIGURE 6.23: Visual checking and quantitative evaluations of **SfM**, **RSBA**, **InfP-RSSfM** and **Iso-RSSfM** for the box dataset.

provides deformed reconstructions for the two datasets. Since the readout directions are close to parallel, **RSBA** obtains extremely deformed results close to planar. **InfP-RSSfM** and **Iso-RSSfM** perform best in both the visual checking and quantitative evaluations for both datasets.

6.8.3 Running Time

The proposed methods were implemented in MATLAB. The experiments were conducted on an i5 CPU at 2.8GHz with 4G RAM. Table 6.3 summarises the results and shows that the running time of both **InfP-RSSfM** and **Iso-RSSfM** grows slightly with the increasing number of point correspondences and views.

Number of points	40	60	80
SfM	4	4	5
RSBA	12	17	24
InfP-RSSfM	45	46	49
Iso-RSSfM	54	61	67
Number of views	6	9	12
SfM	7	12	20
RSBA	54	100	153
InfP-RSSfM	74	93	116
Iso-RSSfM	90	109	132

TABLE 6.3: Comparison of computation time (in seconds) of **SfM**, **RSBA**, **InfP-RSSfM** and **Iso-RSSfM** for 6, 9 and 12 views and 40, 60 and 80 point correspondences with default 3 views and 80 points.

6.9 Discussion and Conclusion

In this chapter, we have proposed two novel methods which addresses the RS-PInP and RSSfM problems respectively from a new angle: using non-rigid vision.

RS-PInP We propose a novel method for RS-PInP problem using SfT. By analyzing the link between the SfT and RS-PInP problems we have shown that RS effects can be explained by the GS projection of a virtually deformed shape. As a result the RS-PInP problem is transformed into a 3D-3D registration problem. Experimental results have shown that the proposed methods outperform existing RS-PInP techniques in terms of accuracy and stability. We interpret this improved accuracy as the result of transforming the problem from a 3D-2D registration into a 3D-3D registration problem. This has enabled us to use 3D point-distances instead of the re-projection errors, which carry more physical meaning and make the error terms homogeneous. A possible extension of our work is to derive the exact differential properties of virtual deformation.

RSSfM We proposed a novel solution to RSSfM using NRSfM. By showing that the RS effects in multiple images can be explained by multiple one-to-one virtual deformations of a rigid 3D shape captured by GS cameras, we drew a link between RSSfM and NRSfM. As a result, RSSfM is transformed into a 3D-3D registration problem, which we show theoretically and experimentally can successfully avoid the risk of collapsing into a degenerate solution with the usual camera capture manner (parallel readout directions). We showed that the proposed methods outperform the existing RSSfM methods using 3D-2D registration in accuracy and stability.

Chapter 7

Conclusion and Future Work

The aim of this thesis is to address all the problems of 3D vision with RS images. These problems form the so called SfM pipeline. As shown in Fig. 7.1, we have proposed several algorithms that can be used for each step of the pipeline when images may show RS effects.

In chapter 3, we presented a single-image RS correction method which uses line features. Unlike existing methods, our method R4C is based on a linear computation of instantaneous-motion using few image features without any prior on scene geometry. Besides, the method was integrated in a RANSAC-like framework which enables us to reject outlier curves making image correction more robust and fully automated. Specifically, our method not only produces visually pleasant corrections, but also preserves consistency of geometry. Thus, the proposed method has been integrated in a 3-steps RSSfM which uses corrected images.

In chapter 4, we proposed a novel C-RSBA to be as final refinement method. Contrarily to classical BA, C-RSBA successfully avoid planar degeneracy which tends to flatten the reconstructed 3D scene when scanning directions of input image sequence are similar or close. With C-RSBA, we relax the constraint on capture style suggested by other authors to handle this degeneracy, and thus extend the use of RS cameras in realistic image capture conditions.

In chapter 5 we proposed a new matrix for the RS case that is equivalent to homography matrix. We first defined the theoretical RS Homography matrix and proposed a solver to retrieve it from an image pair. Then we derived a simplified homography matrix and the associated minimal solver which is more suited for RANSAC based applications. This RS homography was used as the basis for extending two well-known homography-based methods, i.e. relative pose estimation and image stitching, to the RS case.

