



HAL
open science

Local features for RGBD image matching under viewpoint changes

Maxim Karpushin

► **To cite this version:**

Maxim Karpushin. Local features for RGBD image matching under viewpoint changes. Computer Vision and Pattern Recognition [cs.CV]. Télécom ParisTech, 2016. English. NNT: . tel-01483314

HAL Id: tel-01483314

<https://theses.hal.science/tel-01483314>

Submitted on 5 Mar 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



EDITE - ED 130

Doctorat ParisTech

THÈSE

pour obtenir le grade de docteur délivré par

TELECOM ParisTech

Spécialité Signal et Image

présentée et soutenue publiquement par

Maxim KARPUSHIN

03 novembre 2016

Local features for RGBD image matching under viewpoint changes

Supervisor: **Frédéric DUFAUX**
Co-supervisor: **Giuseppe VALENZISE**

Jury

M. Miroslaw Z. BOBER, Professeur, Université de Surrey
M. Jean-Michel MOREL, Professeur, École Nationale Supérieure de Cachan
M. Jean-Luc DUGELAY, Professeur, EURECOM
M. Patrick PEREZ, Distinguished Researcher, Technicolor
M. Stefano TUBARO, Professeur, Politecnico di Milano
M. Giuseppe VALENZISE, Chargé de recherche, CNRS LTCl, Télécom ParisTech
M. Frédéric DUFAUX, Directeur de recherche, CNRS LTCl, Télécom ParisTech

Rapporteur
Rapporteur
Examineur
Examineur
Examineur
Co-directeur de thèse
Directeur de thèse

TELECOM ParisTech

école de l'Institut Mines-Télécom - membre de ParisTech

Abstract

In the last five-to-ten years, 3D acquisition has emerged in many practical areas thanks to new technologies that enable a massive generation of texture+depth (RGBD) visual content, including infrared sensors Microsoft Kinect, Asus Xtion, Intel RealSense, Google Tango, laser 3D scanners (LIDARs). The increasing availability of this enriched visual modality, combining both photometric and geometric information about the observed scene, opens up new horizons for different classic problems in vision, robotics and multimedia. In this thesis, we address the task of establishing local visual correspondences in images, which is a basic task that numerous higher-level problems are settled with. The local correspondences are commonly found through local visual features. While these have been exhaustively studied for traditional images, little work has been done so far for the case of RGBD content.

This thesis begins with a study of the invariance of existing local feature extraction techniques to different visual deformations. It is known that the traditional photometric local features that do not rely on any kind of geometrical information may be robust to various in-plane transformations, but are highly sensible to perspective distortions caused by viewpoint changes and local 3D transformations of the surface. Yet, those visual deformations are widely present in real-world applications. Based on this insight, we attempt to eliminate this vulnerability in the case of texture+depth input, by properly embedding the complementary geometrical information into the first two stages of the feature extraction process: repeatable interesting point detection and distinctive local descriptor computation.

With this objective, we contribute with several new approaches of keypoint detection and descriptor extraction, that preserve the conventional degree of keypoint covariance and descriptor invariance to in-plane visual deformations, but aim at improved stability to out-of-plane (3D) transformations in comparison to existing texture-only and texture+depth local features. In order to assess the performance of the proposed approaches, we revisit a classic feature repeatability and discriminability evaluation procedure, taking into account the extended modality of the input. Along with this, we conduct experiments using application-level scenarios on RGBD datasets acquired with Kinect sensors. The results show the advantages of the new proposed RGBD local features in terms of stability under viewpoint changes.

Keywords: RGBD, texture+depth, keypoint, local image descriptor, local feature.

Table of Contents

1	Introduction	15
1.1	Context and objectives	15
1.2	Contributions	18
1.3	Structure of the thesis	20
2	Background and state of the art	23
2.1	Problematics and motivation	23
2.2	Local features in traditional imaging	27
2.2.1	Corner keypoint detectors	28
2.2.2	Blob keypoint detectors. Scale invariance	29
2.2.3	Conventional local description techniques	31
2.2.4	Binary local features	35
2.2.5	Problem of out-of-plane rotations	36
2.2.6	MPEG standardization of local features	38
2.3	Texture+Depth content description	40
2.3.1	Shape-only descriptors	42
2.3.2	Joint texture and shape description	43
2.3.3	Texture description using shape	44
2.4	Scale spaces	45
2.4.1	Definition	45
2.4.2	Use in feature detection	48
2.5	Performance evaluation of local features	49
2.5.1	Revisited mid-level feature evaluation procedure	49
2.5.2	RGBD image datasets	54
3	Perspective distortions compensation by a local planar normalization	59
3.1	Overview	60
3.2	Proposed normalization approach	61
3.2.1	Estimation of local approximating planes and keypoint filtering	61
3.2.2	Local surface sampling and slant normalization	62
3.3	Experiments and discussion	63

3.3.1	Matching score test	64
3.3.2	Descriptor distinctiveness test	65
3.4	Conclusion	68
4	Binary RGBD descriptor based on a pattern projection	69
4.1	Overview	69
4.2	Proposed descriptor pattern projection	70
4.2.1	Mapping descriptor pattern to scene surface	70
4.2.2	Descriptor computation	73
4.3	Experiments	74
4.4	Conclusion	77
5	TRISK: A local features extraction framework for RGBD images	79
5.1	Overview	79
5.2	Proposed approach	80
5.2.1	The Detector	80
5.2.2	The Descriptor	86
5.2.3	Implementation details	87
5.3	Experiments	88
5.3.1	Compared methods	88
5.3.2	Matching score and ROC	89
5.3.3	Parameter values estimation	90
5.3.4	Visual odometry	93
5.3.5	Note on computational efficiency	95
5.4	Conclusion	95
6	Keypoint detection based on a viewpoint-covariant scale space	97
6.1	Overview	97
6.2	Design of RGBD scale space	98
6.2.1	Laplacian operator definition	98
6.2.2	PDE problem formulation	99
6.2.3	Well-posedness, numerical solution and its causality	100
6.2.4	Relation to Laplace-Beltrami operator	101
6.2.5	GPU implementation of the proposed filter	103
6.3	Proposed detector	104
6.3.1	Candidates selection	104
6.3.2	Candidates filtering	105
6.3.3	Accurate localization	105
6.4	Experiments	106
6.4.1	Viewpoint-covariant filter behavior illustration	106
6.4.2	Repeatability evaluation	108

6.4.3	Scene recognition using Kinect images	109
6.5	Conclusion	111
7	$O(1)$ smoothing operator for non-uniform multiscale representations	113
7.1	Overview	113
7.2	Image smoothing operators for keypoint detection	114
7.3	The proposed filter design	117
7.3.1	Filter kernel definition	117
7.3.2	Continuous response computation	118
7.3.3	Viewpoint-covariant scale space approximation and detector	121
7.4	Experiments and discussion	123
7.4.1	Qualitative assessment	123
7.4.2	Computational time	124
7.4.3	Rotational invariance compared to Gaussian scale space	124
7.4.4	Viewpoint covariant multiscale representation	126
7.5	Conclusion	128
8	Conclusion	131
8.1	Summary	131
8.2	Future research directions	133
	List of Publications	137
	Glossary	140
	Bibliography	141

List of Figures

1.1	An RGBD image from <i>Freiburg</i> dataset, acquired with Microsoft Kinect. In the depth map, the color is mapped to the distance from the camera plane to the observed surface. Pixels where the sensor is unable to measure the depth are displayed in black.	16
2.1	Corners discovered in an image by Harris, GFTT and FAST detectors. 500 corners with the highest score are displayed, <i>OpenCV</i> implementation is used.	30
2.2	Keypoints discovered in an image by different implementations of SIFT and SURF: original implementations (cyan), <i>OpenCV</i> (yellow) and <i>VLFeat</i> (green, not available for SURF). While the both detectors search for distinctive blobs, the resulting keypoint sets produced by different implementations might vary considerably from the points of view of detector sensitivity, number of keypoints, their distribution in the image and dominant directions of descriptor patches. This figure is better viewed in color.	33
2.3	Matching of two images from <i>House</i> sequence (described in Section 2.5.2.1) with SIFT. From 264 positive matches 54 are labeled as true according to the overlap error test.	51
2.4	SIFT descriptor matching using different inter-descriptor similarity scores. Simple distance-based matching is compared to $\rho_{1/2}$ ratio-based matching [22] for standard (blue) and affine normalized (red) SIFT descriptors [116]. To plot ROC, 20K true positive and 20K false positive matches were collected from several image sequences presented in Section 2.5.2.1. Normal SIFT descriptors are more distinctive when being matched using ratio-based score, whereas affine invariant features perform much better with simple Euclidean distance. The best performing scores are used in further experiments in this thesis.	54
2.5	Examples of texture maps from synthetic RGBD sequences used in the mid-level evaluation. From left to right: <i>Arnold</i> , <i>Fish</i> , <i>Graffiti</i> , <i>Bricks</i> , <i>House</i> . In each column: first (reference), center and last view of the corresponding sequence.	55

2.6	Images used for scene recognition task acquired with Kinect 2 sensor. Each column represents a scene taken from different viewpoints, with texture maps followed by their corresponding depth maps.	56
2.7	Images from different sequences of <i>Freiburg</i> dataset.	57
2.8	Image #9 from <i>LIVE1</i> dataset.	58
2.9	An image from <i>KITTI</i> dataset.	58
3.1	Local patch normalization illustration. A keypoint is selected in two input images, and the corresponding local patches are shown. Standard (unnormalized) patches are displayed on Fig. (a) and (c). Patches normalized with the proposed approach are displayed on Fig. (b) and (d). Even though all the four patches are quite similar to each other, the latter two are mainly relied by an in-plane rotation, whereas the former ones – by a more complex transformation.	59
3.2	Estimation of the sampling window size $R(\sigma)$ in the local approximating plane, obtained from descriptor patch size $r(\sigma)$. The corresponding keypoint area S' on the fitted plane is covered by a regular sampling grid which is then projected on the camera plane. The projected grid size is such that it covers the texture keypoint area S	63
3.3	Matching score (left column) and number of features (right column) on synthetic RGBD sequences <i>Arnold</i> , <i>Bricks</i> , <i>Fish</i> and <i>Graffiti</i>	66
3.4	Top row: ROC curves obtained on test data for three different angle ranges. Bottom row: corresponding areas under curves; “3D” refers to the proposed method, “aff” to the affine normalization.	67
4.1	Original BRISK sampling pattern for a keypoint of a unit scale, and an example of its distribution over the scene surface for a keypoint centered at a corner.	69
4.2	Illustration of pattern parametrization components for the keypoint on Fig. 4.1. 71	71
4.3	Image surface parametrization in local camera coordinates.	71
4.4	Matching scores obtained on different sequences with original BRISK detector, SIFT descriptor and the proposed descriptor. The original BRISK keypoints are used with all the descriptors. Black curves represent detector limitation, i.e. numbers of repeated keypoints (<i>repeatability</i>).	75
4.5	Receiver operating characteristics obtained on the entire dataset with original BRISK detector, SIFT descriptor and the proposed descriptor. The original BRISK keypoints are used with all the descriptors.	76
4.6	Areas under ROC curves obtained on test sequences for different ranges of out-of-plane rotations, and on the entire dataset (corresponding curves are presented in Fig. 4.5).	76

5.1	The proposed TRISK pipeline architecture.	79
5.2	Computing of local axes \vec{q}_1 and \vec{q}_2 using surface normal \vec{n} (left) and examples of local axes fields on images from <i>Arnold</i> and <i>Bricks</i> sequences (right). \vec{q}_1 is shown in cyan, \vec{q}_2 in yellow. This figure is best viewed in color.	81
5.3	Illustration of application of Accelerated Segment Test (AST) in standard image axis versus local axes derived from the depth map. A corner viewed under a large angle projects itself at a nearly straight contour on the camera plane, so that the corner test in standard image axes fails causing a repeatability loss.	84
5.4	BRISK descriptor sampling pattern from the original implementation (left) and its mapping to the surface through local planar normalization (right). .	87
5.5	Matching score and receiver operating characteristics demonstrating repeatability and distinctiveness of the compared detectors and descriptors, mainly under out-of-plane rotations (<i>Bricks</i> and <i>Floor</i> sequences) and scale changes (<i>House</i> sequence). Computed on synthetic RGB data. At least 4800 true positive and 4800 false positive matches were selected to plot each ROC curve.	91
5.6	Matching score and receiver operating characteristics demonstrating repeatability and distinctiveness of the compared detectors and descriptors under viewpoint position changes of different kind. Computed on three sequences of <i>Freiburg</i> dataset, acquired with Kinect. In some images in <i>desk</i> sequence and in the whole <i>floor</i> sequence VIP turns unable to detect any feature. . .	91
5.7	Visual odometry with 10 frames skipping on <i>freiburg2_desk</i> sequence (first 500 frames): translation (left) and rotation (right) errors. VIP fails on this sequence, thus it is not reported.	94
5.8	Visual odometry with 5 frames skipping on <i>freiburg1_floor</i> sequence (first 500 frames): translation (left) and rotation (right) errors. VIP fails on this sequence, thus it is not reported.	94
5.9	Visual odometry with 10 frames skipping on <i>freiburg3_structure_texture_far</i> sequence (first 500 frames): translation (left) and rotation (right) errors. . .	94
5.10	Feature extraction time averaged over images from matching score test (Fig. 5.6). Smoothing filter initialization, local axes computation, AGAST over 3 octaves, keypoint candidates processing (“CP”) over 3 octaves (includes accurate localization, Harris corner test and descriptor computation) and remaining processing times and their standard deviations are displayed. . .	95

6.1	Visual comparison of the uniform Gaussian smoothing (left) and the proposed non-uniform smoothing (right) of the same input image. The latter is propagated “along the surface”: farther objects are less smoothed. An accurate formulation of this principle, referred to as <i>viewpoint covariant behavior</i> , leads to a causal multiscale representation (scale space) allowing for repeatable keypoint detection.	97
6.2	An example of the proposed scale space and Laplacian on a real RGBD image compared to the Gaussian scale space and the corresponding Laplacian. Images in each row present obtained with different levels of smoothing: $\sigma = 5, 10$ and 25 for the Gaussian scale space and $\sigma = 0.1, 0.2$ and 0.5 for the proposed one. Bigger views of images (c) are shown in Fig. 6.1.	102
6.3	Keypoints detected using the proposed method in an image of <i>Bricks</i> sequence.	106
6.4	Test setting to assess the viewpoint covariance of a given smoothing filter .	107
6.5	Two views I_1 and I_2 of <i>Bricks</i> sequence and the reconstruction difference $I_{out} = I_2 - I_{1 \rightarrow 2} $ computed using the test setting on Fig. 6.4 without filter (c), with Gaussian filter (d), Perona and Malik filter (e), and our filter (f).	107
6.6	Repeatability score on synthetic RGBD sequences in function of angle of view difference between reference and test images.	108
6.7	Accuracy of scene recognition on the images of Fig. 2.6. The left bars (<i>complete</i>) are computed by matching a query image to all the remaining 74 images in the dataset. In the <i>single reference</i> classification, instead, each image is classified using a set of 15 randomly selected reference images (one per class). In this case the reported results are the average over 1000 repetitions, corresponding standard deviation is displayed.	111
6.8	Raw (putative) feature matches between two RGBD images from <i>Board</i> scene obtained with affine-covariant descriptors on top 1000 keypoints in each image. Left: SIFT detector (243 matches), right: the proposed detector (419 matches).	112
7.1	Bi-dimensional Gaussian filter kernel of unit variance (left) compared to the proposed filter kernel (right). The latter provides a more accurate (closer to the Gaussian) output than the box filter, but the convolution may still be computed in $O(1)$ operations as the kernel surface is polynomial.	113
7.2	Integral image principle [68]. For a given function, the integral over any rectangle AD may be computed immediately, if for any point X on the plane the integral over OX is known. The latter ones are precomputed once during the filter initialization stage, and form the integral image (the rectangles are denoted by their diagonals).	116

7.3	Qualitative comparison of the Gaussian and the proposed multiscale representations for an RGBD image from the <i>LIVE1</i> dataset. In the standard Gaussian scale space σ is constant within each image. In the proposed multiscale representation σ varies, but $\hat{\sigma}$ remains constant. Images in rows (b) and (d) obtained by subtracting adjacent smoothed images from rows (a) and (b).	122
7.4	An input image fragment of 600*600 pixels and filter outputs for $\sigma = 20$. . .	124
7.5	Rotational invariance of the proposed filter vs the box filter. An image is rotated, smoothed and rotated back. The result is then compared to an image smoothed without rotations. Averaged results of 5 images and 3 levels of σ per each rotation angle are shown.	126
7.6	Matching scores subject to in-plane rotations achieved by SIFT features detected with different filters. Three different image sequences of 1296*864 pixels representing in-plane rotations are matched against the corresponding upright images.	127
7.7	Keypoint repeatability obtained with different detectors on two synthetic RGBD sequences.	127
7.8	Repeated keypoints of the proposed detector in two views of <i>Bricks</i> scene. 1257 and 1109 keypoints are detected in each image, 34.9% repeated for $\eta = 0.5$	129

List of Tables

2.1	A classification of the most common visual deformation classes in the context of feature matching robustness. The degree of stability of a given feature extraction approach can be assessed by its capacity to perform well when a deformation of the corresponding class is present between the two matched images. We denote different classes with “G” for geometric deformations and “P” for photometric ones, ordering them in each group by arguably increasing complexity from the feature matching points of view.	25
2.2	Classification of described local features from the point of view of invariance to geometrical visual deformations.	45
2.3	General characteristics of synthetic RGBD sequences used in the mid-level evaluation of local features.	55
5.1	Summary of compared methods.	89
5.2	Minimal, average and maximal number of features extracted from each scene. Minimum and maximum values per row are highlighted in green and yellow.	92
7.1	Interpolation coefficients for integral images of the image moments.	121
7.2	Computation times for different filters in function of input image resolution and smoothing level σ . Each value is averaged over 10 repetitions. Tested on a Windows 7 machine with 12-core 3.5 GHz Intel Xeon CPU, 16 GB RAM.	125

Chapter 1

Introduction

1.1 Context and objectives

With the ongoing advances in computer vision, robotics, image acquisition and processing, a number of previously unfeasible application scenarios have nowadays become practical. Arguably the most exemplary of such scenarios is autonomous driving [1, 2]. It assumes a computer-controlled regular vehicle equipped with a set of sensors, which is able to drive within the regular road infrastructure safely for its passengers and the surroundings. Such a scenario requires a number of problems to be solved reliably and efficiently: data resulting from low-level visual tasks, such as navigation, odometry, object recognition, object tracking, is pipelined to higher-level decision taking mechanisms, e.g., those that control the vehicle behavior. Another similar scenario becoming intensively popular and requiring solutions to the same tasks is related to unmanned aerial vehicles (UAV), their guidance and automatic flight [3].

Efficient computer vision algorithms solving such basic visual tasks have been developed. Both in navigation-related scenarios [4] and in image semantic-related problems, such as image retrieval [5–7], the aforementioned low-level visual processing relies on a shared basic task, referred to as *visual correspondence problem*. The correspondence problem consists in retrieving a set of corresponding points between input images, that represent the same physical locations with respect to the observed content. *Local visual features* describe those points that allow to trace reliably such correspondences. The process of researching such points in images is referred to as *local features extraction*, whereas the correspondences are established during the process of *feature matching*.

Local features have been thoroughly elaborated for traditional images and have nowadays arrived to the exploration of different alternative modalities of visual content representation, including multispectral and infrared images [8], synthetic aperture radar (SAR) images [9], lightfield (plenoptic) images [10], 3D meshes with or without associated photometric information [11, 12], point clouds [13], depth maps [14], and RGBD or texture+depth images.

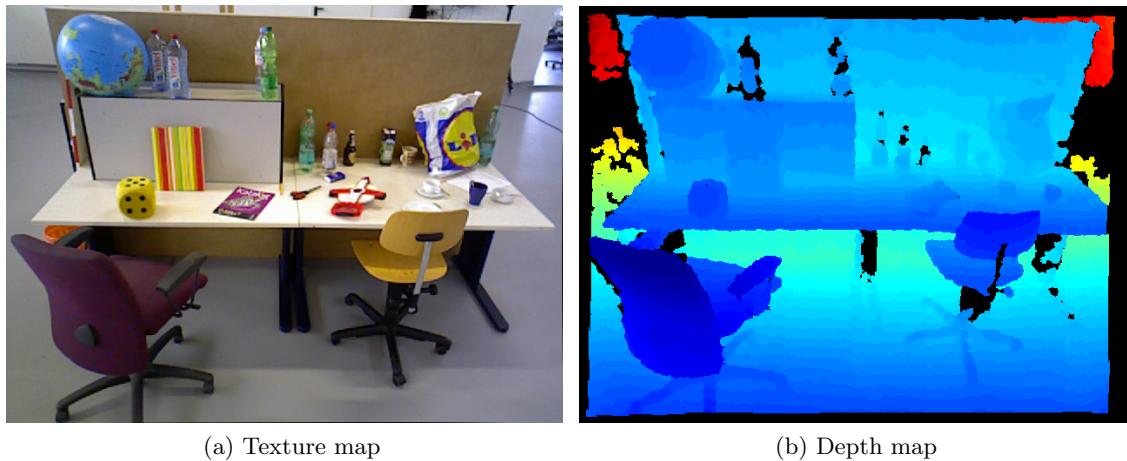


Figure 1.1 – An RGBD image from *Freiburg* dataset, acquired with Microsoft Kinect. In the depth map, the color is mapped to the distance from the camera plane to the observed surface. Pixels where the sensor is unable to measure the depth are displayed in black.

This thesis is focused on the latter kind of visual content, *texture+depth images*. Such a format consists of two images, one describing the photometric information (RGB or grayscale image, referred to as *texture map*), and another one storing the distance from the camera plane to the corresponding point of the scene at each pixel (*depth map*, an example is given in Fig. 1.1). Thanks to this latter image, compared to the conventional RGB or grayscale images, RGBD content carries an explicit complementary geometrical information about the observed scene. This is particularly valuable for computer vision problems: as binocular human vision provides a perception of distance complementing the photometric signal, to a similar extent RGBD imaging may empower the capabilities of a robot to solve different low-level visual tasks.

The motivation to use this type of content in different conventional applications comes from the market: the availability of depth acquisition devices has been constantly increasing over the past few years, whereas the usual RGB or grayscale cameras are already easily affordable. There are several techniques and devices widely used to acquire depth maps, including:

- desktop infrared sensors such as Microsoft Kinect [15] and ASUS Xtion [16],
- mobile infrared sensors such as Intel RealSense, Structure Sensor for iPad,
- mobile devices equipped with both color and depth cameras such as HTC One M8 and Google Tango,
- laser scanners (LIDARs),
- depth estimation from disparity using multiple views or Structure from Motion techniques [17].

It is worth noticing that, in the aforementioned scenario of autonomous driving, most

if not all currently existing prototypes of self-driving vehicles are equipped with a range scanner [2].

While depth maps describe the geometry of the observed surface, similarly, for example, to meshes, they are stored and managed as ordinary images. Such a regularly structured representation allows for efficient access and, consequently, easier basic processing operations, such as interpolation, filtering, subsampling, etc. This is advantageous in practice from computational efficiency and algorithmic design points of view. Putting the texture and depth channels together allows to process them as a whole image using existing conventional image-level approaches. Regarding the correspondence problem, such a setting requires a dedicated discussion, since little work has been done so far on local features for texture+depth content. More attention has been paid to the correspondence problem of the depth maps alone, giving the following essential conclusions on the advantages brought by range imaging.

- + Depth maps are completely insensitive to lightning conditions. This is an important advantage with respect to standard photometric grayscale or color images, which exhibit high sensibility to the illumination changes when searching for local correspondences [14].
- + Depth maps describe explicitly the scene geometry. Since some applications of local correspondences are related to geometrical properties, e.g. visual odometry or object tracking, the depth information directly becomes of interest. As an example, the depth map allows to eliminate scale ambiguity when estimating the camera pose, which is not possible when using only photometric information without a special calibration procedure [18].

On the other hand, when using the extended modality, one immediately observes the following.

- In contrast to the texture map, the depth map is a much more noisy and thus less reliable source of information. This is particularly relevant to low cost infrared depth sensors such as Kinect: often its depth maps are not only noisy, but scattered, i.e., they contain entire regions with undefined depth. This might require specific interpolation techniques, that may allow to restore the missing information and keep the content coherency between its depth and texture counterparts. For this reason, the simple augmentation of texture information by depth might lead to a degraded performance or an unstable behavior.
 - The extended modality unavoidably requires an additional computational effort related to a specific pre- or postprocessing. For example, since RGB and D are typically acquired by different sensors, they have to be aligned to compensate for the parallax. Thus, when involving depth maps into an existing time-aware application scenario, some processing time must be reserved.
-

On the basis of these opposed considerations, a natural question arises, which is put in the focus of this thesis since it has not yet been exhaustively investigated in literature: how can texture and depth be fused together in order to provide more efficient algorithms solving the correspondence problem, compared to approaches that use either only texture or only depth information? Or **what are the qualitative and quantitative advantages of using texture+depth content in the context of visual correspondence problem?**

Attempting to answer it, in this thesis we proceed with the following methodology.

- We first investigate state-of-the-art approaches of RGB, depth and RGBD local features extraction subject to their *covariance* and *invariance* under different kinds of visual deformations. This analysis is based on an elaborated discussion of three key feature stability-related concepts: (i) keypoint covariance, (ii) descriptor invariance and (iii) invariance by design.
- Based on state-of-the-art techniques of local feature extraction, we design alternative approaches that assume an RGBD input and aim at finding reliable correspondences between grayscale texture+depth images in real-world conditions, challenging for standard photometric-only features. These conditions comprise substantial changes in camera position and orientation with respect to the observed scene, that cause considerable perspective deformations in the texture image between matched views.
- We evaluate the performance of the developed local features compared to state-of-the-art approaches in different application scenarios together with a standard mid-level repeatability and distinctiveness evaluation, which is also revisited taking into account the RGBD modality.

1.2 Contributions

The following contributions, mainly to the field of RGBD image matching with local features, are presented in this thesis. The published articles are reported in List of Publications.

1. We investigated how a local planar normalization of texture descriptor patches impacts the performance of local features under out-of-plane rotations, and developed an alternative deterministic derivative-free approach of local planar normalization. This contribution is detailed in the following article:

M. Karpushin, G. Valenzise, and F. Dufaux, “Local visual features extraction from texture+depth content based on depth image analysis,” in *Proceed. of IEEE Intern. Conf. on Image Processing*, (Paris, France), October 2014.

2. We developed a generic technique of binary descriptor pattern mapping onto the scene surface defined by the depth map, allowing to render a surface-intrinsic binary feature describing the photometric information given by the texture map. This is presented in the following article:
-

M. Karpushin, G. Valenzise, and F. Dufaux, “Improving distinctiveness of BRISK features using depth maps,” in *Proceed. of IEEE Intern. Conf. on Image Processing*, (Québec city, Canada), September 2015.

3. We designed a complete feature extraction pipeline for texture+depth content. The proposed approach consists of a corner detector and a binary descriptor, both using the depth information to extract keypoints and their binary signatures from the texture map, in such a way to be robust to arbitrarily complex viewpoint changes. The following article describing this work is submitted:

M. Karpushin, G. Valenzise, and F. Dufaux, “TRISK: A local features extraction framework for texture+depth content matching,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2016, **submitted**, currently under revision.

4. We proposed a scale space formulation for the texture image that exploits the surface metric given by the depth map. This is done by means of a Laplacian-like operator, defining a non-uniform diffusion process for the texture image. This is presented in the following paper:

M. Karpushin, G. Valenzise, and F. Dufaux, “A scale space for texture+depth images based on a discrete Laplacian operator,” in *IEEE Intern. Conf. on Multimedia and Expo*, (Torino, Italy), July 2015.

5. Based on the proposed scale space, we designed a multiscale blob detector for texture+depth images. The scale space simulation is implemented using GPU, which allowed substantial saving of computational time. This is discussed in details in the following article:

M. Karpushin, G. Valenzise, and F. Dufaux, “Keypoint detection in RGBD images based on an anisotropic scale space,” *IEEE Trans. on Multimedia*, vol. 18, no. 9, pp. 1762 – 1771, 2016.

6. We proposed a general purpose smoothing filter based on integral images and suitable for accurate spatially varying smoothing. This work is presented in the following paper:

M. Karpushin, G. Valenzise, and F. Dufaux, “An image smoothing operator for fast and accurate scale space approximation,” in *Proceed. of IEEE Intern. Conf. on Acoustics, Speech and Signal Processing*, (Shanghai, China), March 2016.

7. Using the proposed smoothing operator, we designed an alternative multiscale keypoint detector for texture+depth images, which also performs well under significant changes in viewpoint and requires an affordable computational effort. This is presented in the following article:

M. Karpushin, G. Valenzise, and F. Dufaux, “Keypoint detection in RGBD images based on an efficient viewpoint-covariant multiscale representation,” in *Proceed. of Europ. Signal Processing Conf.*, (Budapest, Hungary), EURASIP, August 2016.

1.3 Structure of the thesis

This thesis is structured in 8 chapters.

- **Chapter 2** discusses the background of image matching through local features and existing approaches for both conventional images, range images and texture+depth images. In this chapter we also discuss the key concepts related to the stability of local features under different visual deformations, and elaborate a systematic description of the most common visual deformation classes. A standard keypoint repeatability and descriptor distinctiveness assessment procedure, used in the following chapters, is revisited taking into account the depth information.
 - The novel technical contributions of the thesis begin in **Chapter 3**, by addressing the second stage of the local feature extraction, which is the descriptor computation. To begin, we assume that the interesting points are given by a conventional approach from the texture part of the input image. In this chapter, we investigate how planar normalization of local descriptor patches may compensate for perspective distortions and what is its effect on descriptor performance. Specifically, we design a generic technique of local planar normalization and test it within two different descriptors.
 - In **Chapter 4** we extend the local descriptor normalization idea to binary descriptor pattern projection on the scene surface. Its impact on both feature repeatability and discriminability is evaluated experimentally and compared to traditional local features and a binary texture+depth descriptor.
 - Based on the preceding considerations, in **Chapter 5** we design TRISK, a complete feature extraction pipeline for texture+depth images. The key idea consists in involving the surface metric, derived from the depth map, into the texture map processing, in the form of *adaptive local axes* replacing the regular image coordinates. This allows to apply Accelerated Segment Test (AST) in an intrinsic way to the scene surface and render keypoints more stable. This test lies at the basis of the proposed detector. The proposed descriptor is based on an efficient local planar normalization resulting from the local axes. The experimental validation of the pipeline is performed in two scenarios, including a visual odometry application on real RGBD data acquired with Kinect, and in comparison to several state-of-the-art techniques of RGB and RGBD image matching.
 - After addressing the corner keypoints, in **Chapter 6** we move to another kind of commonly used keypoints in traditional imaging: blob keypoints. This chapter mainly describes the design of our proposed scale space for texture+depth content. A formal definition of the diffusion process generating the scale space and a theoretical analysis of its properties are presented, justifying its application to keypoint detection. We also
-

discuss a GPU-based implementation of the numerical scheme simulating the diffusion process. The proposed multiscale blob detector, based on the novel scale space is then presented in this chapter. In the experimental part, a viewpoint-covariant behavior of the scale space is assessed, a repeatability analysis of the detector is performed and a scene recognition scenario is employed to assess the keypoint detection performance.

- **Chapter 7** describes the proposed smoothing filter design as well as the alternative computationally efficient blob detection scheme based on this filter. The proposed smoothing operator consists in a convolution of a second order polynomial surface, that provides a more accurate response compared to the box filter due to more smoothed border discontinuities. In this chapter we explain how it can be computed in a constant time at any image location independently to the support size. A multiscale detection scheme, similar to the one in chapter 6, is then used in conjunction with the proposed filter to design an efficient blob detector, demonstrating a good robustness to viewpoint position changes. Several different experiments are used to assess the filter accuracy, performance, speed and repeatability of the proposed detector.
 - **Chapter 8** gives concluding remarks and briefly describes future work perspectives.
-

Chapter 2

Background and state of the art

2.1 Problematics and motivation

The problem of finding local correspondences between images is a fundamental task in vision. The common framework to solve this problem is referred to as *image matching*. It consists of three main stages.

1. **Detection**: each input image is processed independently to find repeatable salient visual points.
2. **Description**: a compact signature representing a neighborhood of each detected point is computed.
3. **Matching**: signature sets from different images are compared, producing a set of correspondences.

Salient visual points are also called as **keypoints**. The finally obtained set of correspondences (**matches**) between keypoints from two given images form the output of the matching process and is sent to the application side. The application analyzes then the matches in function of its needs.

- The number of matches with respect to the overall number of keypoints in each image, as well as the regularity and reliability of matches, allows to measure the degree of similarity of the two processed images from a semantic point of view (i.e., whether similar objects are present in the both images, but not necessarily in the same positions or orientations, under the same lighting conditions, etc.). This is a typical basic setting for visual search, scene classification and recognition.
 - The geometry of correspondences allows to establish geometrical relations between the objects presented into the images and/or between the camera positions and orientations from which the images were taken. This is a typical setting for object tracking, simultaneous localization and mapping (SLAM), visual odometry.
-

This thesis addresses the first two stages in a modular way, which are further referred to as *detector* and *descriptor*. Determining how the *local image features* (keypoints+descriptors) are extracted, these two steps form the *feature extraction pipeline*¹. The keypoints signatures are also referred as *descriptors*. Different ways how the descriptors are matched on the last stage of image matching are discussed as a part of evaluation.

Some approaches of image matching require the images to be present for the matching process, whereas some other deal only with *feature representation* obtained after the second step, and do not need the pixel data on the matching stage. The difference between these two paradigms is discussed in details in [19]. This thesis is focused on approaches that provide such feature representations.

The common purpose of image matching is to recognize (semantically) the same content in different acquisition conditions. In other words, the two images being matched typically represent the same or similar content, and are related by a *visual deformation*.

From this perspective, two key concepts may be defined: *covariance* and *invariance*. In order to be able to match the same content within the two input images, we expect to extract the same or very similar descriptors. Therefore, when the observed content undergoes a deformation, the descriptors are expected to remain the same or *invariant*, so that they provide a representation of the content and not conditions of its acquisition. Contrary to the descriptors, the keypoints are expected to be *covariant*, i.e., when a deformation occurs, they are expected to follow it (change accordingly). The keypoints thus depend on the deformation and represent the conditions of acquisition rather than the content itself.

Together these two concepts are generalized to *feature stability*: a *stable feature* is such a feature that allows to match two images related by a corresponding deformation, i.e., it remains detectable when the content undergoes this deformation. To be stable, a feature needs a covariant keypoint and an invariant descriptor. By convention, we sometimes say that a feature is *invariant* when it is stable to a given visual deformation. Another commonly accepted term referring this quality is *feature repeatability*, which is properly defined and discussed later.

The degree of feature stability may be qualitatively measured by the nature of visual deformations the given feature is robust to. A simple classification of the deformations affecting feature stability is given in Table 2.1. To some of the listed deformation classes, e.g., image noise (**P-III**), no feature could be perfectly stable neither totally instable: one can measure the stability quantitatively, for example, adding a progressively increasing noise to the image and trying to match it against its noiseless original. However, to the most part of other deformations, notably geometric ones, a given feature may be *invariant by design*. For example, many existing local features in traditional imaging are invariant by design to the first three classes describing orthogonal transformations in camera plane [4, 20, 21].

¹In what follows, in function of the context, term *feature* refers both to a feature extraction pipeline or to a single keypoint and associated descriptor extracted from an image.

Also, some simple illumination changes (**P-I**) (such as affine $I \rightarrow \alpha I + \beta$ for α and β constant all over the image I) are typically covered too. A classic example of translation, rotation, scale and (partially) illumination invariant feature is SIFT [22].











G-I	In-plane translations		Geometric	Rigid
G-II	In-plane rotations			
G-III	Scale changes			
G-IV	Out-of-plane rotations			Non-rigid
G-V	Affine deformations			
G-VI	Isometric deformations			
G-VII	Non-isometric deformations			
P-I	Affine illumination changes		Photometric	
P-II	Non-linear illumination changes			
P-III	Image noise			

Table 2.1 – A classification of the most common visual deformation classes in the context of feature matching robustness. The degree of stability of a given feature extraction approach can be assessed by its capacity to perform well when a deformation of the corresponding class is present between the two matched images. We denote different classes with “G” for geometric deformations and “P” for photometric ones, ordering them in each group by arguably increasing complexity from the feature matching points of view.

The invariance by design does not imply perfect repeatability: it is practically unfeasible to reach hundred percent of repeatable features, mainly because of the fact that almost all the visual deformations require the image to be resampled and thus introduce pixel-level variations that lead some keypoints to appear or disappear. However, the invariance by design subject to a deformation entails two important effects on the feature performance:

- the number of stable keypoints and descriptors remains high enough when the deformation occurs,
- this number mostly does not depend on the amount of the deformation.

The latter might not happen sometimes, e.g., when due to a deformation a meaningful part of the scene falls out of the image or becomes occluded by other objects, which cause a repeatability loss but is not directly related to the features stability.

To be invariant by design to a specific deformation class, a feature extraction process must involve processing techniques that are themselves covariant and invariant to that

class. In case of in-plane translations and rotations, invariance by design is relatively simple to achieve, since many image processing operations, such as filtering or local extrema detection are translation and rotation invariant, and thus no dedicated effort is required. To render the descriptor rotation invariant, one typically estimates a dominant direction based on image gradients around each keypoint, and then rotates the descriptor patch according to that direction. In-plane scale changes are handled by involving a multiscale representation on the detection stage that allows to discover scale-covariant keypoints, and the descriptor patch is then scaled accordingly to the detected *characteristic scale*. This is further discussed in Section 2.4.

This thesis begins with the observation that the stability to the deformation classes **G-I**, **G-II** and **G-III** in Table 2.1 provided by many traditional local features is somewhat insufficient in practice. In many application scenarios exploiting image matching as a basic task, the observer and/or the objects can move arbitrarily not only in the camera plane, but in all the three dimensions. This causes *perspective distortions*. In the context of local features, they are often seen as an effect of *out-of-plane rotations* (**G-IV**). It is straightforward to see that an arbitrary three-dimensional rigid displacement of an object might be decomposed into a combination of transformations **G-I**, **G-II**, **G-III** and **G-IV**. Reciprocally, an arbitrary three-dimensional displacement of the camera is equivalent to displacements of all the observed objects in an opposite sense. Due to the locality of the features, these deformations become equivalent to unconstrained *local tridimensional rigid deformations* of the observed content, which is arguably the most common kind of visual distortions in practice. Consequently, local features stable under the four transformations are of high practical interest in most application scenarios. Another argument to the importance of **G-I** – **G-IV** and **G-IV** in particular is the occurrence frequency of different deformations in practical scenarios. Non-rigid geometric deformations, such as **G-V**, **G-VI** and **G-VII**, are less frequent in real world applications. Isometric transformations (**G-VI**) may be used to model tissue deformations, but affine deformations (**G-V**) or more complex elastic transformations without the isometry constraint (**G-VII**) are hard to illustrate with a real-world example. However, in-plane affine deformations (**G-V**) are used to approximate perspective distortions. This is discussed in details in Section 2.2.5 and in Chapter 3.

Consequently, the four transformations **G-I** – **G-IV** reveal a joint deformation class, requiring the most attention in practice from the feature invariance point of view. In classic single image vision, when no complementary geometrical information is provided and only photometric data is available, there is no feature invariant to viewpoint position changes *by design*. Classic methods of first four invariance classes may demonstrate acceptable performance in case of limited out-of-plane rotations [4, 22–24]. For more significant deformations of this type, several specific methods exist, e.g. [23, 25]. However, as some authors show [20, 26], without geometry knowledge their performance leaves much to be desired, compared to quantitative feature stability evaluations for deformation classes **G-I**

– **G-III.**

More generally, invariance *by design* to classes **G-I** – **G-IV** is unlikely to be achieved using only photometric information. This reveals a weak point of the paradigm of photometric local features. This problem may be addressed when the image is complemented by a geometry description, where little work has been done so far. In this thesis we focus on this problem, having the goal to define novel local feature extraction tools that involve the geometrical information provided by depth maps into the features extraction process, in order to make the local features *invariant by design* to rigid 3D transformations or perspective distortions caused by out-of-plane rotations and viewpoint position changes.

2.2 Local features in traditional imaging

A large spectrum of problems in vision and multimedia might be reduced to the image matching settled with local features. For this reason, during the past decades local image features have become one of the most valuable concept in these domains. Some representative such scenarios, where the local features may play the major role, involve:

- *Image indexing and content-based image retrieval* [6, 7, 27, 28] exploit the ability of local features to quantify the degree of visual similarity in order to search for visually (semantically) similar instances of a content.
- *Image classification* [29] aims at attributing a category label to a given input image in function of its content.
- *Visual odometry* [30] consists in determining and tracking the observer position solely by using visual sensors.
- *Visual SLAM* [4, 31], a dual problem to the visual odometry, consists in constructing a map of the surrounding and localizing oneself within that map.
- *Tracking by matching* [32] allows to follow a specified target in space and/or time.
- Feature tracking in time may serve as a basis for specific high-level tasks, e.g. crowd behavior analysis [33].

Numerous comparative evaluations of competing feature extraction approaches have been published [4, 20, 21, 24, 26, 34–36]. The demand of the industry of universally applicable local image features stimulated MPEG standardization activities [37, 38].

The idea of content matching through local features has thus been progressively evolving since a long time, although the concept of a robust universal local image feature, i.e. a feature designed regardless a specific application, is relatively modern. This section revises briefly the evolution of this concept for standard images, classifying and covering separately different detection and description principles.

2.2.1 Corner keypoint detectors

A typical keypoint detection aims at detecting positions of distinctive landmarks within a given image in a repeatable way under different image deformations. Its three key ingredients are (a) a local score of each pixel, showing whether it can be considered as a keypoint candidate or not, (b) a non-maxima or non-extrema suppression procedure selecting only local maxima (extrema) of the keypoint score, and (c) further refinement of keypoint candidates, such as application of additional stability criteria or an accurate candidates localization, which is optional.

Harris corner detector and GFTT

A simple way to designate a given image point to be salient is to test whether it is situated on a visually distinctive corner. *Harris corner detector* [39] is often viewed as the first universal detector able to detect keypoints, repeatable under small distortions of the input image. Harris detector is an improved version of Moravec detector [40] and is based on the observation that, for a given grayscale image $I(x, y)$, both eigenvalues of second moment matrix

$$\mathcal{M} = \begin{pmatrix} \left(\frac{\partial I}{\partial x}\right)^2 & \frac{\partial I}{\partial x} \frac{\partial I}{\partial y} \\ \frac{\partial I}{\partial x} \frac{\partial I}{\partial y} & \left(\frac{\partial I}{\partial y}\right)^2 \end{pmatrix} * K \quad (2.1)$$

are high enough in distinctive image corner points. Here $*$ denotes the convolution product and K is a smoothing kernel. It is also shown that the *cornerness* can be measured without explicit computation of the eigenvalues using the following function:

$$R(\mathcal{M}) = \det \mathcal{M} - k \operatorname{tr}^2 \mathcal{M}, \quad (2.2)$$

where k is a constant. If both the eigenvalues are large enough, R takes a positive value. Local maxima of R are thus taken as keypoints.

This technique allows to detect distinctive corners in a translation and rotation invariant way. It also led to a supplementary detection criteria, which variants were subsequently used in other detectors [22, 41]: to ensure that the eigenvalue ratio of \mathcal{M} is high enough, it is sufficient to threshold the quantity

$$\rho(\mathcal{M}) = \frac{\operatorname{tr}^2 \mathcal{M}}{\det \mathcal{M}}. \quad (2.3)$$

This is often referred to as *Harris cornerness* or *Harris corner test*.

A similar detection principle that has had major impact on the interest point detection in conventional images is proposed by Shi and Tomasi [42], and makes part of the keypoint detection framework often referred to as *GFTT*. Instead of thresholding the cornerness

$R(\mathcal{M})$ or $\rho(\mathcal{M})$, GFTT is based on thresholding the minimum eigenvalue of \mathcal{M} :

$$\min(\lambda_1, \lambda_2) > \lambda_{threshold} \quad (2.4)$$

This test ensures that M is well conditioned, which indicates a location in the image that can be tracked reliably, i.e., remains repeatable.

SUSAN

An alternative family of corner detection approaches begins with **SUSAN** detector proposed in [43]. The keypoints are issued from the following per-pixel test:

- pixels within a circle are compared to its center (“nucleus”),
- the number of those that differ from the nucleus is counted (a threshold of the intensity difference is used); this numbers are stacked into a *cornerness image*,
- cornerness local maxima are taken; a lower threshold is also applied to filter out edges and very obtuse angles.

This idea is further developed in **FAST** detector [44]. Here only the circle boundary is used. To cope with image noise, the authors not only count number of pixels that are different from the nucleus, but look for the longest connected segment on the boundary. Again, the lengths obtained for each candidate pixel are put into the cornerness image, whose thresholded maxima are picked as keypoints.

AGAST [45] further refines the same criteria by introducing a decision trees-based approach that reduces computational costs by properly choosing which pixels, and in what order, to compare with the nucleus when looking for the longest segment on the circle border. AGAST detector reaches high repeatability and takes lower time to discover keypoints. Scale-invariant extensions of FAST and AGAST are used in complete feature extraction pipelines [41, 46].

2.2.2 Blob keypoint detectors. Scale invariance

The aforementioned corner detectors allow to select keypoints in a rotation- and translation-covariant way, i.e., if the image is translated or rotated *in camera plane*, the discovered keypoints mostly follow the same transformations. Corners are also quite robust to moderate scale changes and perspective distortions, mainly thanks to the definition of what is considered as keypoint: a distinctive corner remains detectable under limited scaling and out-of-plane rotations. A weakness of corner keypoints consists in the fact that naturally often they are situated on object boundaries. Consequently, the area surrounding the keypoint contains both a part of object and its background. When the object is moving reciprocally to the camera, the background changes. This impacts the robustness of the description.

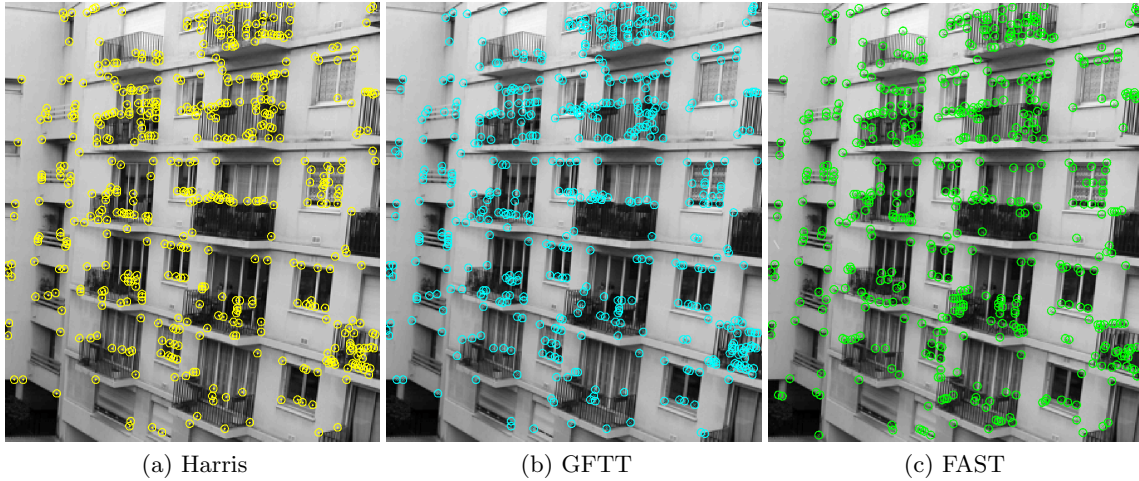


Figure 2.1 – Corners discovered in an image by Harris, GFTT and FAST detectors. 500 corners with the highest score are displayed, *OpenCV* implementation is used.

An alternative definition of keypoints suggests to look for distinctive *blobs* instead of corners. Contrarily to a typical corner, a typical blob-like keypoint represents an image area that has a (not necessarily sharp) boundary. This allows to introduce a notion of a *characteristic size* or a *characteristic scale* of a keypoint, and address properly the problem of scale invariance. To this end, blob-like keypoint detection makes extensive use of the *scale space theory* [47].

An intensive study of image structures at different scales motivated the development of the *scale space* concept [47, 48] and established a more formal notion of the characteristic scale [49]: in addition to the positions in the image, the keypoints were assigned a third degree of freedom: the size of their area of interest in the image. It has been shown [50], that progressive image smoothing and subsampling allows to construct efficiently *multiscale pyramidal representations* revealing image structures of different scales. Such representations can then be used as a scale-invariant keypoint detection modality. More details on this are given in Section 2.4.

Lindeberg [49] explored two basic techniques to detect keypoints of a given scale σ , both based on Hessian matrix

$$H(x, y, \sigma) = \begin{pmatrix} \frac{\partial^2 I}{\partial x^2}(x, y, \sigma) & \frac{\partial^2 I}{\partial x \partial y}(x, y, \sigma) \\ \frac{\partial^2 I}{\partial x \partial y}(x, y, \sigma) & \frac{\partial^2 I}{\partial y^2}(x, y, \sigma) \end{pmatrix}. \quad (2.5)$$

$I(\cdot, \sigma)$ represents the input smoothed to σ , i.e., by convolving the image with a bidimensional Gaussian kernel of variance σ^2 . He showed that both the trace and the determinant of H respond stronger (in absolute value) on distinctive blobs of the given scale level σ , so that their local extrema reveal stable keypoint candidates. The detection criterion based

on the trace of $H(x, y, \sigma)$, which is equal to the Laplacian of $I(x, y, \sigma)$, is used in different scale-invariant keypoint detectors, notably in **DoG** detector which makes part of SIFT [22].

Harris-Laplace (or *Harris-Laplacian*) detector [51] is based on the combination of Harris corner detector with the Laplacian-based characteristic scale selection technique: Harris detector is used to select keypoints on each scale of a Gaussian pyramid and then only those keypoints are preserved that maximize the Laplacian response over scale dimension. In a similar principle, **Hessian-Laplace** detector [25] maximizes both trace and determinant of the Hessian matrix. Such combinations aim not only at selecting distinctive structures at their characteristic scales, but allow also to discard ill-localized blobs, especially the ones along straight edges, as no Harris cornerness neither Hessian determinant exhibit local maxima in such areas.

A simpler detection principle consists in selecting keypoints on each scale independently, without maximization along the scale dimension. It is often referred to as **multiscale detector**. Multiscale Harris, multiscale Laplace and multiscale Hessian detectors are designed in such a way, looking for local maxima of Harris cornerness, trace and determinant of Hessian respectively on each scale level [25].

2.2.3 Conventional local description techniques

Local patch description approaches were evolving in parallel with the keypoint detection.

Conventionally, a *local descriptor* or a *signature* is a point in \mathbb{R}^n , that describes aggregately the visual information present in a local image patch. Points that correspond to visually similar patches are assumed to be close, most commonly in the Euclidean distance sense. The patch itself may serve as a local descriptor with n equal to the number of pixels within the patch. However, in order to provide a compact signature (i.e., with much smaller n , since the matching complexity increases with n), which reflects only relevant visual information invariantly to different deformations, more sophisticated techniques have been developed.

In this section we review some historically significant achievements in local patch description and the most common actual approaches.

Early descriptors

Spin images were initially designed for surface matching [52], but were later extended to local image feature descriptors. A spin image is a histogram-based signature computed at a given point on the surface. The surface is first parametrized in a cylindric coordinate system (r, ρ, ϕ) whose longitudinal axis is aligned with the surface normal. Each vertex then contribute into 2D histogram of its radius r (distance to the axis) and altitude ρ (signed distance along the axis). The angular coordinate ϕ is dropped due to its ambiguity (necessity to locate its zero value). Such a histogram of a relatively low resolution is less affected by surface noise and sampling issues, and is compact enough to be stored and used

as a distinctive aggregated descriptor of the surface at a given keypoint. Lazebnik *et al.* [53] exploited this principle to describe salient points in images. The 2D histogram spans across the radial distance to the patch center and the brightness values of all the pixels within the patch. Such a descriptor provides a high degree of invariance to in-plane rotations by design, but may be less distinctive since it does not store any directional information.

Another common way to describe a patch is based on the use of a set of *invariants*. The descriptor in this case is a numeric vector containing values of different scalar quantities (shortly referred to as invariants), that depend on the pixel content of the patch, but remain invariant under a certain group of geometric and/or photometric transformations. Van Gool *et al.* [54] describe the use of invariants to affine geometric and *affine photometric* changes (**G-V** and **P-I** respectively in Table 2.1). The invariants are typically derived from image *shape* or *intensity moments*. The two moments of order $p + q$ of an image $I(x, y)$ for a patch Ω are defined as follows:

$$I_{x^p y^q}^{\text{shape}}(\Omega) = \iint_{\Omega} x^p y^q dx dy \quad (2.6)$$

$$I_{x^p y^q}^{\text{int}}(\Omega) = \iint_{\Omega} I(x, y) x^p y^q dx dy \quad (2.7)$$

It is straightforward to conclude that, for example, $I_1^{\text{shape}}(\Omega)$ remains invariant under photometric changes, simply because it does not depend on the intensity. It can be shown that the ratio $\frac{I_1^{\text{int}}(\Omega_1)}{I_1^{\text{int}}(\Omega_2)}$ for any two subsets $\Omega_1 \subset \Omega$ and $\Omega_2 \subset \Omega$ remains invariant under contrast changes of the image in Ω . Less trivially, it can also be shown that for a third subset $\Omega_3 \subset \Omega$ the following quantity remains invariant under any combination of deformations of classes **G-V** and **P-I**:

$$\frac{I_1^{\text{int}}(\Omega_1) \cdot I_1^{\text{shape}}(\Omega_2) - I_1^{\text{int}}(\Omega_2) \cdot I_1^{\text{shape}}(\Omega_1)}{I_1^{\text{int}}(\Omega_1) \cdot I_1^{\text{shape}}(\Omega_3) - I_1^{\text{int}}(\Omega_3) \cdot I_1^{\text{shape}}(\Omega_1)} \quad (2.8)$$

Assuming that the patch support Ω is obtained in a repeatable way, the descriptor may be obtained combining different such invariants. Schmid and Mohr [5] use a different set of grayscale invariants obtained from Koenderink's *local jet* derivatives [55] to design a 9-dimensional descriptor applied in an image retrieval scenario. Mindru *et al.* [56] define RGB color invariants based on the image moments and use it to describe region of interests in images in a classification scenario.

Filter bank-based descriptors provide a local patch description by applying a set of filters to the patch and forming the output signature by concatenating scalar characteristics of their responses (typically averaged output). A compact set of filters may be used to extract descriptors in a dense fashion, i.e., at each pixel of the input image, then an aggregating technique, e.g. histogramming or clustering, may be applied to produce a final compact signature of the entire image. Varma and Zisserman [57] tested such a technique with

different filter sets in the context of texture classification. For general non-stationary images, Schmid [58] proposes a spatial-frequency clustering of the densely-extracted descriptors allowing to take into account their distribution all over the image, which renders the resulting descriptor more distinctive in a general image retrieval scenario. Schaffalitzky and Zisserman [59] used a filter bank of 16 complex filters to describe interest points in a classification-by-matching scenario. An extended evaluation of different filter bank-based local descriptors is presented in [20].