Within chapter 6, we proposed for the first time a completely different approach to achieve 3D vision with RS images. This consists in establishing an analogy between deformable-surface vision and RS effects. We presented a novel method which addresses the RS-PEnP problem: using SfT. By analyzing the link between the SfT and RS-PEnP problems we have shown that RS effects can be explained by the GS projection of a virtually deformed shape. As a result the RS-PEnP is expressed as a 3D-3D registration problem. The proposed solution outperforms existing RS-PEnP techniques in terms of accuracy and stability. We also established an analogy between SfM with multiple RS images and Non-Rigid SfM. We proposed a novel solution for RSSfM by showing that the RS effects in multiple images can be explained by multiple one-to-one virtual deformations of a 3D shape captured by GS cameras. Again, RSSfM is transformed into a 3D-3D registration problem that solved using more physically meaningful error functions.

All the proposed methods and algorithms were intensively tested using both synthetic and real data from famous benchmarks. We systematically compared our results to those obtained with the most related state-of-the-art methods. The experiment results

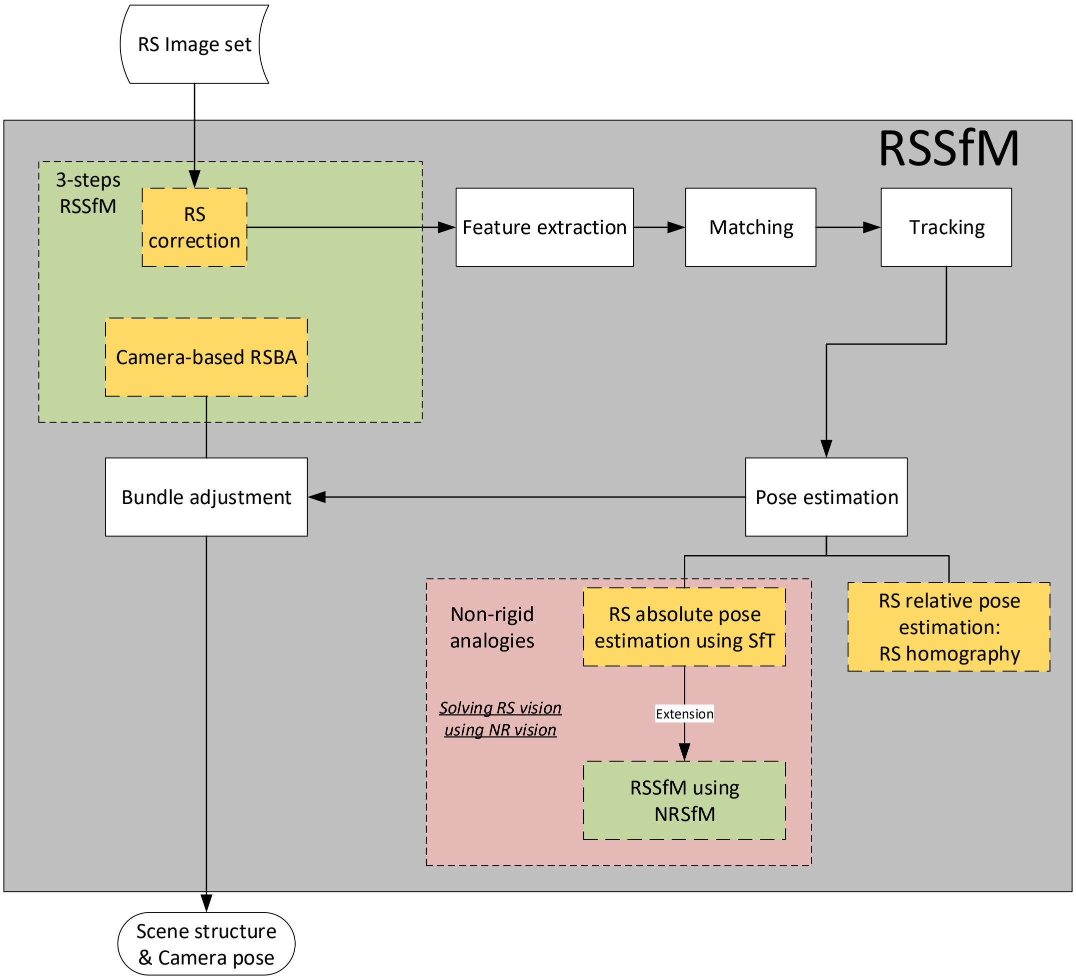


FIGURE 7.1: Revist of the contributions and organization of this thesis.

show that the proposed methods are superior to the state-of-the-art techniques and outperform well known commercial image editing applications.