SIFT

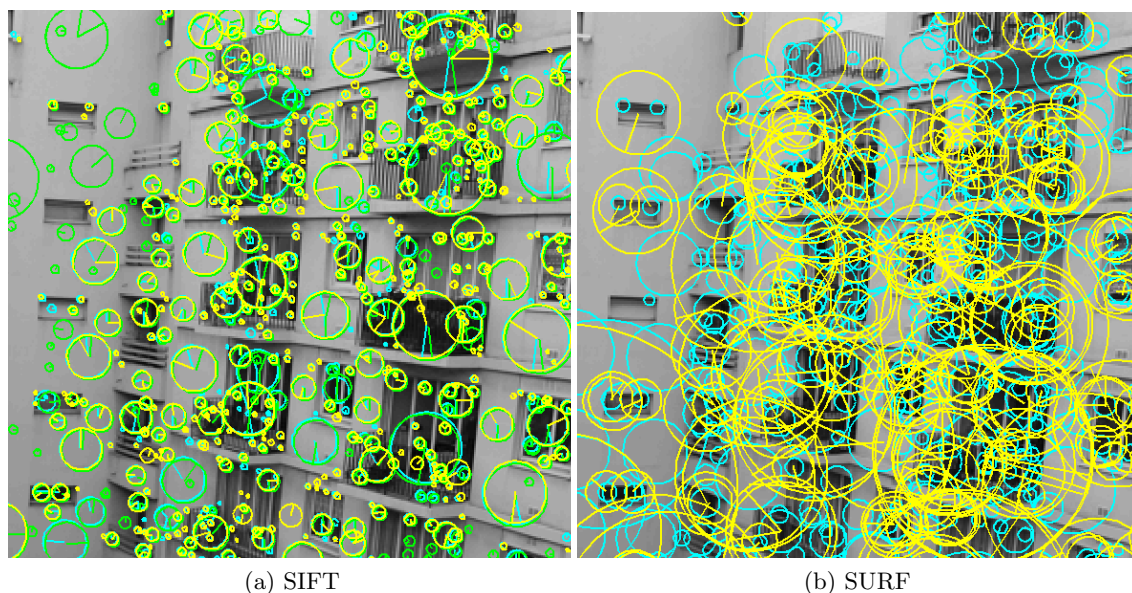


Figure 2.2 – Keypoints discovered in an image by different implementations of SIFT and SURF: original implementations (cyan), *OpenCV* (yellow) and *VLFeat* (green, not available for SURF). While the both detectors search for distinctive blobs, the resulting keypoint sets produced by different implementations might vary considerably from the points of view of detector sensitivity, number of keypoints, their distribution in the image and dominant directions of descriptor patches. This figure is better viewed in color.

SIFT [22] is a complete application-independent framework to detect keypoints and extract corresponding local descriptors that are scale and rotational invariant. Differently to the Harris corner detector, SIFT keypoint detector, often referred to as DoG (*Difference of Gaussians*), looks for distinctive dark and light blobs. Specifically, due to a relation to the heat diffusion equation, the trace of H , which is equal to the standard image Laplacian, might be approximated at a given scale level by a difference of two close levels of the multiscale image pyramid obtained with Gaussian smoothing (a detailed explanation follows in Section 2.4). Consequently, DoG detector exploits local extrema of Laplacian response to reveal keypoint candidates.

Once the initial keypoint candidates are given, SIFT detectors applies additional criteria

to filter out the keypoints that are likely unstable. Specifically,

- the absolute value of Laplacian response is used as indicator of keypoint stability: a weak response denote a potentially unstable keypoint candidate;
- the cornerness measure $\frac{\text{tr}^2 H}{\det H}$ is thresholded to drop keypoints situated along image edges, as they might be unstable to localize;
- an iterative subpixel localization procedure follows, based on interpolation approach from [60]; only candidates whose (x, y, σ) position was successfully interpolated are kept.

An example of SIFT keypoints detected in an image is displayed in Fig. 2.2 (a).

The descriptor is formed by concatenating eight-bin histograms of gradient orientations in the local patch surrounding each keypoint, split in 4×4 subregions, and producing in total a 128-dimensional numeric vector. Invariance to in-plane rotations is achieved by detecting a dominant orientation of the keypoint based on gradient statistics.

Being invariant to translations, in-plane rotations of the image, scale changes and simple illumination variations, SIFT features established a baseline of repeatability and distinctiveness of local features, which still remains competitive. Numerous contributions have been proposed afterwards.

- PCA-SIFT [61] minimizing the descriptor dimensionality by applying principal component analysis principal component analysis (PCA) to its values,
- **GLOH** [20] uses a log-polar patch subregion repartition instead of the original square-like 4×4 , where [62] presents a comparative study of different descriptor topologies,
- [63–66] discuss implementations of SIFT on GPU,
- MPEG CDVS [37] follows SIFT principles for feature detection and description.

SURF

SIFT has been systematically criticized for its relatively high computational complexity. In response to this issue **SURF** were proposed in [67].

The SURF keypoint detector, often referred to as *Fast Hessian*, exploits the Hessian matrix H computed to different layers of a multiscale representation. In contrast to SIFT, SURF uses the determinant of H and not its trace. Moreover, the authors argue the importance of some of scale space axioms, notably the *causality*, with respect to the keypoint stability: rejecting the causality axiom allowed to replace time-consuming convolution with the bi-dimensional Gaussian kernel by the box filtering, which is extremely fast to compute using integral images [68] (more details are given in Chapter 7).

An example of SURF keypoints detected in an image is displayed in Fig. 2.2 (b).

On the descriptor extraction stage, SURF does not compute gradient direction histograms as SIFT does, but exploits the gradient direction in a different way. Using the same 4×4 splitting of the descriptor patch, SURF computes Haar wavelet responses dx_i and dy_i in 5×5 points ($i = 1, \dots, 25$) in each of 64 subregions. Here dx_i and dy_i denote horizontal and vertical Haar wavelet responses at point i of a given subregion. These responses are also efficient to compute using the box filter from a patch sampled according to its dominant orientation, which is also determined using Haar wavelet response statistics. The resulting descriptor is obtained by computing four quantities $\sum_{i=1}^{25} dx_i$, $\sum_{i=1}^{25} dy_i$, $\sum_{i=1}^{25} |dx_i|$, $\sum_{i=1}^{25} |dy_i|$, and concatenating them over all subregions. This results into a keypoint signature of $4 \times 4 \times 4 = 64$ dimensions. This is twice smaller compared to SIFT, which allows faster matching.

2.2.4 Binary local features

Binary descriptors constitute a family of yet more efficient approaches based on a different way of keypoint description.

BRIEF [69] extends the idea of local binary patterns [70], destined for texture analysis tasks, to the keypoint description: a smoothed local patch surrounding the keypoint is sampled in a set of points, obtained values are then compared to each other producing a binary string on the output. This allows to exploit differential characteristics of the local patch in a different way from the gradient histogram-based approaches like SIFT, and leads to a compact signature, represented by a sequence of bits, instead a high dimensional numerical vector like SIFT or SURF. This allows not only to reduce computation time and storage costs, but to match the descriptors in a very efficient way using Hamming distance instead the Euclidean one. It is shown [21] that a binary descriptor of 512 bits achieves similar or better discriminability compared to that of SURF, whereas the signature sizes are equal assuming that SURF feature components are quantified to 8 bits, and the matching of such signatures is faster.

Due to the success of this methodology, a lot of attention has been paid to the design of binary features. **ORB** [41] and **BRISK** [46] are complete pipelines for scale and rotation invariant binary features extraction, based on FAST [44] and AGAST [45] detectors within a multiscale representation of the input image, and estimating dominant orientations of the keypoints. **BRISKOLA** [71] is a further optimization of BRISK for ARM processors, aimed at yet more efficient feature extraction from timing and energy consumption perspectives. **FREAK** [72] is a rotational-invariant binary descriptor, whose pattern is designed taking into account certain aspects of human visual system, such as distribution of retinal receptive fields in human eye and saccadic motion.

While many binary features use handcrafted sampling patterns, ORB learns the pattern points positions that maximize the discriminability. Other approaches propose more

advanced techniques of learning a discriminative binary representation for local descriptor patches. *BinBoost* [73] involves an AdaBoost-based classifier to learn a discriminative hash function, mapping a given descriptor patch to a binary string. A binary descriptor generation through learning-based sparse hashing from non-binary gradient-histogram based descriptor makes part of *CARD* [74].

LATCH [75] is a binary descriptor based on yet a different way of obtaining the output binary string from a patch. Instead of using the idea of the original LBP, it is based on three-patch LBP [76], where three samples p_1, p_2, p_3 are used to obtain one bit on the output: the resulting value is “1” if $|p_1 - p_2| > |p_2 - p_3|$, and “0” otherwise. The samples p_1, p_2, p_3 are not scalar, but are subpatches of the input local patch (the Euclidean norm is taken instead the modulus). The disposition of these subpatches within the input patch is learned to optimize the discriminability.

2.2.5 Problem of out-of-plane rotations

Local features that use only texture information perform generally well under orthogonal deformations and scaling in camera plane **G-I**, **G-II**, **G-III** in Tab. 2.2. However, once the deformations go out of the camera plane, causing 3D distortions, yet rigid, such as perspective deformations, rotations out of the camera plane (**G-IV**), or substantial camera position changes, the matching performance degrades. As an example, the SIFT performance drops quickly when the scene undergoes an out-of-plane rotation of more than 45° [22, 23]. Up to different evaluations [21, 26, 77], this tendency is common for different other approaches, including the ones discussed above.

For this reason, a set of approaches dealing with perspective distortions has been developed.

Local affine normalization

Affine invariant features address the problem assuming that the perspective distortions (**G-IV**) may be locally modeled by affine transformations in camera plane (**G-V**).

Harris-Affine [25] and *Hessian-Affine* [78] detectors extend Harris detector [39] and Hessian matrix-based detector [49, 51] to local affine transformations. Both methods are based on an iterative procedure that estimates an affine frame for each keypoint using second order moment matrix [79]. Specifically, it characterizes the local shape using the intensity gradients. To extract the descriptor a normalization is applied, warping the descriptor patch according to the affine frame. In practice, an affine frame is typically represented with an ellipse, and the normalization warps a given patch such that the ellipse surrounding the keypoint turns into a circle [80]. This idea was first introduced by Baumberg in [81].

Local affine normalization often demonstrates repeatability improvements with respect to standard rotation and scale-invariant features [24, 26, 77, 78]. However, an essential

limitation of the affine covariance paradigm is that perspective distortions are approximated by another class of transformations (**G-V**) that is too general. A typical example is that affine-covariant features do not distinguish between square and rectangle, or circle and ellipse [82]. This may cause a loss of the descriptor discriminability. It is also reported that affine-covariant detectors may be less repeatable under moderate viewpoint angle changes (up to 40°), when standard SIFT demonstrates acceptable performance [22, 23].

Affine simulation

A different way to address the out-of-plane rotation is *simulating* them instead of *normalizing*. The difference between these two concepts may be illustrated using two other invariance classes, **G-II** and **G-III**.

- To achieve the invariance to in-plane rotations (**G-II**), one would typically select a dominant orientation and rotate (normalize) the descriptor patch accordingly. This is *normalization*.
- However, the standard way to deal with the scale invariance (**G-III**) consists in rendering (simulating) a multiscale representation, and then to look for the characteristic scale in it. This is *simulation*.

The idea of affine simulation has initially been proposed in Affine-SIFT (ASIFT) method [23], which is a fully affine invariant image matching solution, outlined in the following steps. The two images being matched are assumed on the input.

- SIFT features are first extracted from two sets of subsampled affinely transformed versions of the two input images.
- The extracted feature sets from these images undergo pairwise cross-matching.
- Top M pairs are selected among the image pairs yielding the biggest number of matches, with a fixed number M . Corresponding affine transitions reflect dominant affine transformations that rely the input images.
- Selected transformations are applied to input images once again, but without subsampling. The obtained sets of descriptors provide relevant features, allowing to match accurately the two inputs.

ASIFT allows for reliable matching, notably for different views of the same objects. The two-resolution scheme allows to reduce the computational complexity to a very reasonable level (authors claim it is comparable to twice the complexity of SIFT). Finally, it is proven by the authors that ASIFT is fully affine-invariant, contrarily to concurrent approaches, e.g. Harris-Affine and Hessian-Affine detectors.

However, the set of features obtained on the last step of the described matching procedure is only relevant to the pair of input images. ASIFT does not produce a compact set of relevant features (a feature representation) for a single given image, as the conventional feature extraction pipelines do, e.g. SIFT, SURF, BRISK. This turns out to be the major limitation of ASIFT when one needs to go beyond a pairwise image matching application, such as large scale image classification or content-by-query retrieval.

A similar simulation-based affine generalization of SURF, *FAIR-SURF* is presented in [83].

2.2.6 MPEG standardization of local features

The industrial demand has stimulated two MPEG standardization activities related to local image features: *CDVS* [37, 84] and *CDVA* [38, 85]. Both standards aim at different spectra of applications based on local features and assuming the “Analyze-then-Compress” (ATC) paradigm [19], i.e., when the features are extracted and transmitted on the content acquisition side.

Compact Descriptors for Visual Search

CDVS offers a standardized tool for image matching aimed at visual search applications. Its normative component covers local features extraction, aggregation and compression, whereas descriptor matching and its application to image retrieval are informative components of the standard.

Assuming given as input an image and a parameter specifying one of 6 possible *operating modes*, the normative part of CDVS [37] specifies

- an interesting point detector dubbed as *ALP*, a version of DoG detector, involving polynomial interpolation of Laplacian of Gaussian response across scale levels within each octave,
- a local feature saliency model aimed at selecting a given number of the most representative keypoints among all the detected ones, taking into account different keypoint characteristics: orientation, scale, distance to the image center, detector response and second order derivative with respect to the scale variable σ , and Harris cornerness (Eq. (2.3)),
- a 128-dimensional local descriptor formed of spatially pooled gradient direction histograms, which is a version of SIFT [22] descriptor,
- a local descriptor lossy compression algorithm, involving a selection/quantization scheme for descriptor components based on their saliency,
- a lossy compression scheme for keypoint positions,

- a local descriptor aggregation scheme, producing a global image signature first applying PCA to local descriptors and then building an aggregated image representation using Fisher kernels [86].

Depending on the operating mode, the resulting image descriptor may contain both a global (aggregated) image signature and a set of local features (keypoint positions and corresponding descriptors). Its maximum size does not exceed a fixed value corresponding to the chosen mode: $2^9 = 512$, 2^{10} , 2^{11} , 2^{12} , 2^{13} or $2^{14} = 16384$ bytes respectively.

As an informative component [84], a feature matching procedure and its application to image retrieval is described. Specifically, it describes

- a global descriptor matching algorithm resulting in a single numerical score, allowing to measure visual similarity between two given images; the score may be thresholded to take the decision whether the two images match,
- a two-way matching scheme of local features (from a query image to a reference image and vice versa), with a weight assigned to each pair of matched features,
- an inliers selection procedure from all the matching pairs, based on distance ratio coherence between keypoints (dubbed as DISTRAT [87]),
- a local features-based numerical score computation, based on the selected inliers and their corresponding weights, allowing to measure visual similarity between the two input images possibly more accurately than the global descriptors-based score.

For a large scale retrieval application, it is suggested to use the score based on global descriptors to generate a shortlist of images for a given query image, since the global score is faster to compute, and then to proceed to a more accurate selection of matching images from the shortlist using the local features-based score.

The normative component of CDVS, describing feature extraction, compression and description, forms part 13 of MPEG-7 standard. Its informative component, describing the matching of local and global descriptors, geometric verification of local descriptors matching and their application to image retrieval, forms part 14 of MPEG-7 standard.

Compact Descriptors for Video Analysis

CDVA has been initiated recently, and is now in an exploration stage. It targets a wider application area with local image features compared to CDVS, which might be outlined in the following three directions [38]: visual search applications (this covers most CDVS use cases), object and event detection, and scene classification.

Being mostly oriented to CCTV needs, CDVA aims at exploiting the temporal aspect of feature extraction and matching to enable the benefits of using video instead of single images as the input content modality, which is an important difference to CDVS. With this

regard, the set of requirements defined for CDVA [85] mostly extends the requirements of CDVS:

- high matching accuracy, compactness, low extraction and matching complexity of the extracted features,
- self-containment: following the ATC paradigm, the extracted features shall not require any additional data for matching,
- coding independence from input image or video formats,
- non-alteration of the content: feature extraction and matching shall not require any alteration of the original content,
- size scalability: descriptors shall be scalable and shall be specified at multiple byte sizes per frame or group of frames,
- descriptors of different sizes shall interoperate,
- spatiotemporal localization: descriptors shall allow localization of matched objects/scenes spatially and temporally in the video stream,
- features shall enable object tracking by tracking the corresponding interesting points,
- partial temporal matching: the extracted features shall support matching of objects or scenes appearing only in a temporal segment of an arbitrarily long query and reference videos,
- robustness to image/video spatiotemporal characteristics, capture conditions and editing operations.

2.3 Texture+Depth content description

Recent development of different extended modalities of visual content, such as Texture+Depth (RGBD) content, 3D meshes, plenoptic images, stimulated the interest of researchers to their feature representations and matching.

RGBD content exhibits a somewhat dual nature. We consider an RGBD image as a four-channel image, where three channels describe the photometric component and the last one describes the geometrical information (in the same way grayscale+depth content might be seen as a two-channel image). Along with this, up to some extent, an instance of RGBD content may be treated as a mesh with associated texture. A number of techniques exists into the literature that propose feature representations for meshes, including those based on local image features.

- **MeshDoG** [12] is a variant of DoG detector for meshes with associated photometric data (a texture mapped onto the surface). Gaussian scale space-like representation is build for the input textured mesh: the 2D Gaussian kernel ((2.9)) is applied to the texture using the surface geodesic distance, and local extrema of DoG-like response are taken as interesting points.
- In the same work [12], **MeshHOG** descriptor is proposed, extending to meshes with or without texture the **HOG** descriptor [88], which was initially developed for standard 2D images.
- MeshDoG idea is further refined in [89], where a subsampling is introduced in the multiscale representation, similarly to SIFT detector, in order to accelerate the detection process.
- **Mesh-LBP** [90] extends the description principle of local binary patterns [70] onto meshes.
- **Harris 3D** [91] and **LD-SIFT** [92] are examples of extension of Harris detector and SIFT respectively to 3D meshes without texture.

Numerous other approaches of feature description are designed for meshes with or without texture [93]. In spite of being often based on similar methodologies, mesh feature description is a separate research area, whereas in this thesis we mainly address related image processing techniques. First, many acquisition devices, including the ones we refer to, allow to acquire images or point clouds rather than 3D models directly, so that the construction of a mesh from the acquired data is a separate processing step. Moreover, some methods dealing with meshes require specific properties to be satisfied, e.g., MeshDoG+HOG [12] require uniformly sampled triangular mesh on input. Typical time-aware feature matching-based applications, such as SLAM or visual odometry, may prefer to avoid any time-consuming preprocessing and deal directly with the acquired data. Second, as a content representation in general, images are simple and well structured, allowing for faster access in memory and easier basic operations of signal processing, e.g. resampling or filtering. This also enables a wider application area for feature matching techniques based on images. Third, the problem of feature stability with respect to perspective distortions caused by changes in viewpoint position seems more appropriate to images. A mesh is typically parametrized into its own coordinate system, i.e., it is defined in an intrinsically independent way to the viewpoint position. In such a setting, the problem of feature stability under viewpoint position changes could be interpreted in terms of robustness of mesh features under resampling with occlusions/disocclusions and possible topology changes, which is a relevant issue for mesh feature extraction, but is rather specific and deserves a separate discussion.

In this section we discuss related image matching techniques that use the geometrical information in the form of a depth map. We split them into three groups: (a) the ones that

do not use photometrical information at all (“shape-only”), (b) the ones that describe the photometrical and geometrical counterparts jointly, and (c) the ones that make use of the geometrical knowledge in order to describe in a better way the photometrical one.

2.3.1 Shape-only descriptors

A family of local descriptors operate only with depth maps. These approaches are advantageous in applications where the geometrical information is prevalent over the photometrical one. In case of specific application needs, such as face recognition, the object texture may be less useful than the geometry. Moreover, elimination of the texture information from descriptor computation makes it perfectly stable to illumination changes, as depth sensors are insensible to the illumination. However, texture details generally reveal a representative clue and may not be neglected.

2.5D SIFT [14] is an extension of SIFT features to range images. 2.5D SIFT is designed for face recognition, having as a goal high feature robustness under different viewpoint positions, since this is essential for most face recognition applications. Keypoint detection in the input range image is based on an adaptation of DoG detector applied to range image data. The descriptor is a concatenation of localized histograms of two types: the originally proposed 8-bin histogram of local orientations and a 9-bin histogram of *shape index*, a coordinate-invariant geometrical characteristic of the surface. The histograms are computed on 3 by 3 pattern, resulting in a vector of $3 \times 3 \times (9 + 8) = 153$ dimensional signature.

The key technique to achieve the viewpoint invariance consists in a normalization of the local frame where the descriptor histograms are computed, i.e. in selecting a proper spatial disposition of each histogram support intrinsically to the object surface. The normalization is based on an estimation of the keypoint normal vector through Gaussian derivatives of the depth map. Once the normal is estimated, the local frame is selected as if the camera is situated in front of the keypoint.

A similar idea of normal-based local frame normalization to achieve the invariance to 3D rotations is used in **NARF** [13]. Custom detector and descriptor are designed for range scans. The interest points are detected on depth borders, defined simply as non-continuous changes from foreground to background. A score is assigned to each border point, depending on surface and its gradient variations in a neighborhood, then the non-local maxima suppression is applied. The descriptor captures local surface shape in a scale-invariant way, using a star-like pattern aligned with the surface normal and an estimated dominant orientation.

PFH [94] and **FPFH** [95] are local descriptors for point clouds based on *surflet-pair features* initially introduced as global descriptors designed for mesh matching [96]. For each pair of points within the local neighborhood (in PFH) or for each point from that neighborhood and its origin (in FPFH) the four geometrical surflet-pair invariants are

computed and aggregated into 16-dimensional histogram. This leads to a shape descriptor, intrinsically invariant to rotations and translations. Scale invariance might be achieved by a proper choice of the neighborhood size. Regarding this, authors also perform a *persistence analysis* of their proposed features on different scales. Informative features are those whose difference from the spatially averaged histogram is larger than a threshold. While (F)PFH are assumed to be extracted densely over the surface, the persistence analysis allows to reveal locations of interesting points.

Other works focus on the description only.

SHOT [97] is a local shape descriptor that consists of spatially pooled histograms of gradient orientations within the keypoint neighborhood, in somewhat mimicking SIFT or GLOH. Concretely, it relies on the angle θ between the surface normal at a given point and a vector, connecting this point to the origin. Histograms of θ are computed in several spatial areas around the keypoint, and are then concatenated to obtain the resulting descriptor. The authors also propose a method to compute a repeatable local reference frame in a given interesting point, which is used to define the coordinates of the keypoint neighborhood to compute the descriptor.

FINDDD [98] descriptor splits the local patch on 4×4 subregions in SIFT fashion and computes a histogram of normal slant directions in each subregion. When cumulating the angles into the histograms it uses a soft voting scheme, allowing to smooth the normal directions and compensate for the noise. Being specifically designed for depth maps, considering them as *structured point clouds*, FINDDD makes use of the well-defined adjacency of sampling points and accelerates the descriptor computation using integral images [68].

2.3.2 Joint texture and shape description

Shape and texture might also be described independently and then combined to form the output descriptor, in a such way that the signature produced at each detected keypoint describes both geometrical and photometrical counterparts simultaneously. Combining the two cues mainly allows for higher feature discriminability.

CSHOT [99] extends the idea of SHOT [97] on four-channel images (trichromatic color + depth). Specifically, the descriptor consists of two parts, one describing the local shape and another one characterizing the colors of points in the keypoint neighborhood. The first one is the previously proposed SHOT descriptor, represented by a set of histograms of the angle θ between surface normal \vec{n}_p in a given point \vec{p} of a local frame and vector $\vec{p} - \vec{p}_0$ (\vec{p}_0 is the local frame origin). The second one is defined in a similar way, but for color: the histograms are computed using a color metric characterizing the difference between colors of points \vec{p} and \vec{p}_0 . To measure the color difference, the authors test *RGB* and *CIE Lab* color representation with several different choices of the metric, concluding that *CIE Lab* color representation, known to be perceptually more uniform than *RGB*, with L_1 norm

leads to a better discriminability of the resulting descriptor.

It is worth noticing that the presence of color information in feature description deserves a separate discussion. Color phenomenon exposes several difficulties to image matching, such as white balance variations or complex lightning changes, which are much easier to cope in the case of grayscale input. With this regard, CSHOT is rather designed for 3D meshes or points clouds, assuming the acquisition-related issues of this kind resolved. However, in spite of the different input modality, CSHOT might be considered relevant to RGBD images too, since it does not pose any special requirements on surface sampling and topology that hinder its direct application to RGBD, as, for example, MeshDOG does.

BRAND [100] is a binary RGBD descriptor that mixes two pattern-based binary descriptors, one for grayscale photometric input and another characterizing the distribution of surface normals near the keypoint. The description principle is similar to BRISK [46] or ORB [41], except that the sampling pattern is not designed manually: its point positions are sampled from an isotropic 2D Gaussian distribution. The dominant orientation is estimated using SURF [67] method applied to the texture map. These design choices result from a performance comparison of several different options. Differently to many other descriptors, BRAND ignores the characteristic scale provided by detector. It re-estimates the keypoint scale from the depth map, in such a way that the descriptor area has the same real-world size. Also, differently to CSHOT, the two signatures are not concatenated to form the output, but undergo bitwise disjunction.

2.3.3 Texture description using shape

In the third and last case, the geometry may be used to provide a robust description of the texture, but is not explicitly incorporated into the resulting descriptors. Differently to the previous case, such techniques are based on texture characteristics that are invariant with respect to the local shape. In this way a consistent deformation of the observed scene that affects both texture and geometry does not impact the descriptor. This reveals a particular interest for invariance to out-of-plane rotations.

VIP [82], **PIN** [101] and **DAFT** [102] extract features from texture+depth content using descriptor patch normalization techniques, aimed at improved stability under significant viewpoint position changes.

PIN performs a local normalization approximating the scene geometry near each keypoint by a plane, and then properly transforming the descriptor patch. SIFT descriptor is then computed from the frontal (normalized) view of the patch. As for the keypoint detection, PIN uses only the texture map, applying a variant of **MSER** [103] region detector. Additional criteria is used to filter degenerate regions.

VIP proceeds in a more global way. It looks for several dominant planes in the scene, then synthesizes corresponding frontal views and extracts SIFT features from each such plane. RANSAC-based scheme is used to discover the dominant planes. This approach

renders VIP features inherently stochastic: with a non-null probability VIP reveals inability to match a given image against itself.

DAFT remains local. Similarly to MeshDoG [12], it projects a *fixed real-world size* Gaussian kernel on the surface and then computes DoG-like response. The real-world size is obtained from the depth map. An approximated algorithm is used to compute the projection efficiently. The description scheme exploits SURF descriptor [67], assuming that the surface is planar in the keypoint neighborhood, and sampling the descriptor values not from the image plane but from the tangent plane to the surface, achieving invariance to perspective distortions.

VIP and DAFT are arguable the first complete frameworks involving grayscale+depth content both for detection of interest points and extraction of local descriptors, aimed at improved feature stability under perspective distortions.

	In-plane translations	In-plane rotations	Scale changes	Out-of-plane rotations	Binary descriptor
SIFT [22]	✓	✓	✓	✗	✗
SURF [67]	✓	●	✓	✗	✗
ASIFT [23]	✓	✓	✓	●	✗
BRIEF [69]	✓	✗	✗	✗	✓
BRISK [46]	✓	✓	✓	✗	✓
ORB [41]	✓	✓	✓	✗	✓
CSHOT [99]	✓	✓	✓	✗	✓
BRAND [100]	✓	✓	✓	✗	✓
VIP [82]	✓	✓	✓	✓	✗
PIN [101]	✓	✓	✓	✓	✗
DAFT [102]	✓	✓	✓	✓	✗

✓ – mostly invariant, ● – partially invariant, ✗ – not invariant

Table 2.2 – Classification of described local features from the point of view of invariance to geometrical visual deformations.