In summary, the methods described in this manuscript:

- Handles common singular and degenerate configurations of RS image sequences.
- Do not make unattainable assumptions on scene geometry or camera kinematics.
- Outperform related state-of-the-art methods.
- Handle very strong RS effects.

We believe that this work will help to take an extra step toward the widespread of use of RS cameras in real-life computer vision applications.

Future Work. The very next steps of this work will be dedicated to making the proposed methods efficient and fully automated in realistic conditions.

First, an efficient implementation of the algorithms will be achieved by migrating the Matlab sources toward real-time compiled languages. Analytical expression of the Jacobian matrix would also help to make faster the steps based on non-linear optimization.

Beside, a careful analysis of the structure of the jacobian and the normal equation matrix may shows a sparse and specific structure that can be exploited to make the resolution more efficient.

Local effect of RS should also be studied. Although this effect is generally negligible, it is of theoretical interest to study how RS impacts feature detectors and descriptors. The goal is to boost the matching performances and to make the algorithms handle even higher speed motion.

Finally, the spatio-temporal projection models used for RS may be generalized and unified with other vision systems that are based on spatio-temporal acquisition principle. Obvious candidates are event-based cameras and dynamic-reconfigurable-ROI systems [Dahmouche et al., 2012].

Bibliography

[pho,] Adobe photoshop cc.

[ICE,] Image composite editor - microsoft research.

[Sau, 2013] (2013). Rolling Shutter Stereo. In *ICCV*.

[Agarwal et al., 2009] Agarwal, S., Snavely, N., Simon, I., Seitz, S. M., and Szeliski, R. (2009). Building rome in a day. In *ICCV*.

[Agudo and Moreno-Noguer, 2015] Agudo, A. and Moreno-Noguer, F. (2015). Simultaneous pose and non-rigid shape with particle dynamics. In *CVPR*.

[Agudo et al., 2016] Agudo, A., Moreno-Noguer, F., Calvo, B., and Montiel, J. M. M. (2016). Sequential non-rigid structure from motion using physical priors. *PAMI*.

[Ait-Aider et al., 2006] Ait-Aider, O., Andreff, N., Lavest, J. M., and Martinet, P. (2006). Simultaneous object pose and velocity computation using a single view from a rolling shutter camera. In *ECCV*.

[Ait-Aider et al., 2007] Ait-Aider, O., Bartoli, A., and Andreff, N. (2007). Kinematics from lines in a single rolling shutter image. In *CVPR*.

[Ait-Aider and Berry, 2009] Ait-Aider, O. and Berry, F. (2009). Structure and kinematics triangulation with a rolling shutter stereo rig. In *ICCV*.

[Akhter et al., 2009] Akhter, I., Sheikh, Y., Khan, S., and Kanade, T. (2009). Nonrigid structure from motion in trajectory space. In *NIPS*.

[Albl et al., 2019] Albl, C., Kukelova, Z., Larsson, V., and Pajdla, T. (2019). Rolling shutter camera absolute pose. *PAMI*.

[Albl et al., 2015] Albl, C., Kukelova, Z., and Pajdla, T. (2015). R6p-rolling shutter absolute camera pose. In *CVPR*.

[Albl et al., 2016a] Albl, C., Kukelova, Z., and Pajdla, T. (2016a). Rolling shutter absolute pose problem with known vertical direction. In *CVPR*.

[Albl et al., 2016b] Albl, C., Sugimoto, A., and Pajdla, T. (2016b). Degeneracies in rolling shutter sfm. In *ECCV*.

[Bapat and Frahm, 2016] Bapat, Akash, E. D. and Frahm, J.-M. (2016). Towards kilo-hertz 6-dof visual tracking using an egocentric cluster of rolling shutter cameras. *TVCG*.

[Bapat and Frahm, 2018] Bapat, Akash, T. P. and Frahm, J.-M. (2018). Rolling shutter and radial distortion are features for high frame rate multi-camera tracking. In *CVPR*.

[Bartoli et al., 2015] Bartoli, A., Gérard, Y., Chadebecq, F., Collins, T., and Pizarro, D. (2015). Shape-from-template. *PAMI*.

[Bartoli et al., 2013] Bartoli, A., Pizarro, D., and Loog, M. (2013). Stratified generalized procrustes analysis. *IJCV*.

[Bay et al., 2006] Bay, H., Tuytelaars, T., and Van Gool, L. (2006). Surf: Speeded up robust features. In *ECCV*.

[Bregler et al., 2000] Bregler, C., Hertzmann, A., and Biermann, H. (2000). Recovering non-rigid 3d shape from image streams. In *CVPR*.

[Brown and Lowe, 2007] Brown, M. and Lowe, D. G. (2007). Automatic panoramic image stitching using invariant features. *IJCV*.

[Brown et al., 2003] Brown, M., Lowe, D. G., et al. (2003). Recognising panoramas. In *ICCV*, volume 3, page 1218.

[Brunet et al., 2014] Brunet, F., Bartoli, A., and Hartley, R. I. (2014). Monocular template-based 3d surface reconstruction: Convex inextensible and nonconvex isometric methods. *CVIU*.

[Burt and Adelson, 1983] Burt, P. J. and Adelson, E. H. (1983). A multiresolution spline with application to image mosaics. *ACM Transactions on Graphics*, 2(4):217–236.

[Chhatkuli et al., 2017] Chhatkuli, A., Pizarro, D., Bartoli, A., and Collins, T. (2017). A stable analytical framework for isometric shape-from-template by surface integration. *PAMI*.

[Chhatkuli et al., 2018] Chhatkuli, A., Pizarro, D., Collins, T., and Bartoli, A. (2018). Inextensible non-rigid structure-from-motion by second-order cone programming. *PAMI*.

[Collins and Bartoli, 2015] Collins, T. and Bartoli, A. (2015). [poster] realtime shape-from-template: System and applications. In *ISMAR*.

[Dahmouche et al., 2012] Dahmouche, R., Andreff, N., Mezouar, Y., Ait-Aider, O., and Martinet, P. (2012). Dynamic visual servoing from sequential regions of interest acquisition. *IJRR*.

[Dai et al., 2016] Dai, Y., Li, H., and Kneip, L. (2016). Rolling shutter camera relative pose: generalized epipolar geometry. In *CVPR*.

[Devernay and Faugeras, 2001] Devernay, F. and Faugeras, O. (2001). Straight lines have to be straight. *Machine vision and applications*, 13(1):14–24.

[Dryden et al., 1998] Dryden, I. L., Mardia, K. V., et al. (1998). Statistical shape analysis.

[Duchamp et al., 2015] Duchamp, G., Ait-Aider, O., Royer, E., and Lavest, J.-M. (2015). A rolling shutter compliant method for localisation and reconstruction. In *VISAPP*.

[Feldman et al., 2003] Feldman, D., Weinshall, D., et al. (2003). On the epipolar geometry of the crossed-slits projection. In *ICCV*.

[Fischler and Bolles, 1981] Fischler, M. A. and Bolles, R. C. (1981). Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*.

[Forssén and Ringaby, 2010] Forssén, P.-E. and Ringaby, E. (2010). Rectifying rolling shutter video from hand-held devices. In *CVPR*.

[Gao et al., 2003] Gao, X.-S., Hou, X.-R., Tang, J., and Cheng, H.-F. (2003). Complete solution classification for the perspective-three-point problem. *PAMI*.

[Golub and Van Loan, 2012] Golub, G. H. and Van Loan, C. F. (2012). *Matrix computations*. JHU Press.

[Gotardo and Martinez, 2011a] Gotardo, P. F. and Martinez, A. M. (2011a). Computing smooth time trajectories for camera and deformable shape in structure from motion with occlusion. *PAMI*.

[Gotardo and Martinez, 2011b] Gotardo, P. F. and Martinez, A. M. (2011b). Kernel non-rigid structure from motion. In *ICCV*.

[Gotardo and Martinez, 2011c] Gotardo, P. F. and Martinez, A. M. (2011c). Non-rigid structure from motion with complementary rank-3 spaces. In *CVPR*.