2.4 Scale spaces

To cope with the scale changes, most modern scale-covariant keypoint detectors involve a *scale simulation technique*. It is used to derive a *characteristic scale* associated to the keypoint spatial position using a *multiscale representation* of the input image.

2.4.1 Definition

A multiscale representation of an image is generally understood as a tridimensional function $f(x, y, \sigma)$ in the image intensity domain, whose first two dimensions are spatial and the third one represents *scale*, such that

- $f|_{\sigma=0}$ is simply the input image,
- increasing σ corresponds to a progressive smoothing of the input image, so that σ represents the *quantity of smoothing* induced into the input image.

The concept of scale space is paramount in image analysis and vision [49, 50, 104, 105]. Differently from general multiscale image representations, **scale spaces** are typically expected to satisfy a set of properties constraining the smoothing process. These properties are often referred to as *scale space axioms*. Their different combinations lead to different *axiomatic definitions* of scale space, which are numerous in the literature: as many as 14 different axiomatic definitions are summarized in [104].

According to axiomatic scale space definition given by Koenderink [48], the following properties have to be satisfied by a set of smoothed images to be a scale space:

- **causality** (non-enhancement of local extrema), i.e., any feature at a coarse level of resolution² is required to possess a (not necessarily unique) “cause” at a finer level of resolution;
- **homogeneity and isotropy**, i.e., the smoothing is spatially invariant.

The *causality axiom* here is a crucial property that relates scale spaces to smoothing [47]. This axiom is particularly important in the context of feature detection, as it guarantees that the filtering process does not introduce any features, only revealing the ones present in the original input image.

The progressive smoothing itself sometimes also makes part of the scale space axioms and might be expressed through *semigroup property* [104].

One of the most common linear image smoothing operators that satisfy these axioms together with the semigroup property is the convolutional filter with uniform Gaussian kernel:

$$K_\sigma(x, y) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right). \quad (2.9)$$

Then the scale space for the input image $f_0 : \Omega \rightarrow \mathbb{R}$ is obtained by varying σ :

$$f(x, y, \sigma) = \int_{\Omega} f_0(u, v) K_\sigma(x - u, y - v) dudv \equiv K_\sigma * f_0 \quad (2.10)$$

The importance of Gaussian filter, as well as its principal advantage to other common low-pass filters, raises from the diffusion equation framework. Specifically, it is well-known that the partial differential equation (PDE) problem:

$$\begin{cases} \frac{\partial f}{\partial t} = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \equiv \Delta f \\ f|_{t=0} = f_0 \end{cases} \quad (2.11)$$

²Here *resolution* means *scale* and not the image size.

possesses a unique solution $f(t, x, y) = (K_{\sqrt{2t}} * f_0)(x, y)$. This problem describes the classic heat diffusion process with initial temperature distribution given by f_0 . This relation between the heat diffusion equation and the scale spaces was first studied by Koenderink, and provides an alternative way to define scale spaces, closely linked to its axiomatic definition ([104, 106]):

- Semigroup property represents the ability to obtain $f(\sigma_2, \cdot, \cdot)$ from $f(\sigma_1, \cdot, \cdot)$ for any $\sigma_1 < \sigma_2$ by convolution $f(\sigma_1) * K_{\sqrt{\sigma_2^2 - \sigma_1^2}}$ instead of $f_0 * K_{\sigma_2}$. From the practical point of view, the latter one is more expensive to compute when the smoothing is implemented directly through the convolution product because of a larger kernel support size.
- Homogeneity and isotropy are represented by constant conduction coefficient and scalar diffusivity, i.e., at each image point the heat is propagated equally in all directions. In particular, this implies invariance of the scale space to (in-plane) rotations.
- Causality is formalized through the *extremum principle*: all the extrema of f are situated in the initial image f_0 or on boundaries of $f(\sigma, \cdot, \cdot)$. Referring to the original formulation of the causality given above, a “feature” in this case is an intensity extremum, i.e., a distinctive light or dark blob in the image (this is, however, not what is considered as a “feature” in keypoint detection; for example, SIFT DoG detector looks for extrema of $\frac{\partial f}{\partial \sigma}$ and not f). Thus, this principle guarantees that no spurious details appear in the image when progressively smoothing it.

Not all common low-pass image filters engender scale space as the Gaussian filter does. For example, box filter (convolution product of the image with a constant rectangular kernel) is not rotational invariant and no causality could be stated for it. However, it is used in Fast-Hessian keypoint detector in SURF [67] as a computationally efficient approximation of scale space.

Some axiomatic scale space definitions reject the second axiom in order to achieve a “semantically consistent” smoothing, i.e., to smooth inside the objects but not across boundaries. Examples of spatially adaptive smoothing filters that preserve image structure abound in the literature, from the classic bilateral filter [107] to the recent work in [108] on guided filter that preserves edges using an arbitrary guidance image, just to mention a few. However, no scale space properties have been proved for them so far. The first model of such nonlinear scale space was proposed by Perona and Malik [106], who formulated a non-linear PDE problem in such a way that the diffusion process is controlled by image gradient norm:

$$\begin{cases} \frac{\partial f}{\partial t} = \text{div}(g(\|f\|)\nabla f) \\ f|_{t=0} = f_0. \end{cases} \quad (2.12)$$

Here $g(\cdot)$ is a function that specifies how the *conductivity* depends on the image gradient norm. In contour points the gradient norm is large enough, and $g(\cdot)$ is supposed to take smaller values to prevent the smoothing across the contour. Two such functions were originally proposed:

$$g_1(\tau) = \exp\left(-\frac{\tau}{K}\right) \quad (2.13)$$

$$g_2(\tau) = \frac{1}{1 + \left(\frac{\tau}{K}\right)^2} \quad (2.14)$$

The images produced by the resulting smoothing processes are visually correct: the diffusion controlled in such a way allows to suppress noise while preserving the distinctive contours. Perona and Malik construction had a major impact on the anisotropic diffusion applications in image processing. The original diffusion process, however, might be ill-posed and diverges sometimes, as some authors report [104].

Further scale space generalizations, notably through *anisotropic diffusion filtering* where the diffusivity becomes non-scalar, are discussed in [104, 109]. It is also possible to generalize the notion of scale space to spatio-temporal domains with or without temporal causality [105], point clouds and meshes [110, 111].

2.4.2 Use in feature detection

Feature detection is one of the most significant applications of scale spaces and general multiscale representations.

SIFT [22] and ORB [41] involve Gaussian scale space to detect scale-covariant keypoints. SURF [67] and BRISK [46] use a box filter-based multiscale representation.

KAZE [112] features use an anisotropic scale space as a keypoint detection modality, taking as keypoints the local maxima of Hessian matrix determinant (Eq. (2.5)), similarly to SURF. To describe the detected features, a modified version of SURF descriptor is used, previously proposed in [113]. Three different anisotropic scale spaces are investigated. Comparing them to other feature detectors, authors observed a repeatability improvement in case when Perona and Malik construction is used with the conductivity controlled by the originally proposed function g_2 .

BFSIFT [9] proposes a variant of SIFT features for SAR images registration. These images contain a significant amount of sensor noise, producing a large number of unreliable features on fine scales. To cope with this noise, authors propose a pyramidal image representation based on the bilateral filter [107], by integrating it in SIFT detector instead the Gaussian scale space. The proposed multiscale representation is likely not causal, however, the obtained features allow for a more robust alignment of SAR images compared to standard SIFT. A similar technique with SIFT detector applied to a multiscale representation generated by an edge-preserving filter which is likely not causal is discussed in [114].

A multiscale representation formulation for plenoptic images has been proposed in [10]. 2D Gaussian kernel is modified taking into account the extended modality, leading to anisotropic *Ray-Gaussian* kernel, which is used to design a scale space-like structure and an associated keypoint detector for lightfield content.

2.5 Performance evaluation of local features

In this thesis we distinguish three different ways of performance evaluation of local image features.

- **Low-level evaluation** assumes a basic consistency check of a feature extraction algorithm. It involves very few images (often just a pair), and consists in analyzing basic matching capabilities of a feature, e.g., the number of keypoints found in each image and the number of matches, as well as visual assessment of how they are distributed over the image.
- **Mid-level evaluation** is one of the standard protocols of a systematical local features evaluations. It involves a sequence of images of a given scene, representing a progressively applied visual deformation, e.g., camera movement, progressive blurring, etc. Different images are matched against a reference image, and two main characteristics are computed: *matching score* and *receiver operating characteristics (ROC) curve*, allowing to trace two main axes of feature performance, *repeatability* and *distinctiveness* or *discriminability* respectively. This is further explained in details.
- **High-level evaluation** or *application-level evaluation* consists in using an experimental setup that models a real application scenario in which the features reveal a performance bottleneck, e.g., visual odometry, SLAM, scene recognition. A large number of images is typically involved. Application-specific criteria are defined to evaluate the features performance. Such an evaluation allows to see how the features perform in real-world conditions.

All the three evaluation levels are involved in the contributions presented hereafter. The low-level evaluation is typically reduced to a single figure illustrating a matching of two images, completing the main result. The mid-level evaluation often serves as the main experiment and is present in all the contributions. We detail it in the next section, since it is often used in the rest of the thesis. Application-level evaluations are involved in some of the following contributions; notably we use visual odometry and scene recognition scenarios.

2.5.1 Revisited mid-level feature evaluation procedure

When assessing the quality of features regardless any specific application, the paradigm of local image features defines the following two requirements:

1. Detected keypoints should remain stable (*covariant*) when the image content changes. The descriptors should remain the same (*invariant*), assuming perfectly covariant keypoints. It leads to a feature that is *repeatable*. The *repeatability* is a characteristic of both the detector and the descriptor.
2. Descriptors should be able to distinguish local similarities, i.e., match visual details that are similar up to a certain degree (not only identical ones, to be robust to the noise) and not to confuse visual details that are actually different. So they should be *distinctive* or *discriminative*. The *distinctiveness* mainly characterizes the descriptor, but may also depend on the nature of detected keypoints (e.g., some descriptors might be rather designed for corner areas, whereas some others better describe blobs) and their precision in positioning, scale, etc.

A commonly used method to evaluate the feature repeatability and distinctiveness was initially proposed by Mikolajczyk *et al.* [20, 77] and then reused in a number of comparative feature evaluations [21, 26, 34] as well as in feature design papers [46, 67, 113, 115], just to name a few. In this section we revisit the original approach taking into account the extended modality (presence of “D” in “RGBD”).

A systematic repeatability and distinctiveness analysis is performed with respect to a certain visual deformation class or feature invariance class as discussed in Section 2.1. This characterizes the input data of the evaluation process: an image sequence is taken, containing the same content that undergoes a (progressive) deformation affecting a given invariance class. Few examples follow.

- A given image is rotated around its center with a fixed angle step, producing the sequence. A rotation invariant detector/descriptor should reveal the same keypoints in each image, rotated accordingly, and the same descriptors (as much as possible).
- A progressively increasing amount of noise is added to a given image, producing the sequence. In this case, both keypoints and descriptors are expected to remain equal to what is detected in the original image.
- In our case, a static scene is captured by a moving camera, producing a set of images (views), so that the same objects are present in all the views, but filmed from different positions. As in the first example, the keypoints detected in different views are expected to keep their physical locations, and the descriptors to be the same.

Repeatability and matching score

Once the input data is given, in the first part of the experiment the feature repeatability is evaluated. The following steps describe the evaluation algorithm in details.

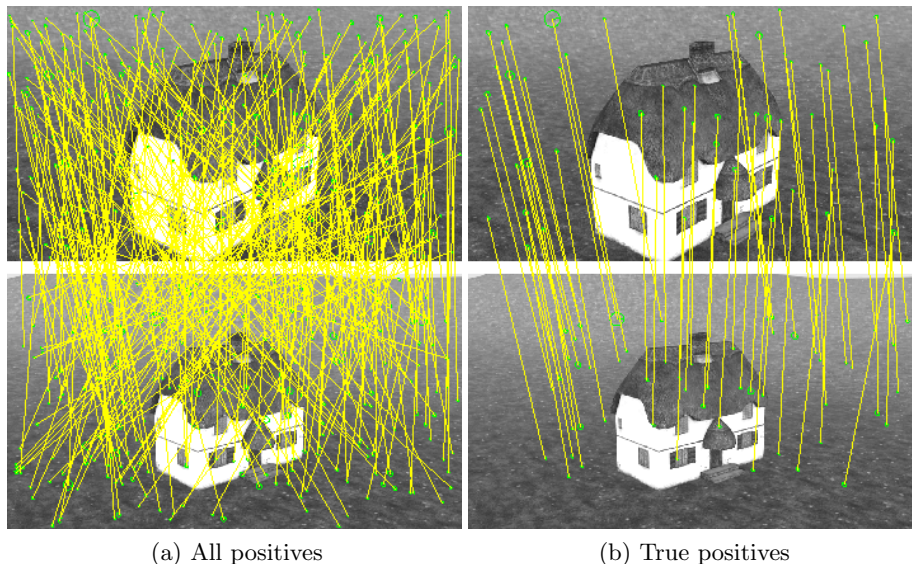


Figure 2.3 – Matching of two images from *House* sequence (described in Section 2.5.2.1) with SIFT. From 264 positive matches 54 are labeled as true according to the overlap error test.

I. Matching against a reference image: in each sequence, its first image is taken as the reference, and the remaining part of the sequence is then matched against this reference.

The matching is performed as follows. A set of local features extracted from the first image is matched against the feature set from the k -th image, $k = 2, 3, \dots$. Let F_k denote the set of features found in the k -th image. The reference descriptors are further referred to as *matchees*, whereas the test descriptors are called *matchers*. The matching consists in finding the closest matcher to each matchee in the sense of an appropriate distance.

- Hamming distance, i.e., number of bit positions where matcher and matchee take different values, is used for all the binary descriptors, such as BRIEF [69], BRISK [46] or CARD [74].
- Euclidean distance between descriptor vectors is taken for non-binary descriptors like SIFT [22] or SURF [67].

II. Matches labeling: the set of matching feature pairs between the two given images (*putative matches*) is split into correct (*true positive*) and incorrect (*false positive*) matches using ground truth.

Two keypoints coming from different images but occupying the same area of the scene are called *repeated keypoints*; they produce a correct match if the descriptors corresponding to these keypoints are matched. The keypoint area overlap is controlled by means of the

overlap error, measured as a function of the keypoint areas A and B :

$$\epsilon(A, B) = 1 - \frac{A \cap B}{A \cup B} \quad (2.15)$$

For the definition of the keypoint areas A and B , we keep two options.

- In the original work [77] A and B represent the elliptical keypoint regions projected on the same camera plane (for example, the reference one). Thus, ϵ represented the degree of overlapping of two “spots” each highlighting a keypoint. This is particularly convenient for scenes that are entirely planar.
- For scenes containing more complex geometry, reprojections of keypoint areas is a more difficult task, and can not be done analytically. For this reason, in most of our further experiments on geometrically rich data, we take tridimensional spheres as keypoint areas, centered at keypoint positions projected on the scene surface. The radius is selected in such a way that the keypoint ellipse may be backprojected from the camera plane onto a 3D circle that fits the sphere boundary.

As the camera positions and orientation matrices are provided, the necessary pixel-level ground truth to compute the overlap is derived by depth maps backprojections.

This provides us with a criterion to determine whether two matching descriptors give a true positive match: a positive match is then labeled as “true” if the corresponding keypoints overlap enough, i.e., if

$$\epsilon(A, B) < \epsilon_0 \quad (2.16)$$

ϵ_0 is a constant typically set to 0.5. If the two keypoints do not overlap, it implies that physically different regions of the scene are signaled as matching, which leads to a false positive match.

Let denote $N_{\epsilon_0}(F_i, F_j)$ the number of true positive matches from set F_i to set F_j . Two properties of this characteristics are worth noticing.

1. In spite of the fact that the overlap error is symmetric with respect to A and B , the number of true positive matches is not a symmetric characteristic of the input descriptor sets, i.e., $N_{\epsilon}(F_i, F_j) \neq N_{\epsilon}(F_j, F_i)$. The reason is that the matcher for each matchee is searched with respect to the set it comes from: when the matchee and the matcher swap, they do not necessarily match each other.
2. It is straightforward to derive that if $\epsilon_1 < \epsilon_2$, then $N_{\epsilon_1}(F_i, F_j) \leq N_{\epsilon_2}(F_i, F_j)$.

III. Matching score computation: the ratio between the number of correct matches and the maximum possible number of matches is reported as *matching score* per image pair.

In the original work [20, 77], the maximum possible number of matches may simply be given by the minimum number of features in the two images. This gives the following formula for the matching score, obtained for k -th test image:

$$\mathcal{M}_{\epsilon_0}(k) = \frac{N_{\epsilon_0}(F_1, F_k)}{\min(|F_1|, |F_k|)} \quad (2.17)$$

However, since the ground truth in our case is provided with depth maps and camera positioning and parameters, it is possible to take into account only the features presented into the area that is present in the two cones of view. This generally decrease the denominator in the formula above, leading to a more elaborated (and more precise) definition of the matching score.

The matching score shows how many features in percentage are actually repeatable in each test image with respect to the reference image. This measures the performance of the entire pipeline (both detector and descriptor). However, it is possible to evaluate only the detection stage by means of the *repeatability score*. It is done by skipping the descriptor matching step, simply checking the overlap criterion for each pair of the keypoints. In such a way only the keypoints are taken into account, and not the descriptors. The portion of keypoints having significant overlap divided by the maximum possible number of matches defines the repeatability score.

To ensure consistency, one must also verify that the spatial density of detected keypoints is not too high. Otherwise, the repeatability measures (both matching and repeatability score) may not rely on the keypoints overlap, since due to the limited image space many keypoints would match each other simply by chance. For this reason the results are often completed by the number of keypoints detected in each image or the overall number of matches.

Distinctiveness

In the second part of the experiment the feature distinctiveness is analyzed. This characteristics is mainly focused on the descriptor performance and depends less on the detector.

IV. Descriptor matches collection: in the same way as it is done in step **I**, all the images of the input sequence (or a subgroup of them that correspond to a certain criterion) are matched against each other, and not just again the reference image. For each image pair the matches are then labeled as it is done in step **II**.

V. Descriptor similarity score computation: an additional score is assigned to each putative match, that depends solely on the descriptors.

The score is often given by the distance between descriptors as explained in step **I**. In some cases, a more complex inter-descriptor score leads to a better distinctiveness. Lowe [22] discovered that ratio $\rho_{1/2}$ of Euclidean distances “matchee – closest matcher”

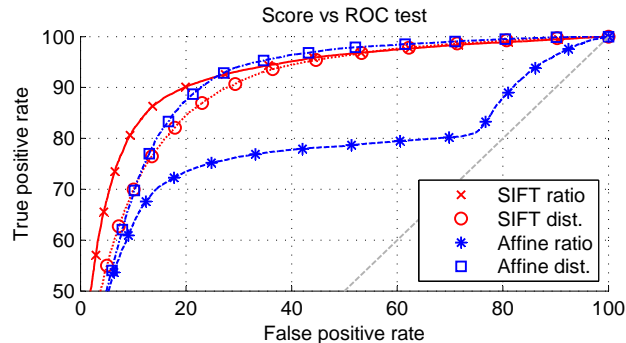


Figure 2.4 – SIFT descriptor matching using different inter-descriptor similarity scores. Simple distance-based matching is compared to $\rho_{1/2}$ ratio-based matching [22] for standard (blue) and affine normalized (red) SIFT descriptors [116]. To plot ROC, 20K true positive and 20K false positive matches were collected from several image sequences presented in Section 2.5.2.1. Normal SIFT descriptors are more distinctive when being matched using ratio-based score, whereas affine invariant features perform much better with simple Euclidean distance. The best performing scores are used in further experiments in this thesis.

and “matchee – 2nd closest matcher” gives a significant discriminability gain with respect to the simple Euclidean distance for SIFT descriptors. This is illustrated in Fig. 2.4.

VI. ROC curve plot: the putative matches labels (true and false) together with the pair scores are used to compute the ROC curves.

The ROC curves are balanced, i.e. an equal number of matching pairs of each class (true and false) is randomly selected among all the matches. The resulting curve shows how discriminative the inter-descriptor similarity score is when it is thresholded to distinguish true and false matches. Concretely, in order to filter out false matches, an application would typically take only those positive matches that have the inter-descriptor score smaller than a threshold. A distinctive descriptor is then such a descriptor that allows to reject as many as possible false positive matches while keeping as many as possible true positive ones during this test.

Matching score allows to judge on the ability of the detector to produce repeatable keypoints as well as on the matching capability of the entire pipeline, whereas ROC mainly assesses how the descriptors are discriminative, e.g., their ability of distinguishing salient visual information in presence of deformations. Together, these characteristics trace the two main axes of the local visual features mid-level evaluation: *repeatability* and *distinctiveness*.

2.5.2 RGBD image datasets

In this section we review briefly RGBD image datasets used in this thesis and our publications.

2.5.2.1 Synthetic dataset for mid-level feature evaluation

A synthetic RGBD dataset was created for the mid-level feature evaluation described in Section 2.5.1. This dataset consists of 5 image sequences (120 images in total of 960*540 pixels). The images are obtained using static 3D scenes, rendered from different viewpoints. The scene content is mainly composed of several publicly available textured 3D models³ with various texture and geometry characteristics. *Graffiti* sequence is synthesized from the frontal view of the original Graffiti sequence [77]. Being synthetically generated, this dataset provides a highly accurate ground truth for the mid-level feature evaluation.

Examples of images are shown in Fig. 2.5. Some characteristics of the scenes are recapitulated in Table 2.3.

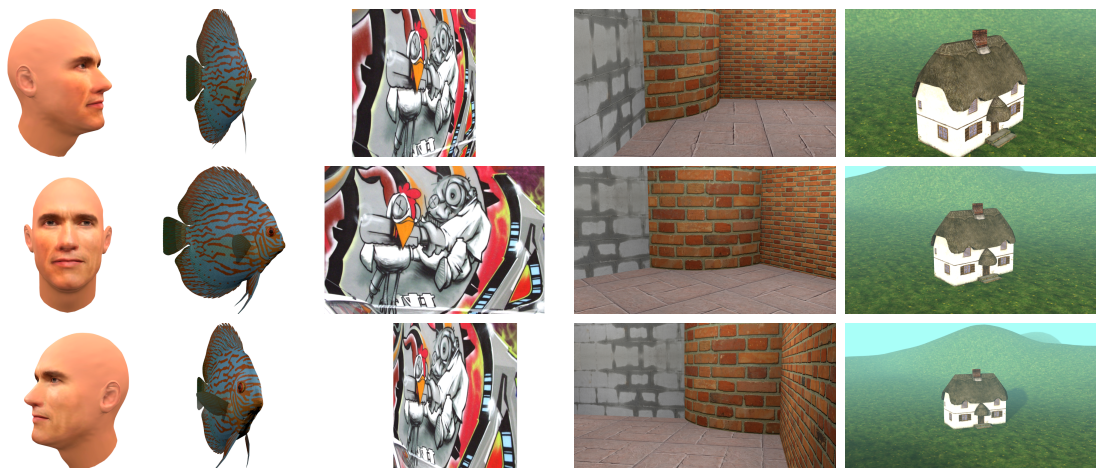


Figure 2.5 – Examples of texture maps from synthetic RGBD sequences used in the mid-level evaluation. From left to right: *Arnold*, *Fish*, *Graffiti*, *Bricks*, *House*. In each column: first (reference), center and last view of the corresponding sequence.

Sequence	Number of images	Out-of-plane rotations range	Scale changes	Texture complexity	Geometry complexity	Occlusions
<i>Arnold</i>	25	120°	No	Low	High	High
<i>Fish</i>	25	120°	No	Medium	Medium	Low
<i>Graffiti</i>	25	120°	No	High	Low	No
<i>Bricks</i>	20	90°	No	Medium	Low	No
<i>House</i>	25	~25°	Significant	High	High	Medium

Table 2.3 – General characteristics of synthetic RGBD sequences used in the mid-level evaluation of local features.

³3D model courtesy of <http://archive3d.net> and <http://www.turbosquid.com>, accessed in Oct.-Nov. 2013

2.5.2.2 Kinect 2 dataset for scene recognition

Using Microsoft Kinect 2 sensor we constructed a dataset of 75 indoor RGBD images, suited to exemplify a scenario of indoor localization of a robot. The dataset is organized in 15 scenes each containing 5 images. The images within the same scene are captured from different positions, but represent essentially the same content. The depth maps are aligned to the texture maps using calibration coefficients carried by the hardware. An overview of the dataset is shown in Fig. 2.6. The images were cropped and subsampled to 720×540 pixels, no other preprocessing is applied. The depth maps are of a standard Kinect quality (may contain regions with undefined depth).

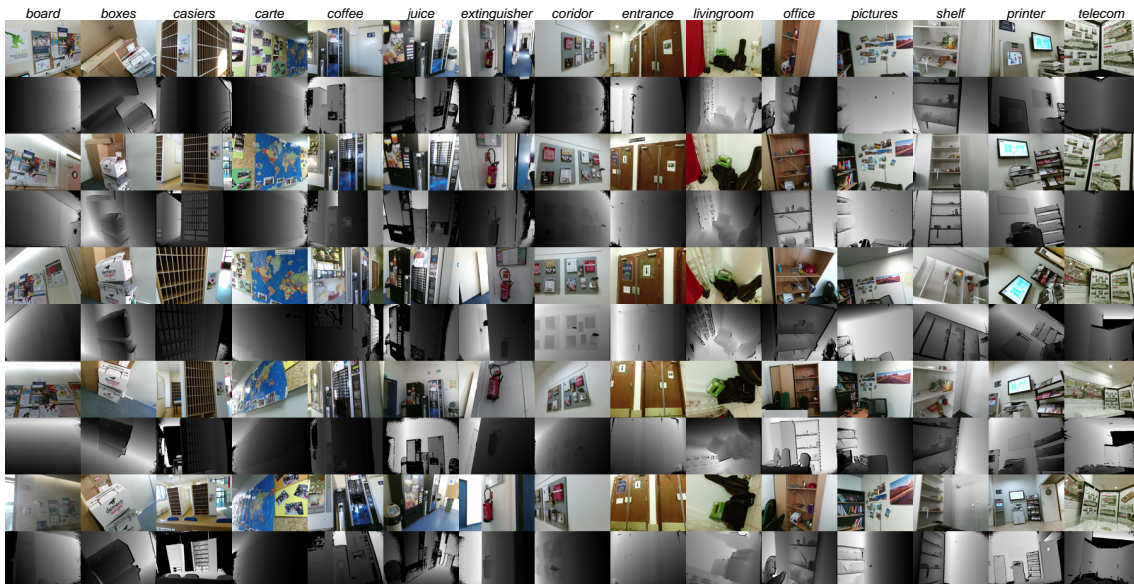


Figure 2.6 – Images used for scene recognition task acquired with Kinect 2 sensor. Each column represents a scene taken from different viewpoints, with texture maps followed by their corresponding depth maps.

2.5.2.3 Freiburg dataset

Freiburg dataset [117] consists of several indoor RGBD image sequences of 640×480 pixels acquired with Microsoft Kinect and ASUS Xtion sensors. Ground truth sensor position and orientation is tracked using a motion-capture system, making this dataset suitable for SLAM and visual odometry experiments. Several images from different sequences are shown in Fig. 1.1 and Fig. 2.7. The depth maps are of a standard Kinect quality (may contain regions with undefined depth).

2.5.2.4 LIVE1 dataset

LIVE1 dataset [118–120] consists of 12 outdoor RGBD images of 1280×720 pixels acquired with a DSLR camera and a laser range scanner. We use this dataset for qualitative

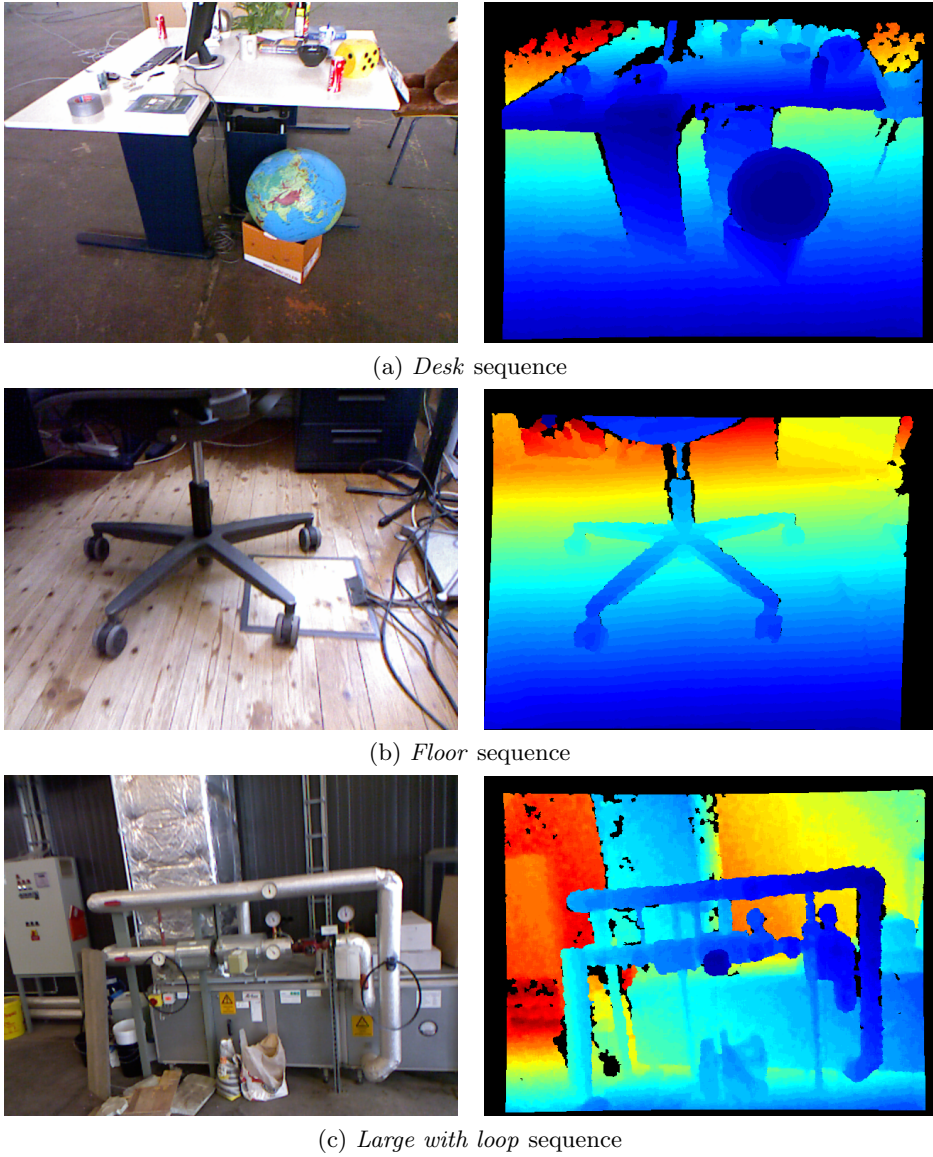


Figure 2.7 – Images from different sequences of *Freiburg* dataset.

assessment of image smoothing processes related to feature detection. An example of an RGBD image from this dataset is displayed in Fig. 2.8.

2.5.2.5 KITTI dataset

KITTI dataset [2] consists of outdoor image sequences acquired with grayscale cameras and a laser range scanner (LIDAR). The sensors are installed on a car together with a GPS and an accelerometer-based motion tracking system that provide the ground truth data for visual odometry experiments. The images are of 1382×512 pixels, the depth information is provided in the form of unstructured point clouds together with the necessary tools to render the depth maps. An example of an image is presented in Fig. 2.9.

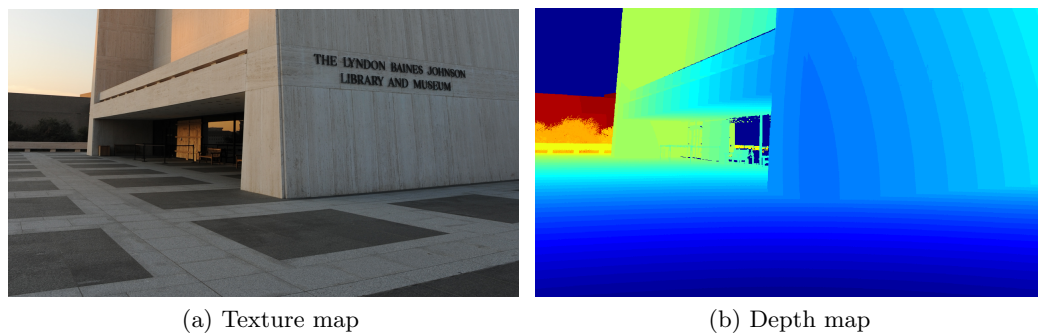
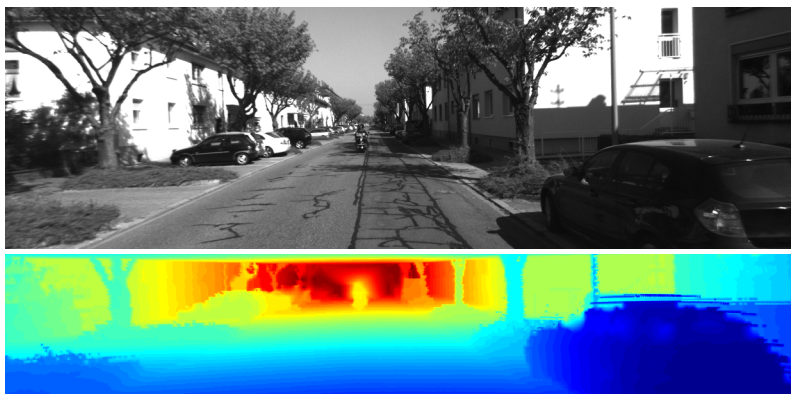


Figure 2.8 – Image #9 from *LIVE1* dataset.



Chapter 3

Perspective distortions compensation by a local planar normalization

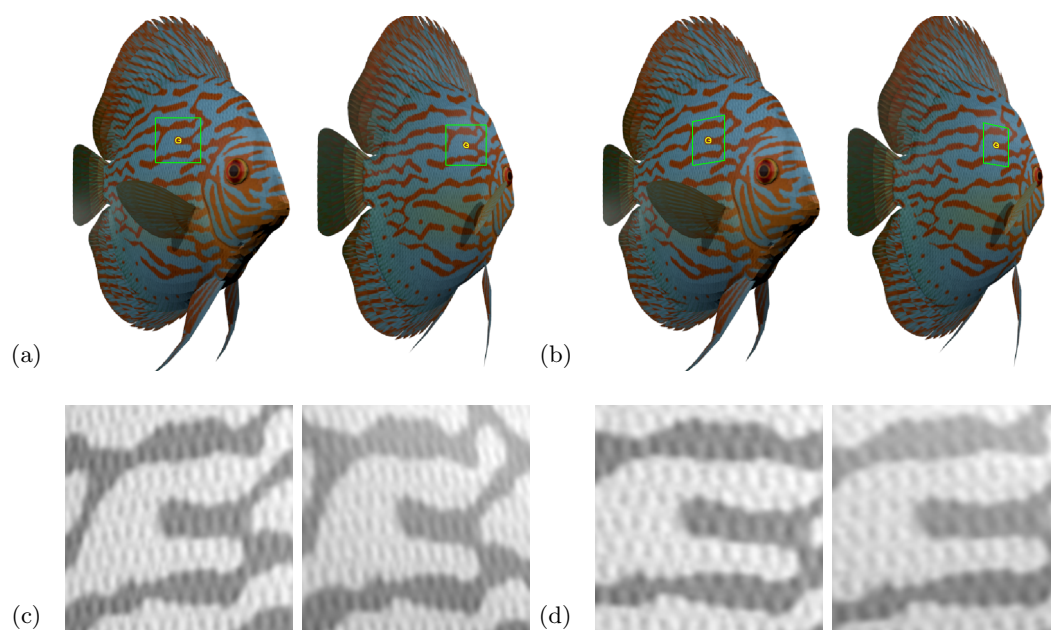


Figure 3.1 – Local patch normalization illustration. A keypoint is selected in two input images, and the corresponding local patches are shown. Standard (unnormalized) patches are displayed on Fig. (a) and (c). Patches normalized with the proposed approach are displayed on Fig. (b) and (d). Even though all the four patches are quite similar to each other, the latter two are mainly relied by an in-plane rotation, whereas the former ones – by a more complex transformation.

3.1 Overview

Local image features in traditional imaging demonstrate good stability under in-plane visual deformations, specifically *translations*, *rotations* and *scale changes*, both at the detection and the description stage. An exemplary baseline of local features fulfilling stability to these deformations classes is given by SIFT [22]. However, once visual deformations go out of the camera plane, and perspective distortions appear in the images, specific techniques are needed to render local features stable [22, 23].

The goal of this chapter is to investigate the potential outcome of involving depth maps into the feature extraction process in order to compensate the perspective distortions on the descriptor level. Depth map available within RGBD image allows to model the perspective in a vicinity of each detected salient visual point. Based on this key observation, we design a local normalization-based feature extraction algorithm, aimed at improved stability of local image features under out-of-plane rotations, and investigate repeatability and distinctiveness of resulting features.

Several different approaches have been proposed in the literature to address the problem of perspective distortions, as discussed in Sections 2.2.5 and 2.3.3. *Affine-invariant descriptors* [78] use the photometric information to recover a proper normalizing transformation. *ASIFT* [23] matches two given images trying to discover a set of dominant affine transformations relating them. This, however, does not allow to obtain a compact feature representation of an image, but only to match the two given images. *PIN* [101] and *DAFT* [102] use local normalization based on surface normals. *VIP* [82] uses a more global normalization approach based on planar decomposition of the scene.

We propose a new technique based on a similar principle: find distinctive locally planar landmarks and compensate perspective distortions before the descriptor is computed, by rendering frontal views of these landmarks. An illustration is given in Fig. 3.1. We highlight two important differences to the existing approaches that use similar planar normalization techniques [82, 101, 102].

- Our proposed approach is designed as an add-on to a conventional keypoint detector and a descriptor extractor, without any specific restrictions on them; in our experiments we test it with SIFT [22] and BRISK [46] features.
 - We perform an extended mid-level evaluation, comparing the proposed approach with the baseline given by texture-only features, including affine-invariant descriptors that also aim at correcting the perspective distortions but without using depth maps. We also involve into the comparison VIP features [82].
-

3.2 Proposed normalization approach

The proposed *slant normalization* approach is outlined in the following four steps (the items in bold are those affected by slant normalization, which employs the depth map):

1. Keypoint detection in the texture image;
2. **Local planar keypoint regions approximation and filtering of unstable keypoints;**
3. **Slant normalization of texture image patches;**
4. Descriptor computation.

Since the proposed technique performs independently of the used detector/descriptor pair, in the following we only discuss steps 2 and 3 in detail. The remaining steps, 1 and 4, are assumed provided within an existing local feature extraction framework, e.g. SIFT [22] or BRISK [46].

3.2.1 Estimation of local approximating planes and keypoint filtering

In order to perform slant normalization, the normal to the texture surface at each keypoint detected in the texture has to be estimated robustly. Clearly, this cannot be done on texture only and requires 3D information provided by depth. In [14], depth first-order derivatives are used to estimate surface normal vectors. However, this approach can be imprecise due to the fact that quantized depth maps are often piecewise constant. Moreover, differential characteristics might be more prone to noise. Instead, our approach is based on locally approximating the surface around the keypoint with a plane.

More formally, let $d(i, j)$ be the depth value in the pixel (i, j) , (i_0, j_0) the keypoint coordinates, S the keypoint area determined as a function of its scale. We approximate the depth map region corresponding to a keypoint area with the bilinear function $f_{A,B,C}(x, y) = A(x - i_0) + B(y - j_0) + C$. The normal vector of the plane, $\mathbf{n} = [A, B, C]$ is obtained by minimizing the average fitting error

$$F(A, B, C) = \sum_{(i,j) \in S} |f_{A,B,C}(i, j) - d(i, j)|^2, \quad (3.1)$$

which can be efficiently solved by least squares. Derivative approximation on range images through least squares is originally proposed and tested in [121], where a biquadratic surface, not planar, is locally fitted to the depth map at a given point.

The robust estimation of the normal vectors may be subject to estimation errors. Thus, we aim to detect those keypoints whose normal is likely to have been poorly estimated, and filter them out from the set of interest points in the texture image. First, we filter

keypoints based on the maximum plane fitting error $\rho = \max_S |f_{A,B,C}(i, j) - d(i, j)|$. We keep the keypoints that satisfy the condition:

$$\rho < T \min_S d(i, j). \quad (3.2)$$

It is convenient to avoid an absolute threshold in Eq. (3.2), since the dynamic range of the depth map may be arbitrary (depending on the unit value and the content). Instead, the ratio between ρ and minimum depth value in the keypoint area S does not depend on the dynamic range of the depth map. If this ratio is lower than T , the keypoint is accepted. Moreover, we consider the minimum depth value to take into account the effects of parallax changes according to the distance from the camera – even important viewpoint changes can be approximated by simple shift for background details, whereas near objects undergo more complex perspective transformations. The value of T is tuned experimentally:

- for blob detectors $T = 0.01$ achieves better performance in most cases, discarding non-planar areas,
- for corner detectors a higher value $T = 0.1$ performs better, since corners in texture are often surrounded by surface variations.

As a second filtering strategy, we reject surfaces with large slant angle, i.e., the angle between the normal and the optical axis of the camera. More precisely, we compute the slant angle as:

$$\theta = \arctan \sqrt{A^2 + B^2} \quad (3.3)$$

and reject keypoints with $\theta > 85^\circ$. The rationale is that such a surface, when viewed at large angles, might produce artifacts when sampling the normalized local descriptor patch.

3.2.2 Local surface sampling and slant normalization

For each geometrically-filtered texture image keypoint, we build a square regular sampling grid window on the approximating plane $Ax + By - z + C = 0$. More specifically:

- The **center point** of the window corresponds to the pixel (i_0, j_0) projected on the approximating plane.
- The window **size** is computed as a function of the initial keypoint scale σ , slant angle θ and the descriptor patch size, in such a way that the quadrilateral area obtained by the window boundary projection on the camera plane covers the keypoint area in the texture image.
- The **orientation** of the sampling window in the plane can be arbitrarily chosen, since the orientation of the texture keypoint has to be estimated *after* slant normalization.

As for the sampling window size, if $R(\sigma)$ represents the descriptor patch size projected on the approximating plane, we choose a square of side $M = 2R(\sigma)$ spatial units. In turn,

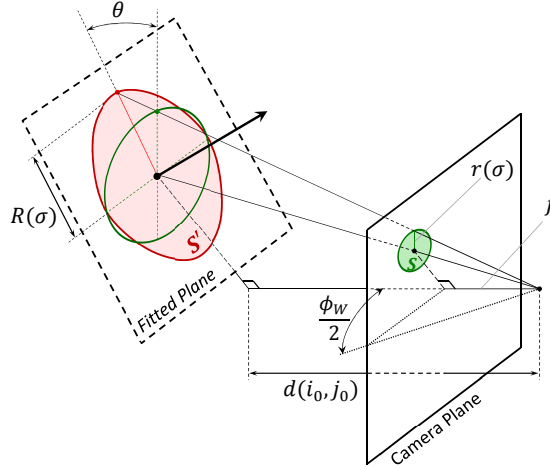


Figure 3.2 – Estimation of the sampling window size $R(\sigma)$ in the local approximating plane, obtained from descriptor patch size $r(\sigma)$. The corresponding keypoint area S' on the fitted plane is covered by a regular sampling grid which is then projected on the camera plane. The projected grid size is such that it covers the texture keypoint area S .

if $r(\sigma)$ is the descriptor patch size on the screen and f is camera focal length (both *in pixels*), it is straightforward to figure out using triangular similarity that

$$R \cos \theta : r(\sigma) = d(i_0, j_0) : f. \quad (3.4)$$

Having $\frac{W}{2f} = \tan \frac{\phi_W}{2}$, where W is image width in pixels, ϕ_W is horizontal angle of view of the camera, we get expression of $R(\sigma)$:

$$R(\sigma) = 2 \frac{r(\sigma) d(i_0, j_0)}{W \cos \theta} \tan \frac{\phi_W}{2}. \quad (3.5)$$

Finally, we compute a rectangular grid in the sampling window which is then projected from the local approximating plane to the camera plane. The grid points are distributed regularly in the window, i.e. with an equal step *in spatial units*. Then we apply the perspective projection model in order to compute grid points positions in pixels. This yields a warped, slant-invariant sampling grid used to sample a patch in the texture image, over which we can compute a local descriptor. Figure 3.2 illustrates how the window sampling is built in the approximating plane, and how the correct window size is found.

3.3 Experiments and discussion

We evaluate the proposed approach according to the procedure described in Section 2.5.1.

Two encapsulating local feature frameworks are involved in the experiments: *VLFeat* [80] implementation of SIFT [22] and the original implementation of BRISK [46]. Precisely, we compare the following methods:

- standard RGB-only SIFT and BRISK features;
- SIFT and BRISK features that undergo the proposed local planar normalization; the original keypoints, i.e., detected by the original detector in the texture image, are used;
- SIFT and BRISK features that undergo an iterative affine normalization, originally proposed for *Harris-Affine detector* [25], implemented within *VLFeat* library¹. This method represents a state-of-the-art alternative for RGB-only matching under significant perspective distortions and uses the original keypoints.
- We also involve into the comparison VIP features [82] that use depth maps to compensate for the perspective distortions (the original implementation is used).

All the input parameters are kept by default, except the detector sensibility threshold for BRISK: it is adjusted in such a way that the number of discovered features remains comparable to the one from SIFT. The original implementation of VIP does not allow to control this parameter, so in some cases VIP gives more features than the other methods.

As it is discussed in 2.5.1, for the ground truth in these experiments on the match labeling step (step **II**) we use the originally proposed ellipses reprojection between views. On the descriptor similarity score (step **IV**) we use Hamming distance for BRISK descriptors, 1st-to-2nd-closest ratio for the original SIFT descriptors, our normalized SIFT descriptors and VIP, and simple Euclidean distance for affinely normalized SIFT descriptors. In case of SIFT-based descriptors the choice is motivated by performance: we simply take the best performing distance function for each method independently of the others.

Four test sequences are used as the input data: *Arnold*, *Bricks*, *Fish* and *Graffiti*, giving 95 images in total. Examples of images and some characteristics of the dataset are presented in Section 2.5.2.1.

According to the evaluation procedure, we compute the matching score and the ROC curves.

3.3.1 Matching score test

Achieved matching scores are presented in Fig. 3.3. It could be noticed that features based on SIFT and BRISK keypoints demonstrate comparable overall performance in each group. In some cases our proposed method exhibits a slight gain (e.g., on *Fish* and *Arnold* sequences with SIFT features, on *Graffiti* with BRISK features), but may also lose up to 15 points (on *Bricks* sequence). Interestingly, on *Graffiti* sequence with binary BRISK features, our normalization achieves similar or better performance to affinely normalized SIFT, which is non-binary. However, affine-covariant descriptors are generally better than the unnormalized ones.

¹See VLFeat's `v1_covdet` function reference for details

VIP results deserve a dedicated discussion. First, since VIP looks for dominant planes, in the scenes with non-planar geometry it may fail: in the whole *Arnold* sequence VIP reveals unable to detect any feature. Second, as it uses RANSAC-based algorithm to discover the planes, its matching score becomes stochastic. This explains the leaps in matching score exhibited by VIP. Moreover, as a consequence, with a non-null probability VIP may fail to match a given image against itself. However, in overall VIP demonstrate good (often the best) matching score, especially for low angles between reference and test views. Thus, *Graffiti* sequence, consisting of a single plane, presents ideal conditions for VIP. On some sequences much lower scores are achieved for large angles. This could be explained by sampling artifacts in synthesized views of the discovered dominant planes.

3.3.2 Descriptor distinctiveness test

As it is explained in 2.5.1, the matching score reflects more the detector capabilities. However, our contribution concerns mainly the descriptor part, since the original keypoints are used in all the cases. For this reason, here we present and discuss mainly the ROC curves, allowing to analyze the descriptor performance in a more independent way from the used detector.

For a more thorough evaluation of the descriptor part, we collect the matching pairs from different images and split them into three groups based on the angular difference between their corresponding points of view: limited distortions (up to 30° of out-of-plane rotations), medium distortions ($30\text{--}60^\circ$) and large distortions (more than 60°). In each group we collect at least 40000 matches of each kind (true and false positive) and then plot the ROC curves. The matches from different scenes are put together providing a higher variability of the content. This results are displayed on Fig. 3.4.

We notice first, that the proposed depth-based normalization scheme allows to render the descriptors more distinctive, especially in the binary case. It can be stated for the whole spectrum of rotation angles, but especially for the large angles.

Our second conclusion concerns the affine normalization: the features resulting from this methods exhibit much less discriminability even compared to the standard features, both in the binary and non-binary case. This result is coherent with what VIP authors state [82]: affine-invariant features do not distinguish between a square, a parallelogram and a rectangle, or a circle and an ellipse, since these shapes are equivalent up to an in-plane affine transform. Such affine transforms allow to approximate well the perspective distortions, but form a transformation class in somewhat too large, and do not preserve the information that can be essential sometimes to distinguish the features. Prospectively, this causes a discriminability loss.

As for VIP features, they demonstrates high discriminability for low rotation angles, but loses quickly with the angle increasing. This is correlated with the lower matching scores, shown by VIP for large angles.

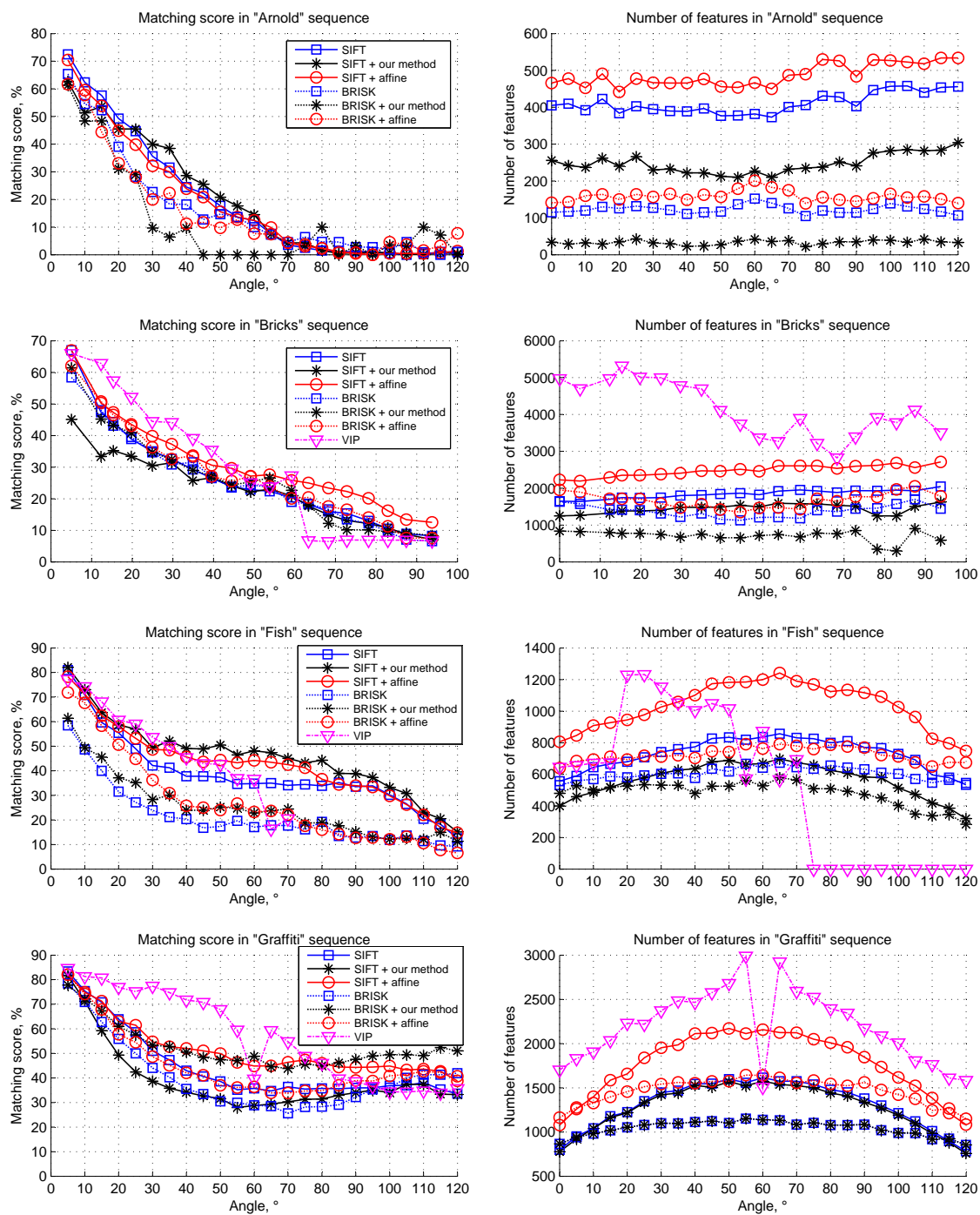


Figure 3.3 – Matching score (left column) and number of features (right column) on synthetic RGBD sequences *Arnold*, *Bricks*, *Fish* and *Graffiti*.

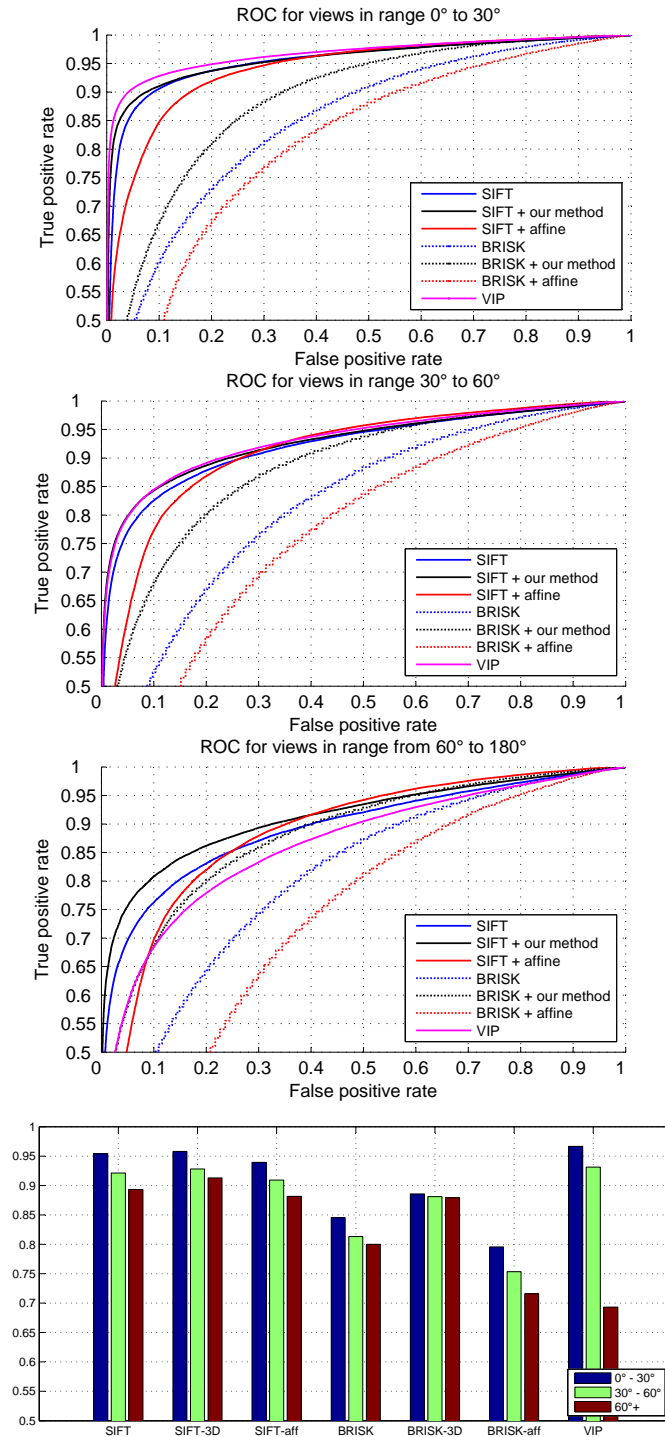


Figure 3.4 – Top row: ROC curves obtained on test data for three different angle ranges. Bottom row: corresponding areas under curves; “3D” refers to the proposed method, “aff” to the affine normalization.