[Grundmann et al., 2012] Grundmann, M., Kwatra, V., Castro, D., and Essa, I. (2012). Calibration-free rolling shutter removal. In *ICCP*.

[Guan et al., 2018] Guan, B., Vasseur, P., Demonceaux, C., and Fraundorfer, F. (2018). Visual odometry using a homography formulation with decoupled rotation and translation estimation using minimal solutions. In *ICRA*.

[Haouchine et al., 2014] Haouchine, N., Dequidt, J., Berger, M.-O., and Cotin, S. (2014). Single view augmentation of 3d elastic objects. In *ISMAR*.

[Haralick et al., 1991] Haralick, R. M., Lee, D., Ottenburg, K., and Nolle, M. (1991). Analysis and solutions of the three point perspective pose estimation problem. In *CVPR*.

[Hartley and Kahl, 2007] Hartley, R. and Kahl, F. (2007). Critical configurations for projective reconstruction from multiple views. *IJCV*.

[Hartley and Vidal, 2008] Hartley, R. and Vidal, R. (2008). Perspective nonrigid shape and motion recovery. In *ECCV*.

[Hartley and Zisserman, 2003] Hartley, R. and Zisserman, A. (2003). *Multiple view geometry in computer vision*. Cambridge university press.

[Hartley, 1995] Hartley, R. I. (1995). In defence of the 8-point algorithm. In *ICCV*.

[Hedborg et al., 2012] Hedborg, J., Forssen, P.-E., Felsberg, M., and Ringaby, E. (2012). Rolling shutter bundle adjustment. In *CVPR*.

[Hedborg et al., 2011] Hedborg, J., Ringaby, E., Forssén, P.-E., and Felsberg, M. (2011). Structure and motion estimation from rolling shutter video. In *ICCV Workshops*.

[Hee Park and Levoy, 2014] Hee Park, S. and Levoy, M. (2014). Gyro-based multi-image deconvolution for removing handshake blur. In *CVPR*.

[Heinly et al., 2015] Heinly, J., Schonberger, J. L., Dunn, E., and Frahm, J.-M. (2015). Reconstructing the world* in six days*(as captured by the yahoo 100 million image dataset). In *CVPR*.

[Henrion and Lasserre, 2003] Henrion, D. and Lasserre, J.-B. (2003). Gloptipoly: Global optimization over polynomials with matlab and sedumi. *ACM Transactions on Mathematical Software*.

[Horn et al., 1988] Horn, B. K., Hilden, H. M., and Negahdaripour, S. (1988). Closed-form solution of absolute orientation using orthonormal matrices. *JOSA A*.

[Hu et al., 2013] Hu, Y., Zhang, D., Ye, J., Li, X., and He, X. (2013). Fast and accurate matrix completion via truncated nuclear norm regularization. *PAMI*.

[Im et al., 2018] Im, S., Ha, H., Choe, G., Jeon, H.-G., Joo, K., and Kweon, I. S. (2018). Accurate 3d reconstruction from small motion clip for rolling shutter cameras. *PAMI*.

[Ito and Okatani,] Ito, E. and Okatani, T. Self-calibration-based approach to critical motion sequences of rolling-shutter structure from motion. In *CVPR*.

[Jia and Evans, 2012] Jia, C. and Evans, B. L. (2012). Probabilistic 3-d motion estimation for rolling shutter video rectification from visual and inertial measurements. In *MMSP*, pages 203–208.

[Kim et al., 2016] Kim, J. H., Cadena, C., and Reid, I. (2016). Direct semi-dense slam for rolling shutter cameras. In *ICRA*.

[Kim et al., 2011] Kim, Y.-G., Jayanthi, V. R., and Kweon, I.-S. (2011). System-on-chip solution of video stabilization for cmos image sensors in hand-held devices. *IEEE transactions on circuits and systems for video technology*, 21(10):1401–1414.

[Klingner et al., 2013] Klingner, B., Martin, D., and Roseborough, J. (2013). Street view motion-from-structure-from-motion. In *ICCV*.

[Kukelova et al., 2018] Kukelova, Z., Albl, C., Sugimoto, A., and Pajdla, T. (2018). Linear solution to the minimal absolute pose rolling shutter problem. In *ACCV*.