3.4 Conclusion

In this chapter, we investigated the outcome of using depth maps to compensate perspective distortions on the descriptor extraction stage. We proposed a depth map-based local planar normalization technique, aiming at compensating perspective distortions in each descriptor patch before computing the descriptors. The proposed approach is tested within conventional SIFT and binary BRISK features and compared with affine normalization and VIP features. The results show a stable improvement of the descriptor discriminability in both binary and non-binary case over all the spectrum of out-of-plane rotations.

We also analyzed quantitatively at which extent the affine normalization that does not take into account the scene geometry may lead to less distinctive features, as it is claimed in [82].

As the matching score experiments show, the feature repeatability is not always improved by the proposed approach. The main reason consists in the fact that the depth maps are not involved into the keypoint detection. Consequently, one of the further steps will consist in involving the depth information on the keypoint detection stage, aiming at finding more repeatable keypoints.

Chapter 4

Binary RGBD descriptor based on a pattern projection

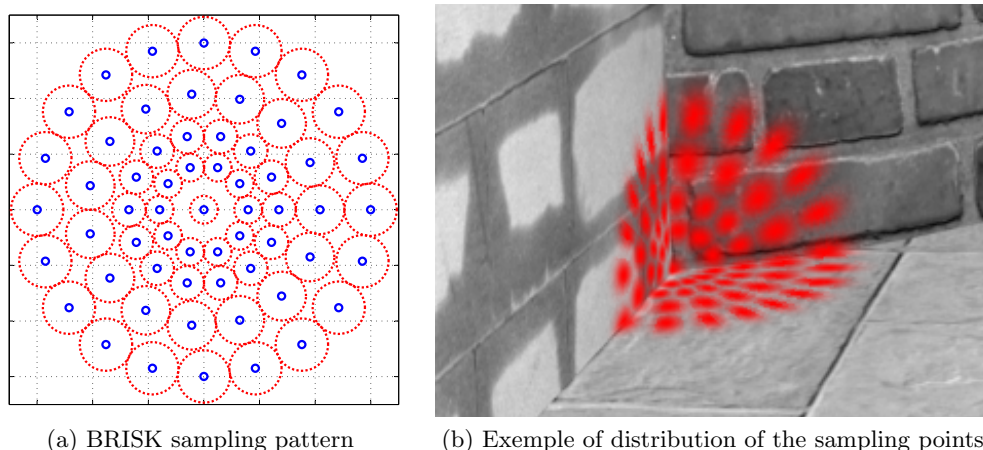


Figure 4.1 – Original BRISK sampling pattern for a keypoint of a unit scale, and an example of its distribution over the scene surface for a keypoint centered at a corner.

4.1 Overview

In this chapter, we continue to investigate the use of depth map for the descriptor normalization aimed at improving stability under significant viewpoint position changes. This time, we address both the repeatability of the features together with their distinctiveness, but again on the descriptor level.

The planar normalization is not always suitable. Smooth surfaces and complex geometrical forms require a higher order approximation. Some features, for example, issued from a corner detector, often fall near object boundaries, where the approximating plane is unlikely to fit well. In the normalization approach proposed previously in Chapter 3 such

features were rejected. They, however, can serve as representative landmarks in the content.

In this chapter we are addressing this limitation. With the initial idea of normalizing the descriptor support through a higher order approximation we immediately encounter a problem: generally high order smooth surfaces can not be mapped to a plane *isometrically*, i.e., the normalizing transformation will have an impact on gradient distribution of the texture image. This renders inapplicable a SIFT-like gradient histogram-based descriptor, since the resulting signature becomes dependent on the normalizing transformation, which violates the basic idea of the normalization.

Our key observation in this point consists in the fact that the isometric constraint on the normalization mapping may be omitted if the descriptor is not based on gradient statistics. A conventional example of suitable descriptors is given by *binary descriptors* that require to sample the image only in few points, i.e., no “continuous” image patch per keypoint is assumed. In this case, the isometric mapping is not required if the sampling points may be distributed over the surface in *some* stable way. An illustration is given in Fig. 4.1. After the image is sampled in these properly distributed points, the original binary descriptor computation technique may be used, i.e., a binary string is formed through pairwise comparisons of obtained samples. A descriptor-to-surface mapping that does not impose any special assumptions on the surface allows to avoid (or at least significantly reduce) the rejection of keypoints, which should improve the *repeatability*. Moreover, it makes the description intrinsic to the surface texture, which should result in a better *distinctiveness*. The features are then expected to become more robust to viewpoint position changes and, potentially, some more complex surface deformations.

More precisely, the goal of this chapter is to design such a descriptor-to-surface mapping and to test it within a binary descriptor in order to render binary features stable to viewpoint position changes. According to the comparison of different binary features in [21], BRISK [46] demonstrate better overall results. For this reason we use BRISK feature pipeline as the baseline to develop the proposed surface texture descriptor. In the following we present details on the descriptor computation, as well as results of evaluation of its distinctiveness and matching performance, mainly comparing to the conventional BRISK features.

4.2 Proposed descriptor pattern projection

4.2.1 Mapping descriptor pattern to scene surface

As any other conventional feature extraction algorithm, BRISK consists of two separated stages: keypoint detection and descriptor computation. In this chapter, similarly to the previous one, we use the original keypoint detection algorithm, and step in only on the description stage. Once the keypoints are detected, we first compute the local polar parametrization at each keypoint as it is explained below, i.e. we look for *radial* and *angular*

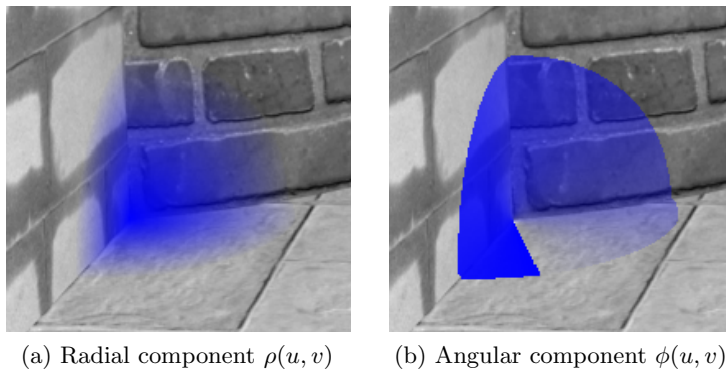


Figure 4.2 – Illustration of pattern parametrization components for the keypoint on Fig. 4.1.

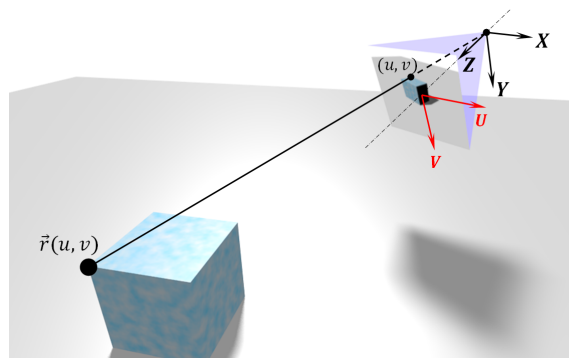


Figure 4.3 – Image surface parametrization in local camera coordinates.

coordinate of each pixel, that are intrinsic to the scene surface.

Local parametrization: radial component

Let $I(u, v)$ denote the intensity image, $D(u, v)$ the depth map, H and W their heights and widths in pixels, and ω the horizontal angle of view of the camera and the depth sensor. Using the pinhole camera model, we set up the following global parametrization of the scene (see Fig. 4.3):

$$r(u, v) = \begin{pmatrix} 2u \tan \frac{\omega}{2} \\ 2v \frac{H}{W} \tan \frac{\omega}{2} \\ 1 \end{pmatrix} D(u, v). \quad (4.1)$$

This global parametrization will be used to compute the desired local polar parametrization at each keypoint.

For a given keypoint centered at (u_0, v_0) , we first compute the distances $\rho(u, v)$ from (u_0, v_0) to other pixels applying the fast marching algorithm [122], allowing to compute efficiently a map of geodesic distances from a given point of a surface to other points. Fast marching is a family of numerical methods solving the Eikonal equation $\|\nabla u\| = F$ in one sweep, i.e., by simulating a front propagation through the image starting from a given

source point. This technique is perfectly adapted to our needs, as we do not have to process the whole image but the keypoint neighborhood only.

The fast marching is started at the keypoint center and stopped when a certain limiting distance, corresponding to the keypoint scale, is reached (this distance is further referred to as *geodesic keypoint scale* σ_g). The "keypoint area" may thereby be defined as $M = \{(u, v) : \rho(u, v) < \sigma_g\}$. The resulting geodesic distances to the keypoint center are intrinsic to the scene and do not depend on the viewpoint position. Thus, the image resulting from the fast marching algorithm gives us directly the radial component of the parametrization we are looking for.

Geodesic keypoint scale σ_g , that limits the fast marching process, may be seen as the characteristic keypoint area size expressed in scene spatial units. It is related to the sphere radius that surrounds the keypoint area, expressed in these units. For a keypoint of scale σ , the corresponding radius is given by the following formula derived from the pinhole camera model:

$$R = \sigma D(u_0, v_0) \frac{2 \tan \frac{\omega}{2}}{W} \quad (4.2)$$

In our tests we set σ_g equal to $6R$. This determines the scaling of the sampling pattern in function of the keypoint scale. This value is set experimentally and is reasonable in comparison to the patch extents of other descriptors; larger extent will require more time to compute the descriptor, where smaller values cause distinctiveness losses.

Local parametrization: angular component

The estimation of the angular component is more difficult. Differently to the polar geodesic parametrization in [123], we limit ourselves to an approximation, that is reasonable due to the locality and using the depth map but not an arbitrary mesh.

In a nutshell, we approximate the angular coordinate of a given point in M using precomputed values from a set of points forming a closed curve around the keypoint center. So, we first extract a level curve on the geodesic distance map $\rho(u, v)$, i.e. an oriented closed contour $C = \{(u, v) : \rho(u, v) \approx a\sigma_g\} = \{C_i\}_{i=1}^n$, where $a < 1$ is a constant. At the same time, we compute the spatial length of C by summing up the spatial distances between neighboring points. During this summation, we keep the array of cumulated lengths

$$L_k = \sum_{i=1}^k \|r(C_i) - r(C_{i+1})\|, k = 1, \dots, n. \quad (4.3)$$

By normalizing L_k to the interval $[0, 2\pi)$ we get the "angles" ϕ_k of points of the curve C . The angular coordinate of any other point of M is then estimated by selecting that of the point in C minimizing the angle $\alpha(\vec{x}, \vec{y}) = \arccos\left(\frac{(\vec{x}, \vec{y})}{\|\vec{x}\| \|\vec{y}\|}\right)$ between corresponding two

vectors from keypoint center:

$$i^* = \arg \min_i \alpha(r(u, v) - r(u_0, v_0), r(C_i) - r(u_0, v_0)) \quad (4.4)$$

$$\phi(u, v) = \phi_{i^*} \quad (4.5)$$

In our tests, we used $a = 0.8$, so that the reference curve C cuts the keypoint area M in two roughly equal parts in terms of number of points.

The two computed components $\rho(u, v)$ and $\phi(u, v)$, that form the local surface parametrization, are illustrated in Fig. 4.2a and Fig. 4.2b.

4.2.2 Descriptor computation

Following the BRISK architecture, we now need to smooth the image locally at each sampling point. Working in polar coordinates, we propose a "polar Gaussian kernel", a naive extension of the classic bi-dimensional Gaussian kernel to the polar coordinates, i.e. a function providing square-exponential decreasing, but in radial and angular sense. To give its analytic formulation, let us study the sampling pattern in Fig. 4.1a in more details.

This pattern may be split radially into 5 layers. The first layer consists of the center point only, each following layer contains a set of points with a constant radius and equally spaced angles. Let us take a layer l having n_l points, and a point number k . Let r_l be the layer radius and s_l the associated layer scale (i.e. scale of sampling points on that layer). We define the smoothing kernel corresponding to the selected point of the layer as follows.

$$K_{l,k}(\rho, \phi) = \exp \left(-\frac{(\rho - r_l)^2}{2s_l^2} - \frac{\left(\text{mod}\left(\phi - \frac{k-1}{n_l} \frac{n_l r_l}{4\pi}\right)\right)^2}{2s_l^2} \right) \quad (4.6)$$

We denote by mod a function that wraps the angle in radians (i.e. modulo 2π).

The kernels for different (l, k) values are illustrated in Fig. 4.1b.

The response at the selected sample point is then given by

$$S_{l,k} = \frac{\sum_M K_{l,k}(\rho(u, v), \phi(u, v)) I(u, v)}{\sum_M K_{l,k}(\rho(u, v), \phi(u, v))}. \quad (4.7)$$

$S_{l,k}$ for all l and k gives us the required sample values. To end up with a binary descriptor, we then proceed in a very similar way to BRISK. In a nutshell,

- we estimate the characteristic pattern direction using the long-distance sample pairs exactly as it is done in BRISK,
 - we shift the angular component $\phi(u, v)$ in such a way that the characteristic direction becomes zero,
 - we sample the image again with shifted $\phi(u, v)$,
-

- we compute the descriptor using the short-distance sample pairs (comparing the intensity values).

Long and short-distance pairs (sample indexes in the sampling pattern) do not depend on the content and are precomputed as in the original BRISK algorithm.

4.3 Experiments

To test the proposed approach we use the evaluation procedure and the dataset described in Section 2.5.1. For the keypoint match labeling criteria, we associate to each keypoint a spherical area and compute the volumetric overlap of spheres instead of the planar overlap of ellipses. The radius of the spheres is computed by Eq. (4.2). This allows to compute the overlap analytically without sampling the keypoint areas, and to reduce an eventual influence of the sampling issues on the resulting score.

Our method is compared to the original BRISK descriptor [46] and binary RGBD descriptor BRAND [100]. Authors implementation is used. For completeness, we also add to the comparison SIFT descriptor implemented in VLFeat library [80]. All the descriptors are tested with the original BRISK detector, even though in practice SIFT typically uses its own detector.

We present obtained matching score in Fig. 4.4 and ROC in Fig. 4.5. To complete the results we compute areas under ROC curves (AUC), splitting the matches in three groups for limited (up to 30°), moderate (30° – 60°) and large (more than 60°) viewpoint angle changes to evaluate the descriptors stability for different ranges of out-of-plane rotations. These results are presented in Fig 4.6.

The geometrical correction performed in our descriptor allows to match the image patches viewed under a large spectrum of angles of view, that nor BRISK neither SIFT descriptor could achieve. This is confirmed by higher matching scores, as shown in Fig. 4.4, especially for high-detailed texture (*Graffiti*). BRAND also shows a high matching score except for *Graffiti* sequence, where the geometry is not representative enough. Our descriptor outperforms the other methods in terms of ROC and AUC as presented in Fig. 4.5 and Fig. 4.6. The limited performance of SIFT descriptor is mainly explained by using the unadapted corner BRISK detector instead of the original blob detector.

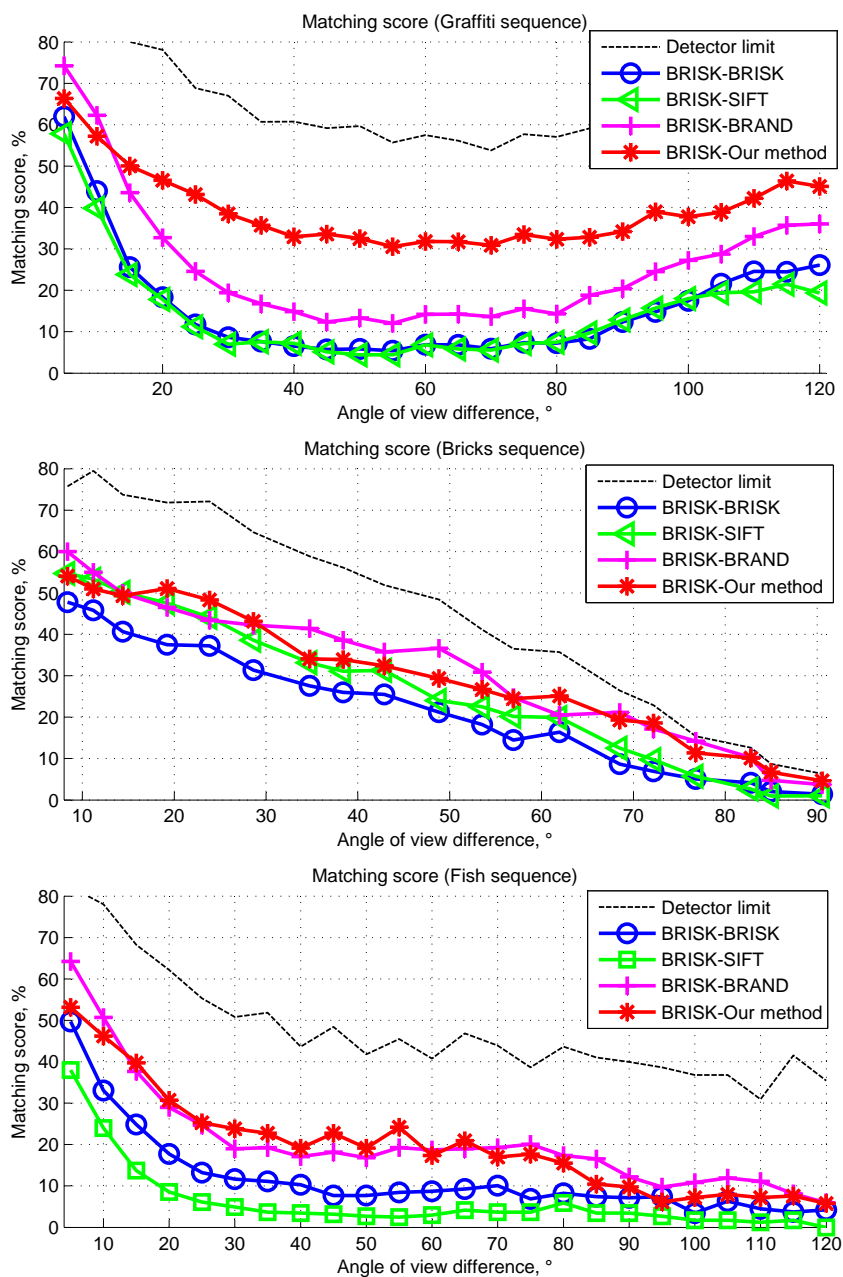


Figure 4.4 – Matching scores obtained on different sequences with original BRISK detector, SIFT descriptor and the proposed descriptor. The original BRISK keypoints are used with all the descriptors. Black curves represent detector limitation, i.e. numbers of repeated keypoints (*repeatability*).

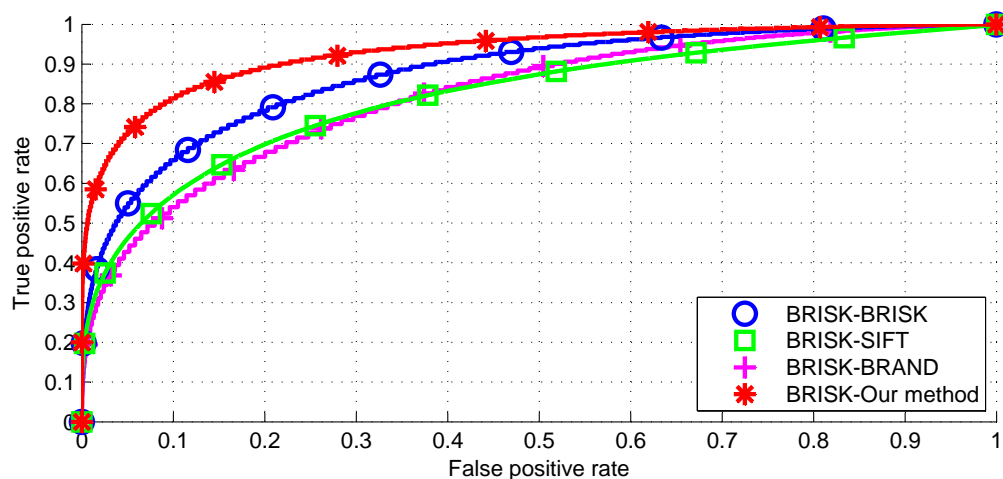


Figure 4.5 – Receiver operating characteristics obtained on the entire dataset with original BRISK detector, SIFT descriptor and the proposed descriptor. The original BRISK keypoints are used with all the descriptors.

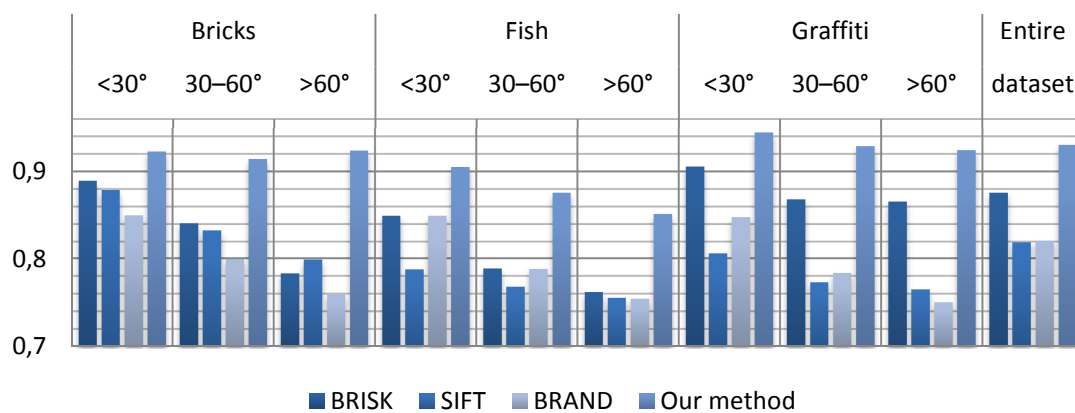


Figure 4.6 – Areas under ROC curves obtained on test sequences for different ranges of out-of-plane rotations, and on the entire dataset (corresponding curves are presented in Fig. 4.5).

4.4 Conclusion

In this chapter we proposed a binary descriptor pattern mapping technique that allows to render repeatable and distinctive binary features of the surface texture. Since the descriptor is not extracted in the camera plane, but sampled from the surface, it makes the description intrinsic to the content and less dependent on the camera position within the scene, which is confirmed by the experimental results.

By design the proposed technique is not limited to rigid scene deformations only. As the descriptor pattern mapping to the scene surface is based on the geodesic distance, the descriptor covers *non-rigid isometric transformations* too (referred to as **G-VI** in Table 2.1). However, this is difficult to evaluate due to test data specificity and uncommonness of this deformation class in practice. Such an evaluation might be considered as a part of the future work.

The projection algorithm does not depend on the selected descriptor sampling pattern. BRISK pattern that we used might be easily replaced by any other binary pattern, designed manually or resulted from a learning process such as [41], [73] or [75]. It is sufficient to parametrize the pattern in polar coordinates, then the descriptor might be sampled from the surface texture through the proposed local parametrization. Learning an appropriate pattern for the surface description from data is promising to improve distinctiveness, and is another point of future work.

The use of geodesic distance in the pattern parametrization requires a non-negligible computational effort. This limits the application area of the proposed descriptor, since it requires more time than conventional binary features like BRISK or even than SIFT. For this reason, a faster pattern-to-surface mapping algorithm, possible approximative, is of interest.

Chapter 5

TRISK: A local features extraction framework for RGBD content matching

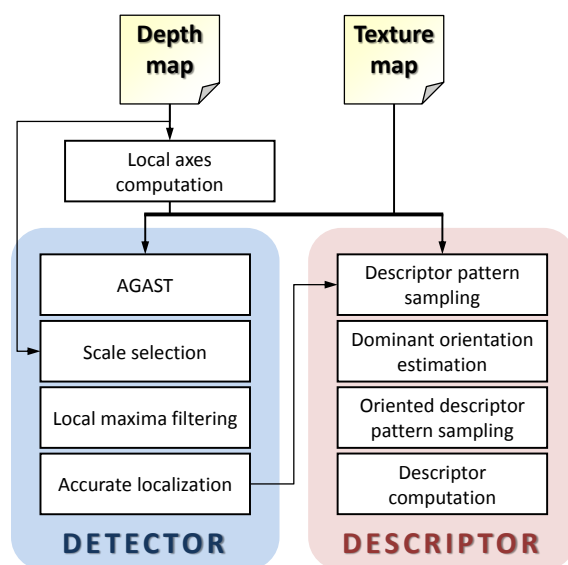


Figure 5.1 – The proposed TRISK pipeline architecture.

5.1 Overview

The local features extraction process includes two main stages: detection and description. So far, we were leveraging the depth map to render stable local features only on the description stage. In this chapter we design an entire extraction pipeline, addressing not only the description, but the detection stage as well.

More specifically, our goal is to obtain reliable features under significant viewpoint

position changes, which take into account depth map imperfections and do not require a high computational effort. The ingredients of the proposed RGBD features extraction pipeline are some of the tools that have been successfully used for features extraction on conventional texture images, adapted to exploit the scene geometry inferred from depth maps. Specifically:

- The keypoint detector is based on non-local maxima suppression of AGAST-based corner score [45]. In our method the score is computed in **local surface axes** deduced from the depth map instead of the ordinary image axes.
- The scale invariance of the proposed features is ensured in a very efficient way using a **depth-based scale selection** technique similarly to [100].
- We reemploy the accurate keypoint localization procedure of SIFT [22], adapted to the local surface axes.
- Local surface axes used on the detection stage also allow for fast **local plane normalization**.
- The dominant orientation of the mapped descriptor pattern is estimated in the same way as in BRISK [46].

As our proposed method is inspired by BRISK features, hereafter it is referred to as **TRISK** for “Tridimensional Rotational Invariant Surface Keypoints”. The detailed presentation of the method follows.

5.2 Proposed approach

The overall scheme of TRISK is shown in Fig. 5.1. This Section describes in details the building blocks of the proposed detector and descriptor.

5.2.1 The Detector

The proposed feature extraction algorithm begins with the following steps.

Local surface axes computation

Our main goal is to render the feature extraction process independent as much as possible of the camera position, by transferring all processing operations from the camera plane to the scene surface with a reasonable computational cost. In TRISK this is done by selecting a proper basis at each image point, which we further refer to as *local axes*. Assuming that keypoint detection and description are rotationally invariant, it is straightforward to show that *any* two orthonormal vectors in the tangent plane at a given point of the surface may serve as such a basis. Examples are shown in Fig. 5.2.

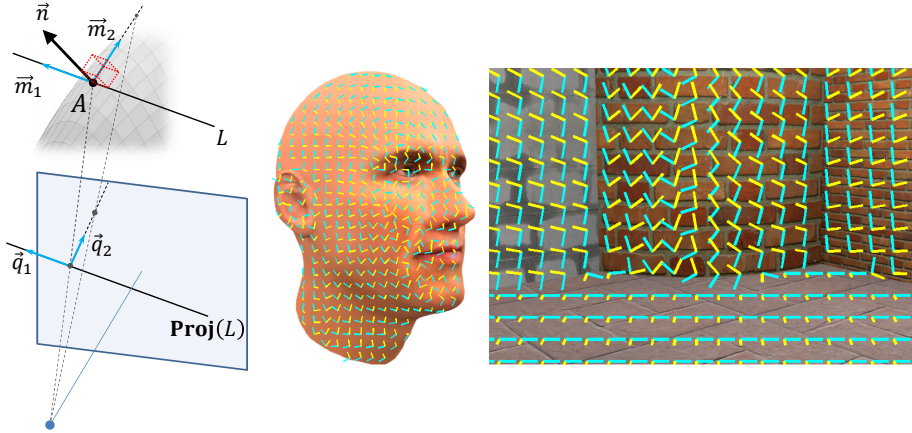


Figure 5.2 – Computing of local axes \vec{q}_1 and \vec{q}_2 using surface normal \vec{n} (left) and examples of local axes fields on images from *Arnold* and *Bricks* sequences (right). \vec{q}_1 is shown in cyan, \vec{q}_2 in yellow. This figure is best viewed in color.

Thus, deriving the adaptive local axes from the depth map is at the base of TRISK. The following local operations will then be performed in the derived local axes: AGAST corner score computation, Harris cornerness test, accurate keypoint localization and descriptor pattern sampling. We first explain the proposed technique to compute the local axes and then present the details of their use in the feature extraction.

Let us consider a camera with the centered principal point. According to the perspective projection model, the relation between a spatial point (x, y, z) and its projection point $(u, v) = \mathbf{Proj}(x, y, z)$ on the camera plane is then expressed by the following formula (the corresponding coordinate systems are presented in Fig. 4.3):

$$\begin{aligned} u &= \frac{x}{z} \\ v &= \frac{y}{z} \end{aligned} \quad (5.1)$$

Let A denote a scene point, \vec{A} its coordinate vector in the camera coordinates, and $(u, v) = \mathbf{Proj}(\vec{A})$. Let $\vec{n} = (n_x, n_y, n_z)$ be the surface normal of unit norm at A (cf. Fig 5.2). With no generality loss we assume $0 < n_z < 1$.

The following reasoning is based on the observation that the degree of perspective distortions along a contour on the scene surface passing through A depends on its direction with respect to the camera plane. Specifically, a tangent line L parallel to the camera plane is not affected by the perspective distortions: there is no contraction along L when projecting it on the camera plane. Nothing prevents to use this line as the first local axis. Thus, let us take a vector $\vec{m}_1 = (-n_y, n_x, 0)$ and its projection on the camera plane $\vec{q}_1 = \mathbf{Proj}(\vec{A} + \vec{m}_1) - \mathbf{Proj}(\vec{A})$. It is straightforward to show that \vec{m}_1 is a directional vector of line L . As there is no contraction along L , the corresponding basis vector is of

unit norm: $\vec{q}_1 = \|\vec{q}'_1\|^{-1}\vec{q}'_1$. This gives

$$\vec{q}_1 = \frac{1}{\sqrt{n_x^2 + n_y^2}} \begin{pmatrix} -n_y \\ n_x \end{pmatrix}. \quad (5.2)$$

The second required spatial vector \vec{m}_2 must be orthogonal to both \vec{n} and \vec{m}_1 , as together with \vec{m}_1 it forms an orthogonal basis on the surface. This can be found by the cross product: $\vec{m}_2 = \vec{m}_1 \times \vec{n}$. We estimate the degree of perspective distortions along the second axis by the cosine of its angle to the viewpoint vector \vec{A} . This gives the norm of the second projected vector, i.e., $\vec{q}_2 = \frac{(\vec{m}_2 \cdot \vec{A})}{\|\vec{m}_2\| \|\vec{A}\|} (\mathbf{Proj}(\vec{A} + \vec{m}_2) - \mathbf{Proj}(\vec{A}))$. Doing the algebra, we end up with the following expression:

$$\vec{q}'_2 = \begin{pmatrix} \frac{n_x n_z - u}{\sqrt{n_x^2 + n_y^2 - 1}} - u \\ \frac{n_y n_z - v}{\sqrt{n_x^2 + n_y^2 - 1}} - v \end{pmatrix}, \quad \vec{q}_2 = \frac{n_x u + n_y v - n_z}{\|\vec{q}'_2\| \sqrt{u^2 + v^2 + 1}} \vec{q}'_2. \quad (5.3)$$

The derived expressions of \vec{q}_1 and \vec{q}_2 depend only on the surface normal and the point position on the camera plane (u, v) , but not on the depth map values directly. To estimate the normal vector we use PCA-based normal estimation [124]. Using this approach the local axes field may be computed in $O(N)$ operations for an input image of N pixels. Moreover, it avoids explicit manipulations with differential characteristics of the depth map, which are sensible to noise.

The described technique allows to compute the adaptive local axes from the depth map in a computationally efficient way and robustly to the noise.

AGAST and scale selection

Adaptive Generic Accelerated Segment Test [45] is an approach for corner detection in images. According to this test, to be a corner, a pixel must be darker or brighter than at least N connected pixels on the surrounding circle. The pixel score is then defined as the lowest pixel value still fulfilling this condition. Pixels whose score reaches a local maximum are taken as keypoint candidates. This detection principle was first used in SUSAN (*the Smallest Univalve Segment Assimilating Nucleus*) corner detector [43], and was then successfully involved in scale-covariant keypoint detection [41, 46]. Due to its isotropic (rotational invariant) and derivative-free design, this detection principle demonstrates good stability to image noise and moderate geometric deformation. In our case, isotropic detection is required for using local adaptive axes. Moreover, AGAST allows to save time by reducing the number of intensity comparisons using a properly learned decision tree. This also responds well to our needs, since the intensity interpolation in the local adaptive axes is time consuming.

Specifically, inspired by BRISK detector [46], we apply AGAST to pick the keypoint

candidates as follows:

- Aiming at improved stability to viewpoint position changes, we apply AGAST9-16 in the local surface axes (“9-16” stands for at least 9 darker or brighter pixels on a circle of 16 pixels). The texture map is interpolated using the local surface axes defined in Eq. (5.3). Precisely, for a Bresenham’s circle $\{(u_k, v_k)\}_{k=1}^{16}$ and local axes $\vec{q}_1 = (\xi_1, \eta_1)$ and $\vec{q}_2 = (\xi_2, \eta_2)$ at a given image point, we sample the texture map at locations

$$(x_k, y_k) = (u_k\xi_1 + v_k\xi_2, u_k\eta_1 + v_k\eta_2), k = 1, \dots, 16. \quad (5.4)$$

The corner test is then performed on the obtained samples. Depending on the content, some of these samples might be unnecessary for the corner test: AGAST allows to sample only those locations that are required to derive the pixel score.

The idea of performing such a test is illustrated in Fig. 5.3. Non-local maxima suppression is then applied on the generated score map in order to select the keypoint candidates.

- For improved stability to significant scale changes we run AGAST test on each level of a multiscale image pyramid. The pyramid consists of the original image and its subsampled versions (*octaves*); each next level is halfsampled with respect to the previous level. After the keypoint is detected on a given level, it is kept only if its AGAST score is greater than AGAST scores in the same position in adjacent level. Differently to the original BRISK, the pyramid we use is sparse, i.e., there is only one level per octave. This is mainly motivated by the fact that we do not use the pyramid to derive the keypoint scale, but need it only to avoid missing keypoints when the image scale changes significantly. The proposed scale selection method in TRISK is explained below.
- A typical corner revealed by AGAST is an intersection of two straight contours or a point-like structure. We believe that the characteristic size of such a structure (its *visual scale*) is difficult to define properly: local patches of slightly different sizes centered around such a corner are visually similar, contrarily, for example, to a blob-like structure which exhibits more clearly such a characteristic size. However, scale estimation accuracy has a major impact on repeatability. For this reason, we use AGAST response only to derive the keypoint position but not its scale, since in case of RGBD images a better clue of scale is available in the depth map. To achieve scale invariance, we employ *geometrical* scale. Namely, we get the keypoint scale from the depth map assuming that the underlying visual detail is of a fixed spatial size σ_0 . The resulting scale is simply equal to $\sigma = \frac{\sigma_0}{z}$, where z is the average depth of the keypoint. To avoid scale estimation errors for keypoints situated near depth boundaries, we estimate z iteratively, at the same time when the keypoint dominant

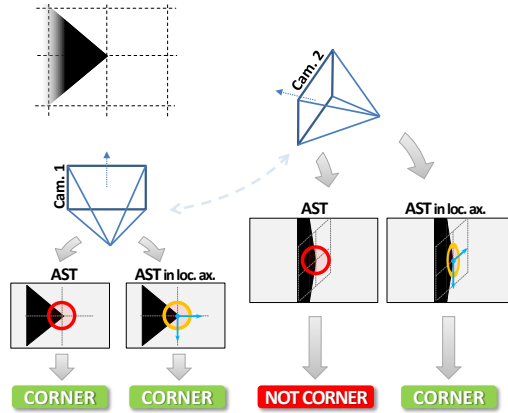


Figure 5.3 – Illustration of application of Accelerated Segment Test (AST) in standard image axis versus local axes derived from the depth map. A corner viewed under a large angle projects itself at a nearly straight contour on the camera plane, so that the corner test in standard image axes fails causing a repeatability loss.

orientation is selected. This is explained below. The desired scale invariance with this technique is then achieved in a very simple way: the farther the surface is, the smaller the keypoint is. A similar idea is used in [100]. The keypoint area is finally described by an ellipse spanning the scaled local axes $\sigma\vec{q}_1$ and $\sigma\vec{q}_2$. Thus, TRISK keypoints are not circular as those of SIFT or BRISK, but elliptical similarly to the keypoints produced by affine-covariant detectors [116].

Local maxima filtering

The initial keypoint candidates given by local maxima of AGAST score are then analyzed subject to their stability. A well-known supplementary criterion to filter out unstable keypoint candidates is based on Harris cornerness measure [39]. It was first used in SIFT and then reemployed in other detectors, e.g. ORB [41]. Some keypoints reported by a corner detector may actually be situated on straight edges, for example due to aliasing artifacts. These keypoints are prone to localization errors. In order to filter them out, the eigenvalue ratio of Hessian matrix H is thresholded [22]:

$$H = \begin{pmatrix} I_{xx} & I_{xy} \\ I_{xy} & I_{yy} \end{pmatrix}. \quad (5.5)$$

Here I denotes the smoothed texture image.

In our approach, differently to the presented classic technique, we replace the standard derivatives of I by the directional derivatives computed in the adaptive local axes \vec{q}_1 and \vec{q}_2 , i.e., we deal with the eigenvalues of

$$H_q = \begin{pmatrix} I_{\vec{q}_1\vec{q}_1} & I_{\vec{q}_1\vec{q}_2} \\ I_{\vec{q}_1\vec{q}_2} & I_{\vec{q}_2\vec{q}_2} \end{pmatrix}. \quad (5.6)$$

The reason is always the same: changing the axes allows to reduce the impact of perspective distortions when dealing with the texture curvature. We compute the eigenvalue ratio in the same way as in SIFT, and use the same threshold value: a keypoint is rejected if the ratio is greater than 10 [22].

Accurate localization

On the last stage of the detection process, we perform an accurate localization of the remaining keypoint candidates. This allows to localize accurately the keypoints detected on subsampled versions of the input image and also serves as an additional criterion of keypoint stability: not all the keypoint candidates may be precisely localized, and the ones that reveal unstable behavior during the accurate localization are rejected.

We reemploy the interpolation technique used in SIFT and SURF and initially presented in [60], based on the Taylor expansion of the score function up to the quadratic terms. We apply it to the AGAST score reducing the number of dimensions from three to two, as no scale dimension is considered in our case, and in the adaptive local axes instead of the standard ones.

More precisely, let S be the AGAST score, (x, y) a candidate point, (x^*, y^*) an accurately localized local maximum, and $Q = (\vec{q}_1 \ \vec{q}_2)$ the coordinate transformation. We first express S in the local coordinates:

$$\tilde{S}(\xi, \eta) = S \left(Q \begin{pmatrix} \xi \\ \eta \end{pmatrix} - \begin{pmatrix} x \\ y \end{pmatrix} \right) \quad (5.7)$$

We develop the Taylor expansion of $\tilde{S}(\xi^*, \eta^*)$ where $(\xi^* \ \eta^*)^T = \vec{\delta} = Q^{-1} \begin{pmatrix} x^* - x \\ y^* - y \end{pmatrix}$ with respect to the local coordinate center:

$$\tilde{S}(\xi^*, \eta^*) \approx \tilde{S} + \begin{pmatrix} \tilde{S}_\xi & \tilde{S}_\eta \end{pmatrix} \vec{\delta} + \frac{1}{2} \vec{\delta}^T \begin{pmatrix} \tilde{S}_{\xi\xi} & \tilde{S}_{\xi\eta} \\ \tilde{S}_{\xi\eta} & \tilde{S}_{\eta\eta} \end{pmatrix} \vec{\delta}. \quad (5.8)$$

\tilde{S} and its derivatives on the right side of the equation above are taken at point $(0, 0)$. Deriving this and using the fact that (ξ^*, η^*) is a local maximum, i.e. $\tilde{S}_\xi|_{\xi^*, \eta^*} = \tilde{S}_\eta|_{\xi^*, \eta^*} = 0$, we obtain:

$$\vec{\delta} = - \begin{pmatrix} \tilde{S}_\xi \\ \tilde{S}_\eta \end{pmatrix} \begin{pmatrix} \tilde{S}_{\xi\xi} & \tilde{S}_{\xi\eta} \\ \tilde{S}_{\xi\eta} & \tilde{S}_{\eta\eta} \end{pmatrix}^{-1}. \quad (5.9)$$

The displacement in standard image axes is equal to $Q(\vec{\delta})$.

Similarly to the known SIFT implementation [80] we apply this process iteratively, cumulating the offset and reinterpolating the derivatives of \tilde{S} . For a better selection of stable keypoints, we reject a keypoint during the iterations if the Hessian of \tilde{S} is rank-deficient. Following [80], in our implementation we perform at most 5 iterations.

5.2.2 The Descriptor

Once the set of interesting point positions and scales is provided, a compact description is computed for each point.

In Chapter 4, we studied how binary features may be used to extract a surface-intrinsic information from RGBD images in order to provide a description robust to rigid 3D deformations. A descriptor sampling pattern was projected on the scene surface, providing a depth-based descriptor normalization procedure aimed at producing invariant features. However, such a projection is (1) very sensitive to depth map noise and (2) requires a high computational effort. To be robust to the viewpoint position changes on the descriptor level, in this work we propose a simpler approach based on a similar concept: the descriptor normalization is performed according to a local plane that approximates the scene geometry nearby the keypoint.

Non-binary local planar normalization-based descriptors are studied in the literature [82, 101, 102]. In this work we apply this principle to produce a binary descriptor. Precisely, we reuse the BRISK descriptor sampling pattern, applying it to the image in adaptive local axes computed at the keypoint that immediately gives us the approximating local plane. The pattern used in the original BRISK implementation and an example of how it is mapped onto the scene using local axes at a given corner point is shown in Fig. 5.4. We notice that our design is not restricted to the BRISK sampling pattern; any other manually designed or appropriately learned pattern might be used with no additional cost.

In TRISK we proceed as follows. Let $\{(\xi_k, \eta_k)\}_{k=1}^M$ represent the Cartesian coordinate pairs of the descriptor sampling pattern points. In case of BRISK, $M = 60$. As discussed in Section 4.2.2, (ξ_k, η_k) values may be easily derived analytically thanks to the radially regular disposition of the pattern points.

For a given keypoint position (X, Y) and scale σ , we reuse the local axes \vec{q}_1 and \vec{q}_2 in order to map the pattern points to the image plane:

$$\begin{pmatrix} x_k \\ y_k \end{pmatrix} = \sigma \xi_k \vec{q}_1 + \sigma \eta_k \vec{q}_2 + \begin{pmatrix} X \\ Y \end{pmatrix} \quad (5.10)$$

The original BRISK uses a two-pass scheme that consists in sampling the pattern, computing its dominant orientation from obtained samples and sampling the oriented version of the pattern (by a “pass” we mean sampling the pattern). In TRISK we proceed similarly. However, the descriptor pattern in our case is more sensitive to keypoint parameter estimation errors due to (a) perspective warping introduced by the local axes, (b) depth map imperfections and (c) scale errors for keypoints situated near object boundaries, where the depth varies abruptly. The latter is crucial since we average depth to derive the geometric scale of the keypoint as explained above. For this reason, we propose the following three-pass scheme estimating both dominant orientation and scale in an accurate way.

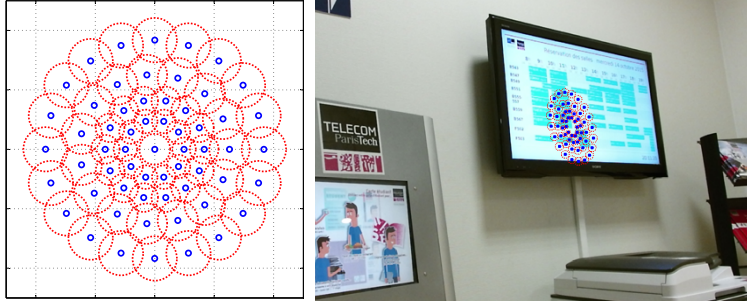


Figure 5.4 – BRISK descriptor sampling pattern from the original implementation (left) and its mapping to the surface through local planar normalization (right).

We begin with the geometric scale $\sigma = \frac{\sigma_0}{z}$, where z is an average depth value in the keypoint center. This provides a rough initial estimate of the scale which is further refined.

1. The pattern is sampled in locations (x_k, y_k) : averaged image intensity is computed at each point. The neighborhood radius per point is taken as shown in Fig. 5.4 and scaled by σ . The pattern is sampled both from texture and depth maps, producing two sets of smoothed intensity and depth values P_I and P_D respectively.
2. Descriptor dominant orientation Θ is computed using the BRISK methodology from P_I ; the depth value z used in the initial estimate of the scale is recomputed as average of all the values of P_D .
3. The unmapped pattern (ξ_k, η_k) is reoriented according to Θ : each point is simply turned around the pattern center by $-\Theta$ radians. The oriented pattern is mapped on the plane and sampled again using the updated value of σ .
4. Dominant orientation Θ and scale σ are re-estimated once again in the same way as in step 2, producing final values Θ^* and σ^* .
5. The pattern is sampled again according to Θ^* and σ^* , giving final P_I and P_S sets.
6. Control scale value σ_c is computed as before; the keypoint is kept only if σ_c differs from σ^* by no more than 1% of the latter, i.e., if the scale error is negligible.
7. Finally sampled P_I values undergo pairwise intensity comparison tests to produce a binary string forming the descriptor.

5.2.3 Implementation details

TRISK has several parameters that control different stages of the feature extraction process. For most of them we use the same values as in the original BRISK or SIFT papers or their implementations [22, 46, 80]. Other parameters, such as 1%-error threshold in the scale

estimation, are derived from experiments and do not impact significantly the performance. All these values are mentioned in the text.

The remaining parameters are (1) neighborhood size κ for PCA-based normal estimation used when computing the adaptive local axes, (2) AGAST score threshold t and (3) basic scale σ_0 used in the scale selection. These parameters impact the matching performance and their optimal values may depend on the content. We learn appropriate values for these main parameters based on the matching performance using a procedure described in Section 5.3.3.

For all the texture smoothing and interpolation operations we use the image filter presented in Chapter 7.

The depth map values are used for normal estimation and scale selection. In both cases, they are not used directly, but a neighborhood of each pixel is considered. This allows to cope with the noise and small “holes” (areas with no depth). Larger “holes” are simply skipped (e.g. no keypoint detection is performed in these areas).

5.3 Experiments

In this section, we evaluate the proposed method compared to several well-known local visual features in two scenarios:

- a mid-level feature evaluation in terms of matching score and receiver operating characteristics (ROC) as explained in Section 2.5.1;
- a visual odometry experiment on three sequences of *Freiburg* dataset [117].

In what follows we provide a detailed description of the experiments and present the results.

5.3.1 Compared methods

The following local feature extraction methods are used in the experiments.

- The baseline is given by the BRISK features [46]. Publicly available original implementation is used.
 - BRAND descriptor [100] is a recent approach for RGBD content matching. We use it in conjunction with STAR detector as proposed in the original paper. This method is referred to as STAR-BRAND. STAR is an OpenCV implementation of the Center Surround Extremas (*CenSurE*) [113]. The original implementation of the descriptor is used.
 - VIP [82] is based on SIFT descriptors computed on RGBD images and aimed at improved viewpoint invariance. We use publicly available authors implementation.
-

Method	Keypoint type	Descriptor type and size	Depth map usage
TRISK	Corner	Binary 512 bit	detector and descriptor
BRISK	Corner	Binary 512 bit	no
STAR-BRAND	Blob	Binary 512 bit	descriptor
VIP	Blob	Numeric 128 dim.	preprocessing
AFFINE	Blob	Numeric 128 dim.	no
SIFT	Blob	Numeric 128 dim.	no

Table 5.1 – Summary of compared methods.

- As we deal with out-of-plane rotations, we compare the proposed method to an affine-covariant detector [116] initialized with SIFT keypoints and referred to as AFFINE. *VLFeat* [80] implementation is used.¹
- For completeness, standard SIFT features [22] are also involved in the evaluation (*VLFeat* implementation is used).

Hence we have six approaches being compared. Table 5.1 summarizes some characteristics of the compared methods.

5.3.2 Matching score and ROC

We first perform mid-level evaluation according to the description given in Section 2.5.1, as also used in the preceding chapters. We use three sequences from the synthetic RGBD dataset described in Section 2.5.2.1 and three sequenced of RGBD images acquired with Kinect from *Freiburg* dataset [117], containing more complex camera position changes: sequences *desk* (we used 40 frames with 10 frames skipping) and *structure_texture_far* (59 frames with 5 frames skipping) represent out-of-plane rotations, whereas in *floor* sequence (19 frames with 5 frames skipping) the camera moves arbitrarily within the scene.

The resulting matching score and ROC curves obtained on the test sequences are presented in Fig. 5.5 and 5.6. The numbers of features detected by each method are presented in a compact way in Table 5.2.

It can be seen from the results, that in all the test sequences TRISK demonstrates improved overall matching score. In some cases (*Graffiti*, *House*, *Floor*) TRISK also shows the slowest decay, which indicates improved feature stability under viewpoint position changes. The second best matching score on synthetic sequences (top row in Fig. 5.5) is arguably achieved by VIP. Based on a planar normalization technique, VIP performs well in case of simple geometry, i.e., when the scene surface is mostly planar or very smooth, otherwise it may even be unable to detect any features. TRISK also exploits the principle of planar normalization, but in a much more local way, which allows it to perform reasonably

¹See `v1_covdet` function reference for more details.

well not only on simple surfaces but also in scenes with more complex geometry, such as *desk* and *House*.

With few exceptions, the remaining four approaches that do not use depth information on the detection stage globally show comparable matching scores.

As for the descriptor discriminability examined with ROC curves (bottom rows in Fig. 5.5 and 5.6), the best performance is shared among TRISK, VIP and sometimes SIFT. TRISK outperforms the other approaches on sequences with simple geometry and detailed texture (*Graffiti* and *structure_texture_far*), but in other cases turns out to be comparable to or moderately less distinctive than non-binary descriptors, notably SIFT and VIP. This result is consistent for the following two reasons.

First, non-binary descriptors used in the tests are represented by 128-dimensional numeric vectors. They are naturally more distinctive than 512-bit binary descriptors since they carry more information. This is coherent to other evaluations in the literature [19, 34, 46]. It is also worth noticing that the other binary competitors are mostly always significantly outperformed by TRISK.

Second, gain in terms of ROC is not always practically meaningful. In the *House* sequence VIP demonstrates the best discriminability but low matching scores. Corresponding matching score graph shows that only the first 9 images are reliably matched against the reference. Matching features from the remaining images have a limited impact on the ROC curve since they are few, i.e., ROC mainly depends on the first few images. However, the first images of any sequence are visually more similar to its reference image than its remaining part. Consequently, the matched descriptors from these images are less deformed and their true and false positive matches are more easily distinguishable by the inter-descriptor difference. This leads to a gain in terms of ROC, which is actually spurious, since the most challenging part of the sequence is mostly unmatched. Thus, a ROC gain is only practically meaningful if the matching score is reasonably high over the whole deformation spectrum. Even though to a lesser extent, the other sequences exhibit a similar phenomenon. This renders questionable the observed moderate ROC gains of non-binary features over TRISK.

5.3.3 Parameter values estimation

We use matching score and ROC to analyze the impact of input parameters of TRISK on its performance and find empirically optimal values.

Specifically, we collected 500 image pairs from *large_with_loop* and *long_office_household* Freiburg sequences each. These two sequences represent different kinds of viewpoint position changes (from out-of-plane rotations in *long_office_household* to scale changes and 3D translations in *large_with_loop*). The three main input parameters of TRISK are treated as follows: we took 6 values of neighborhood size κ used in the normal estimation, 6 values of basic scale σ_0 and 5 values of AGAST score threshold t . This gives in total $6 \times 6 \times 5$

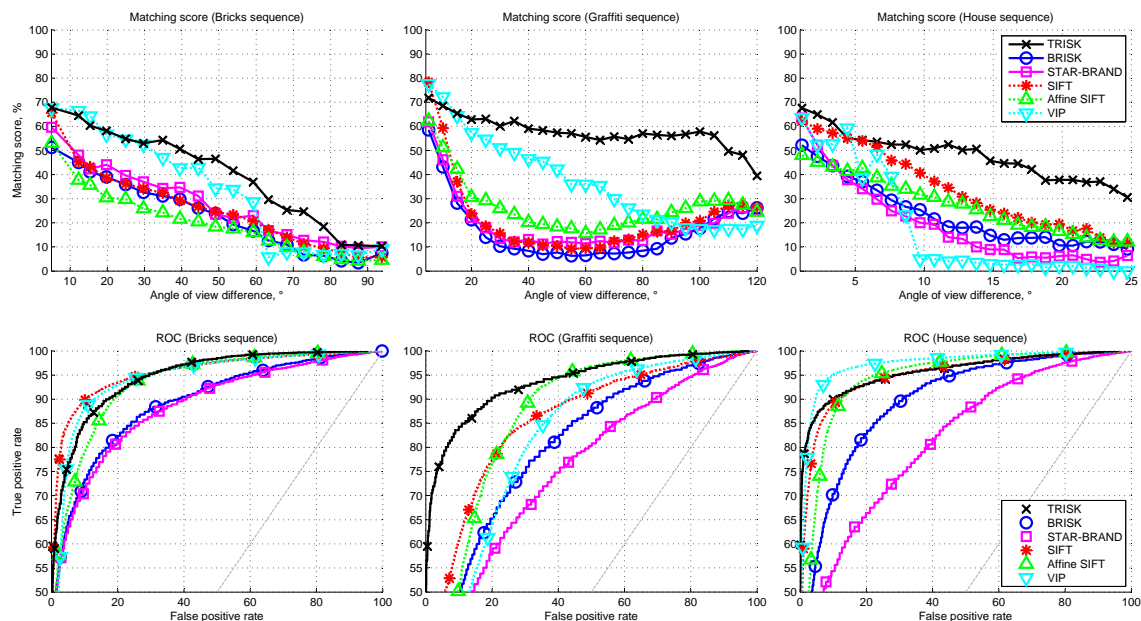


Figure 5.5 – Matching score and receiver operating characteristics demonstrating repeatability and distinctiveness of the compared detectors and descriptors, mainly under out-of-plane rotations (*Bricks* and *Floor* sequences) and scale changes (*House* sequence). Computed on synthetic RGB data. At least 4800 true positive and 4800 false positive matches were selected to plot each ROC curve.