[Kukelova et al., 2008] Kukelova, Z., Bujnak, M., and Pajdla, T. (2008). Automatic generator of minimal problem solvers. In *ECCV*.

[Lao and Ait-Aider, 2018] Lao, Y. and Ait-Aider, O. (2018). A robust method for strong rolling shutter effects correction using lines with automatic feature selection. In *CVPR*.

[Lao et al., 2018a] Lao, Y., Ait-Aider, O., and Araujo, H. (2018a). Robustified structure from motion with rolling-shutter camera using straightness constraint. *Pattern Recognition Letters*.

[Lao et al., 2018b] Lao, Y., Ait-Aider, O., and Bartoli, A. (2018b). Rolling shutter pose and ego-motion estimation using shape-from-template. In *ECCV*.

[Li and Hartley, 2006] Li, H. and Hartley, R. (2006). Five-point motion estimation made easy. In *ICPR*.

[Liang et al., 2008] Liang, C.-K., Chang, L.-W., and Chen, H. H. (2008). Analysis and compensation of rolling shutter effect. *IEEE Transactions on Image Processing*, 17(8):1323–1330.

[Lin et al., 2015] Lin, C.-C., Pankanti, S. U., Natesan Ramamurthy, K., and Aravkin, A. Y. (2015). Adaptive as-natural-as-possible image stitching. In *CVPR*.

[Litwiller, 2001] Litwiller, D. (2001). Ccd vs. cmos: Facts and fiction. *Photonics Spectra*.

[Liu et al., 2013] Liu, S., Yuan, L., Tan, P., and Sun, J. (2013). Bundled camera paths for video stabilization. *ACM Transactions on Graphics*.

[Lourakis and Argyros, 2009] Lourakis, M. I. and Argyros, A. A. (2009). Sba: A software package for generic sparse bundle adjustment. *ACM Transactions on Mathematical Software*.

[Ma et al., 2012] Ma, Y., Soatto, S., Kosecka, J., and Sastry, S. S. (2012). *An invitation to 3-d vision: from images to geometric models*. Springer.

[Magerand et al., 2012] Magerand, L., Bartoli, A., Ait-Aider, O., and Pizarro, D. (2012). Global optimization of object pose and motion from a single rolling shutter image with automatic 2d-3d matching. In *ECCV*.

[Malis and Vargas, 2007] Malis, E. and Vargas, M. (2007). *Deeper understanding of the homography decomposition for vision-based control*. PhD thesis, INRIA.

[Malti et al., 2015] Malti, A., Bartoli, A., and Hartley, R. (2015). A linear least-squares solution to elastic shape-from-template. In *CVPR*.

[Malti and Herzet, 2017] Malti, A. and Herzet, C. (2017). Elastic shape-from-template with spatially sparse deforming forces. In *CVPR*.

[Maybank, 2012] Maybank, S. (2012). *Theory of reconstruction from image motion*. Springer Science & Business Media.

[Meingast et al., 2005] Meingast, M., Geyer, C., and Sastry, S. (2005). Geometric models of rolling-shutter cameras. In *OMNIVIS*.

[Morawiec, 2003] Morawiec, A. (2003). *Orientations and rotations*. Springer.

[Nistér, 2004] Nistér, D. (2004). An efficient solution to the five-point relative pose problem. *PAMI*.

[Pajdla, 2002] Pajdla, T. (2002). Stereo with oblique cameras. *IJCV*.

[Parashar et al., 2018] Parashar, S., Pizarro, D., and Bartoli, A. (2018). Isometric non-rigid shape-from-motion with riemannian geometry solved in linear time. *PAMI*.

[Patron-Perez et al., 2015] Patron-Perez, A., Lovegrove, S., and Sibley, G. (2015). A spline-based trajectory representation for sensor fusion and rolling shutter cameras. *IJCV*.

[Purkait and Zach, 2017] Purkait, P. and Zach, C. (2017). Minimal solvers for monocular rolling shutter compensation under ackermann motion. In *WACV*.

[Purkait et al., 2017] Purkait, P., Zach, C., and Leonardis, A. (2017). Rolling shutter correction in manhattan world. In *ICCV*, pages 882–890.

[Rengarajan et al., 2017] Rengarajan, V., Balaji, Y., and Rajagopalan, A. (2017). Unrolling the shutter: Cnn to correct motion distortions. In *CVPR*.

[Rengarajan et al., 2016] Rengarajan, V., Rajagopalan, A. N., and Aravind, R. (2016). From bows to arrows: Rolling shutter rectification of urban scenes. In *CVPR*.

[Ringaby and Forssén, 2012] Ringaby, E. and Forssén, P.-E. (2012). Efficient video rectification and stabilisation for cell-phones. *IJCV*.

[Russell et al., 2014] Russell, C., Yu, R., and Agapito, L. (2014). Video pop-up: Monocular 3d reconstruction of dynamic scenes. In *ECCV*.

[Salzmann and Fua, 2011] Salzmann, M. and Fua, P. (2011). Linear local models for monocular reconstruction of deformable surfaces. *PAMI*.

[Saurer et al., 2016] Saurer, O., Pollefeys, M., and Hee Lee, G. (2016). Sparse to dense 3d reconstruction from rolling shutter images. In *CVPR*.

[Saurer et al., 2015] Saurer, O., Pollefeys, M., and Lee, G. H. (2015). A minimal solution to the rolling shutter pose estimation problem. In *IROS*.

[Saurer et al., 2017] Saurer, O., Vasseur, P., Boutteau, R., Demonceaux, C., Pollefeys, M., and Fraundorfer, F. (2017). Homography based egomotion estimation with a common direction. *PAMI*.

[Schindler et al., 2006] Schindler, G., Krishnamurthy, P., and Dellaert, F. (2006). Line-based structure from motion for urban environments. In *3DV*.

[Schubert et al., 2018] Schubert, D., Demmel, N., Usenko, V., Stückler, J., and Cremers, D. (2018). Direct sparse odometry with rolling shutter. *ECCV*.

[Shoemake, 1985] Shoemake, K. (1985). Animating rotation with quaternion curves. In *SIGGRAPH*.

[Sturm et al., 2012] Sturm, J., Engelhard, N., Endres, F., Burgard, W., and Cremers, D. (2012). A benchmark for the evaluation of rgb-d slam systems. In *IROS*.

[Taylor et al., 2010] Taylor, J., Jepson, A. D., and Kutulakos, K. N. (2010). Non-rigid structure from locally-rigid motion. In *CVPR*.

[Torr et al., 1999] Torr, P. H., Fitzgibbon, A. W., and Zisserman, A. (1999). The problem of degeneracy in structure and motion recovery from uncalibrated image sequences. *IJCV*.

[Varol et al., 2009] Varol, A., Salzmann, M., Tola, E., and Fua, P. (2009). Template-free monocular reconstruction of deformable surfaces. In *ICCV*.

[Vasu et al., 2018] Vasu, S., MR, M. M., and Rajagopalan, A. (2018). Occlusion-aware rolling shutter rectification of 3d scenes. In *CVPR*.

[Vautherin et al., 2016] Vautherin, J., Rutishauser, S., Schneider-Zapp, K., Choi, H. F., Chovancova, V., Glass, A., and Strecha, C. (2016). Photogrammetric accuracy and modeling of rolling shutter cameras. *ISPRS Annals*.

[Von Gioi et al., 2010] Von Gioi, R. G., Jakubowicz, J., Morel, J.-M., and Randall, G. (2010). Lsd: A fast line segment detector with a false detection control. *PAMI*.

[Wu, 2011] Wu, C. (2011). Visualsfm: A visual structure from motion system.

[Yang et al., 2018] Yang, N., Wang, R., Gao, X., and Cremers, D. (2018). Challenges in monocular visual odometry: Photometric calibration, motion bias, and rolling shutter effect. *IEEE Robotics and Automation Letters*.

[Zaragoza et al., 2014] Zaragoza, J., Chin, T.-J., Brown, M. S., and Suter, D. (2014). As-projective-as-possible image stitching with moving dlt. In *PAMI*.

[Zhou et al., 2012] Zhou, Z., Jin, H., and Ma, Y. (2012). Robust plane-based structure from motion. In *CVPR*.

[Zhuang et al., 2017] Zhuang, B., Cheong, L.-F., and Lee, G. H. (2017). Rolling-shutter-aware differential sfm and image rectification. In *ICCV*.