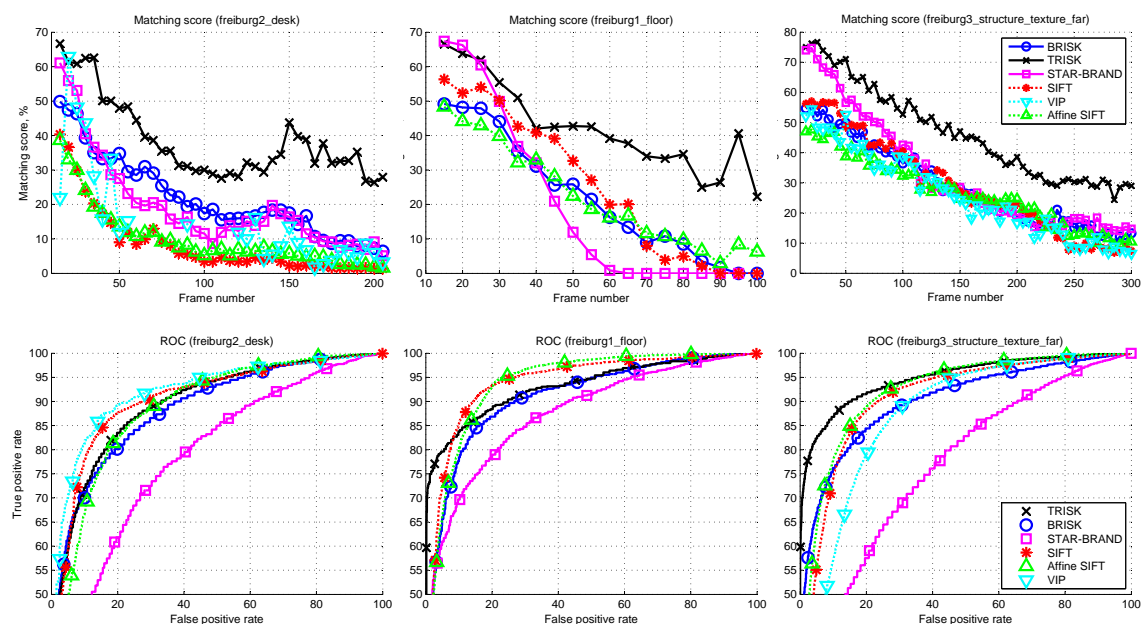


Figure 5.6 – Matching score and receiver operating characteristics demonstrating repeatability and distinctiveness of the compared detectors and descriptors under viewpoint position changes of different kind. Computed on three sequences of *Freiburg* dataset, acquired with Kinect. In some images in *desk* sequence and in the whole *floor* sequence VIP turns unable to detect any feature.

Sequence		TRISK	BRISK	BRAND	SIFT	AFFINE	VIP
<i>Bricks</i>	MIN	314	766	1072	1638	2194	3346
	AVG	954	915	1188	1841	2482	4293
	MAX	1247	1163	1330	2047	2714	5458
<i>Graffiti</i>	MIN	695	855	595	782	1079	1603
	AVG	1181	1041	809	1305	1764	2280
	MAX	1432	1151	917	1615	2171	3029
<i>House</i>	MIN	339	1069	462	1924	2445	237
	AVG	1233	1364	889	2235	3056	1831
	MAX	2171	1604	1240	2637	3609	3503
<i>desk</i>	MIN	427	436	433	898	1115	0
	AVG	742	881	524	1036	1343	113
	MAX	1129	1375	611	1213	1597	420
<i>floor</i>	MIN	694	686	522	1176	1565	–
	AVG	895	1089	833	1290	1692	–
	MAX	1145	1410	1045	1460	1895	–
<i>structure_texture_far</i>	MIN	705	479	509	1060	1461	672
	AVG	1178	964	692	1154	1615	976
	MAX	1557	1393	838	1298	1820	1220

Table 5.2 – Minimal, average and maximal number of features extracted from each scene. Minimum and maximum values per row are highlighted in green and yellow.

triples $(\kappa_i, \sigma_{0i}, t_i)$, that cover a spectrum of reasonable values for the input parameters. We matched then all the selected image pairs using each parameter triple. This provided us with about 20 millions matching pairs of features in total. As a function \mathcal{F} to maximize, we choose the product of averaged matching score over all the image pairs and area under ROC curve, which seems a reasonable joint performance index of detector and descriptor.

We notice that the AGAST score threshold t in our proposed algorithm plays the same role as in BRISK, has a major impact on the number of detected features and strongly depends on the content. If its value is too small, generally a lot of features are detected, but a small part of them is matched. If its value is too large, few features are detected, but they are very stable and lead to a high matching score. However, variations of t within a reasonable range do not impact a lot the matching score nor ROC. A similar detection score threshold is also present in other detectors. For this reason, for all the approaches we use, we tune this parameter in such a way that the numbers of detected features in each used dataset remain comparable among the methods (see Table 5.2). However, not all the implementations allow to tune this parameter, e.g., the original implementation of VIP.

Based on this reasoning, we averaged \mathcal{F} over 5 parameters of t , reducing the search space to two dimensions (κ, σ_0) . This finally gives us 6×6 parameters sets, having from 70K to 180K matching feature pairs per each set, where \mathcal{F} exhibits a distinctive maximum near point $(\kappa, \sigma_0) = (25, 14.27)$. Taking into account that these values might also be device- and content-dependent (mainly on a representative characteristic size of captured visual details

and the depth maps measurement unit), we consider them as default ones for Kinect depth maps given in meters and indoor office-like environments. These values are used in all the experiments in this work, including the ones on the synthetic RGBD data, whose depth maps were scaled to fit Kinect statistics.

5.3.4 Visual odometry

Finally, we evaluate TRISK in a visual odometry scenario using two Kinect and ASUS Xtion image sequences from *Freiburg* dataset [117].

The goal consists in retrieving camera pose evolution relatively to an initial pose using only the acquired images. The ground truth pose is recorded with a motion capture system and is provided within the dataset. We follow the setting of [100]: to compute the camera transformation (translation and rotation) between two frames, we match them, apply random sample consensus (RANSAC) to filter putative matches and, finally, run the Iterative Closest Point algorithm [125] retrieving the relative translation vector and rotation matrix. The resulting pose is recovered by cumulating deduced translations and rotations. In this experiment we limit the number of keypoints extracted from each image by each detector, keeping at most 1000 keypoints with the highest response. In case of TRISK, the detector response is the interpolated AGAST score.

Two types of errors are used in the evaluation:

- *translation error*: the distance between estimated and ground truth positions,
- *rotation error*: $\varepsilon = \arccos \frac{\text{tr}(R^{-1}R_{gt}) - 1}{2}$, where R is the estimated camera orientation matrix with respect to the initial pose, and R_{gt} is the ground truth one.

Typically, each registered frame is matched against the next one, providing a “delta-pose” that is added to the current position. In our experiment, we proceed differently: we skip more than one frame, i.e., we look for the transformation relating frame 0 to frame $K > 1$, then frame K to frame $2K$, etc. This technique has a twofold effect. On one hand, it allows to compensate the visual drift being cumulated with each new “delta”, as well as to reduce the computational time. On the other hand, the resulting errors depend strongly on the features quality (matching capabilities and localization accuracy), as the visual difference between frames n and $n + K$ is typically more significant than the one between n and $n + 1$. This setting is thus a good scenario to evaluate the features.

Translation and rotation errors evolution on different sequences is presented in Fig. 5.7, 5.8 and 5.9. To compensate for the stochasticity induced by RANSAC, we run the experiment 10 times on each sequence averaging the results.

All the methods have similar error values in the first frames. However, as the scene evolves, the drift cumulates differently for different features.

It can be observed that TRISK generally achieves smaller errors. An exception is *floor* sequence (Fig. 5.8), where all the methods achieve small errors compared to other sequences

(less than 12 cm and 5°), but AGAST-based features turn out to be slightly less precise in rotations. The possible reason is that in this sequence the camera moves quickly (for this reason we set $K = 5$ for this sequence and not 10 as for the others). This causes a noticeable directional blur in texture maps, which interferes with corner detection but is manageable by blob detectors. A drastic difference in the odometry precision is revealed on *desk* sequence (Fig. 5.7), where mostly all the other approaches, notably BRISK, experience severe errors in matching consecutive frames. TRISK is the only approach providing precision within 10 cm and 4° . Finally, on *structure_texture_far* sequence (Fig. 5.9), TRISK is mainly competing with VIP, which also perform well thanks to the locally planar geometry. It is worth noticing that on the other two sequences VIP proves unable to provide enough matches for continuous trajectory estimation.

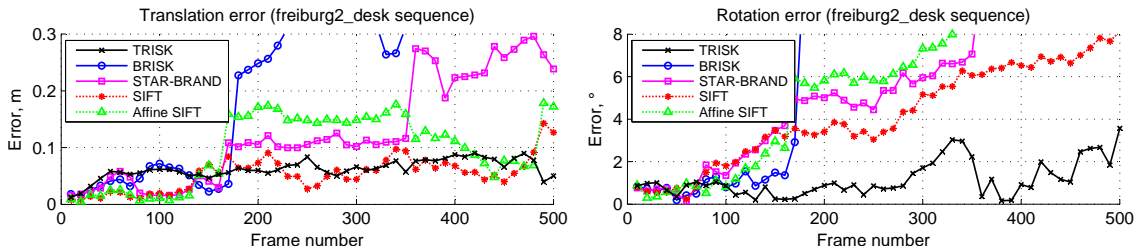


Figure 5.7 – Visual odometry with 10 frames skipping on *freiburg2_desk* sequence (first 500 frames): translation (left) and rotation (right) errors. VIP fails on this sequence, thus it is not reported.

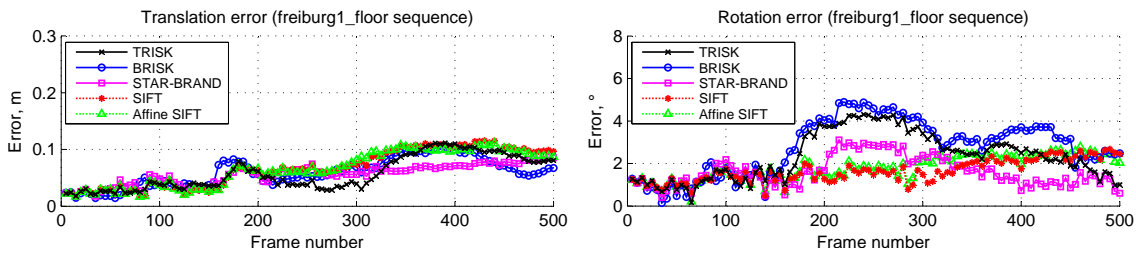


Figure 5.8 – Visual odometry with 5 frames skipping on *freiburg1_floor* sequence (first 500 frames): translation (left) and rotation (right) errors. VIP fails on this sequence, thus it is not reported.

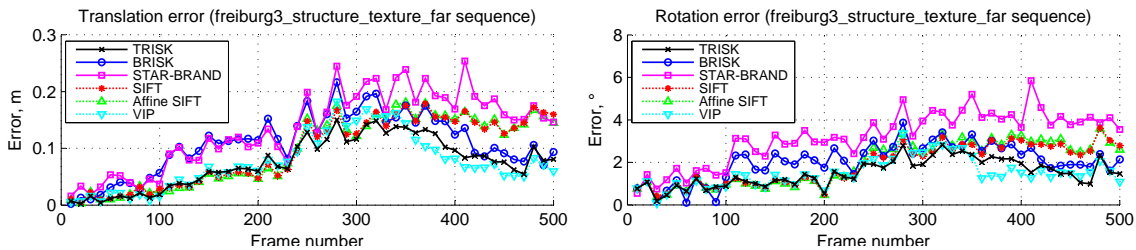


Figure 5.9 – Visual odometry with 10 frames skipping on *freiburg3_structure_texture_far* sequence (first 500 frames): translation (left) and rotation (right) errors.

5.3.5 Note on computational efficiency

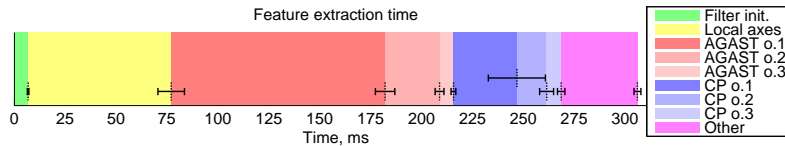


Figure 5.10 – Feature extraction time averaged over images from matching score test (Fig. 5.6). Smoothing filter initialization, local axes computation, AGAST over 3 octaves, keypoint candidates processing (“CP”) over 3 octaves (includes accurate localization, Harris corner test and descriptor computation) and remaining processing times and their standard deviations are displayed.

We also evaluate the computational efficiency of TRISK: the time spent on each stage of feature extraction from real RGBD images from *Freiburg* dataset is presented in Fig. 5.10.

Being invoked from MATLAB environment through MATLAB MEX interface², our C++ TRISK implementation takes **306 ms** per VGA image on average over about 150 images, with 21.2 ms standard deviation. The most time consuming steps are the local axes computation and AGAST on the first octave. The description time is included in the keypoint candidates processing on each octave, and thus is much lower than the detection time. It can also be observed that the highest time deviation occurs during the first octave candidates processing (about 14 ms), since this stage is the most content-dependent.

It is worth noticing that the local adaptive axes might be computed differently, e.g., PCA-based normal estimation technique [124] may also provide two orthogonal vectors to the normal that might be used as the local axes. This, however, requires the complete PCA decomposition of the point cloud covariance matrix at each pixel. We tested this approach and obtained very similar performance, but the average local axes computation time increased by 60 ms.

For real applications TRISK can be speeded up considerably by using multiple threads. The adaptive local axes computation, AGAST and local maxima suppression are purely local, and all the keypoint candidates are processed independently starting from the accurate localization to the descriptor computation. This makes TRISK easily parallelizable, allowing for distributed and GPU-based implementations.

5.4 Conclusion

In this chapter we presented a complete pipeline of local feature extraction for texture+depth image matching. The proposed TRISK features target application scenarios where significant viewpoint position changes are present in the input data. The experiments showed that TRISK improves consistently both feature stability and distinctiveness, which allows for better performance on the application level. TRISK can be applied on real RGBD images

²Tested on a 64-bit Windows machine with a 3.5 GHz 6-physical core CPU and 16 Gb of RAM.

acquired with low-cost RGB-depth camera pair, such as Microsoft Kinect or ASUS Xtion, without any complex preprocessing of the depth map. The computational effort required to process an image is sufficiently low, so that TRISK is able to perform at near-realtime rates.

The major limitation of TRISK consists in its limited ability to deal with complex geometry. If the observed scene contains fine detailed shapes, TRISK may not perform well. The main reason is in the local planar normalization used to compute the descriptor. A more complex way to render the descriptor stable and invariant to viewpoint position changes, such as the technique presented in Chapter 4, is more computationally expensive and sensible to the depth map imperfections. Rendering the descriptor robust to geometrically complex scenes with a reasonable computational cost is one of the objectives for future.

Chapter 6

Keypoint detection in RGBD images based on a viewpoint-covariant scale space

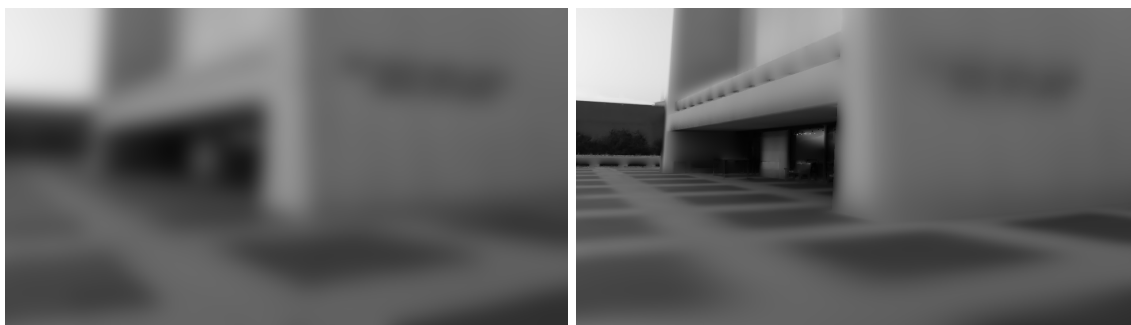


Figure 6.1 – Visual comparison of the uniform Gaussian smoothing (left) and the proposed non-uniform smoothing (right) of the same input image. The latter is propagated “along the surface”: farther objects are less smoothed. An accurate formulation of this principle, referred to as *viewpoint covariant behavior*, leads to a causal multiscale representation (scale space) allowing for repeatable keypoint detection.

6.1 Overview

Repeatable keypoint detection is a performance bottleneck of local features. A weak detector performance cannot be compensated on subsequent stages of the feature extraction process: even with a highly distinctive descriptor the robust matching is feasible only if the detected keypoints remain stable under content deformations.

In this chapter, we deal with the detection of repeatable RGBD keypoints, making use of both texture and depth information. Differently to our previous considerations, here we interpret the problem of keypoint stability under viewpoint position changes into the covariance of underlying keypoint detection modality, *scale space*. Specifically, in this

chapter we first design a scale space for the texture map, that exhibits an (approximative) *viewpoint-covariant behavior*, i.e., it changes accordingly when the camera moves (an illustration is given on Fig. 6.1; we detail this concept further in the text). Second, we build a blob keypoint detector for texture+depth images that searches for the keypoints in the designed scale space and analyze its repeatability in different scenarios.

Our proposed scale space is based on a non-linear diffusion process, established for the texture map, but controlled by the depth map. Together with the viewpoint covariance, allowing for repeatable keypoint detection under viewpoint position changes, the proposed scale space has several important properties.

- It respects the causality axiom, which is crucial for a multiscale image representation used in keypoint detection.
- The numerical stability of the underlying iterative process is proved.
- It is based on a smoothing filter that is linear as a function of the texture map.
- It respects depth boundaries, i.e., no smoothing propagates across object borders.

We also propose an efficient implementation of the designed scale space on GPU, using widely available OpenGL framework.

6.2 Design of RGBD scale space

As described in Section 2.4, a scale space might be defined through a PDE problem with the input image taken as the initial data. Such a problem defines a diffusion process, in which the time is interpreted as the image scale. The variability of such a definition is contained in the right side of the differential equation, which determines how the smoothing propagates in image space with time.

Therefore, to design the new scale space, we pose such a PDE problem, first defining a Laplacian-like operator for RGBD content that enables to establish the desired diffusion process.

6.2.1 Laplacian operator definition

Let the input image be of size $W \times H$ pixels, so that $\Omega = \left[-\frac{W}{2}, \frac{W}{2} - 1\right] \times \left[-\frac{H}{2}, \frac{H}{2} - 1\right]$ denotes the image support. In what follows, spatial image variables taking values from Ω are referred to as u and v . We denote by $D : \Omega \rightarrow \mathbb{R}^+$ the depth map associated to the image I being processed. We assume known the horizontal angle of view ω of the camera.

It can be easily shown using the pinhole camera model, that the function $\vec{r} : \Omega \rightarrow \mathbb{R}^3$ defined below parametrizes the image surface in local camera coordinates as illustrated in

Fig. 4.3:

$$\vec{r}(u, v) = \begin{pmatrix} 2u \tan \frac{\omega}{2} \\ 2v \frac{H}{W} \tan \frac{\omega}{2} \\ 1 \end{pmatrix} D(u, v). \quad (6.1)$$

Let us now proceed to a discrete image support Ω_d obtained by sampling Ω with step h in both dimensions. For a function f defined on the continuous support Ω , we introduce the following differential quantities, which are similar to the notion of directional derivatives in [12]:

$$\partial_u f = \frac{f(u+h, v) - f(u-h, v)}{\|\vec{r}(u+h, v) - \vec{r}(u-h, v)\|} = \frac{f(u+h, v) - f(u-h, v)}{r_u^{+-}} \quad (6.2)$$

$$\partial_v f = \frac{f(u, v+h) - f(u, v-h)}{\|\vec{r}(u, v+h) - \vec{r}(u, v-h)\|} = \frac{f(u, v+h) - f(u, v-h)}{r_v^{+-}} \quad (6.3)$$

where r_u^{+-} and r_v^{+-} are introduced in order to simplify notation. Applying twice this operator yields second-order differential quantities, e.g., $\partial_{uu}f = \partial_u(\partial_u f)$. For a better operator kernel locality, we also introduce a definition through one-sided finite differences as follows:

$$\partial_{u+}f = \frac{f(u+h, v) - f(u, v)}{\|\vec{r}(u+h, v) - \vec{r}(u, v)\|} = \frac{f(u+h, v) - f(u, v)}{r_u^+}, \quad (6.4)$$

$$\partial_{u-}f = \frac{f(u, v) - f(u-h, v)}{\|\vec{r}(u-h, v) - \vec{r}(u, v)\|} = \frac{f(u, v) - f(u-h, v)}{r_u^-}, \quad (6.5)$$

$$\begin{aligned} \partial_{uu}f &= \frac{\partial_{u+}f - \partial_{u-}f}{r_u^{+-}} \\ &= \frac{f(u+h, v)}{r_u^+ r_u^{+-}} - \frac{f(u, v)}{r_u^+ r_u^{+-}} - \frac{f(u, v)}{r_u^- r_u^{+-}} + \frac{f(u-h, v)}{r_u^- r_u^{+-}}. \end{aligned} \quad (6.6)$$

$\partial_{v+}f$, $\partial_{v-}f$ and $\partial_{vv}f$ are defined similarly.

Finally, we define a Laplacian-like second order differential operator summing up the second-order differential quantities defined above:

$$L \equiv \partial_{uu} + \partial_{vv}. \quad (6.7)$$

6.2.2 PDE problem formulation

Next, we set up a partial differential equation problem that describes the diffusion process with the proposed Laplacian operator (6.7):

$$\begin{cases} \frac{\partial f}{\partial t} = Lf \\ f|_{t=0} = f_0. \end{cases} \quad (6.8)$$

This problem is very similar to the classic diffusion problem (2.11). To study this similarity and set up some useful properties, let us return back to the continuous definition

domain. We obtain a continuous generalization of the differential quantities (6.2) and (6.6) by letting h tend towards zero, that is:

$$\begin{aligned}\mathcal{D}_u f &= f_u \|\vec{r}_u\|^{-1} \\ \mathcal{D}_{uu} f &= f_{uu} \|\vec{r}_u\|^{-2} - f_u \|\vec{r}_u\|^{-4} (\vec{r}_u, \vec{r}_{uu}).\end{aligned}\quad (6.9)$$

Thus, we get the continuous version of problem (6.8):

$$\begin{cases} \frac{\partial f}{\partial t} = \mathcal{D}_{uu} f + \mathcal{D}_{vv} f \\ f|_{t=0} = f_0. \end{cases}\quad (6.10)$$

It is worth noticing that if the depth D is constant (i.e., we have a non-informative depth map), this PDE problem becomes equivalent to the classic linear diffusion filtering (2.11), as the differential operator on the right side of the equation turns into the classic Laplacian up to a constant multiplier due to $\vec{r}_u = \vec{r}_v \equiv \text{const}$ and $\vec{r}_{uu} = \vec{r}_{vv} \equiv 0$. This allows for a “backward compatibility” of the proposed scale space to the classic Gaussian scale space when the depth map is not provided. Moreover, this property is satisfied locally, i.e., at points where D is continuous and the surface normal is parallel to the camera optical axis.

6.2.3 Well-posedness, numerical solution and its causality

In order to make use of the PDE problem (6.8), we have to ensure that it has a unique solution that depends continuously on the initial data f_0 . This is a fundamental property known as *well-posedness*.

To establish the well-posedness of problem (6.8) we use some of the results of [104]. We rewrite (6.8) in a vector form, i.e., $f(t) \in \mathbb{R}^{W \times H}$ and the application of L to f is represented by a matrix multiplication $\mathcal{A}f$. The coefficients of matrix \mathcal{A} depend only on \vec{r} and are explicitly deduced from its definition (6.6).

First, we apply theorem 4 of [104], which specifies sufficient conditions for well-posedness and extremum principle of a generic semi-discrete PDE problem (6.8). It is straightforward to show that our defined operator matrix satisfies all the conditions except the symmetry, i.e., it has vanishing row sums ($S3$), nonnegative off-diagonals ($S4$) and is irreducible ($S5$). Lipschitz-continuity ($S1$) is satisfied unconditionally as \mathcal{A} does not depend on f . The violated condition of the matrix symmetry ($S2$) is not required for well-posedness and extremum principle, as it is noticed afterwards [104, p. 76].

This proves that not only is the problem well-posed, but that the solution f respects the extremum principle allowing to set up the causality. It implies that the resulting filter is *causal in spatial image variables*, guaranteeing that no spurious features will appear during the smoothing process.

Furthermore, theorem 8 of [104] proves a sufficient criterion of stability for the following

explicit numerical scheme that allows to simulate the diffusion process:

$$\begin{aligned} f^{(n+1)} &= f^{(n)} + \tau \mathcal{A} f^{(n)} \\ f^{(0)} &= f_0. \end{aligned} \quad (6.11)$$

The condition of stability consists in limiting the temporal step of simulation τ . We reinterpret theorem 8 of [104] to obtain the analytic expression:

$$\tau \leq \tau^* = \left[2 \max_{\Omega_d} \left\{ \frac{1}{r_u^+ r_u^{+-}} + \frac{1}{r_u^- r_u^{+-}} + \frac{1}{r_v^+ r_v^{+-}} + \frac{1}{r_v^- r_v^{+-}} \right\} \right]^{-1}. \quad (6.12)$$

Now, using equations (6.11) and (6.12), we are able to perform the computation of the filter response for a given image $I = f_0$ and depth map D . For a constant time step τ , the quantity of resulting smoothing at the n -th iteration is then determined by $t^{(n)} = n\tau$. However, nothing prevents to vary τ from one iteration to the next one; we have only to respect the condition $\tau < \tau^*$ in order to have a stable process.

The designed filter simulates a uniform smoothing along the scene surface through a non-uniform diffusion in the image plane. Since smoothing along surfaces is, in principle, independent on the observer position, the proposed scale space can provide keypoints that are invariant to viewpoint position changes. This behavior is referred to as *viewpoint covariance*. It mainly comes from the definition of the first order differential operators (6.2), where we weight the derivative computed on two neighboring samples by the real distance between the corresponding sample points on the scene surfaces, inferred from the depth map. In practice, this diffusion process only approximates a diffusion process on the manifold defined by the depth map, due to depth errors and texture sampling precision. Therefore, the resulting scale space behavior will be approximately viewpoint covariant.

Some examples of images obtained with the proposed smoothing operator compared to the Gaussian smoothing are presented in Fig. 6.2. The input image is taken from the LIVE dataset [118, 119], which provides depth maps captured through a laser scanner. The viewpoint-covariant behavior could be observed on large scales (images (b), (c), (e), (f)): as the smoothing is propagating along the surface, and not uniformly in the image plane (as in case of the Gaussian scale space), the image becomes less smoothed when the distance increases.

6.2.4 Relation to Laplace-Beltrami operator

The gradient-like quantities defined in Eq. (6.2) enable to smooth the image *intrinsically* to the surface, using the geometric properties conveyed by depth. Even if the proposed Laplacian $\mathcal{D}_{uu} + \mathcal{D}_{vv}$ is not strictly invariant to orthogonal coordinate changes, in case of smooth surfaces and limited high frequency variations of texture, its response remains stable. Moreover, the operator vanishes on locally planar surfaces when the texture is a linear function, whereas classic 2D Laplacian, applied to the texture image only, does not.

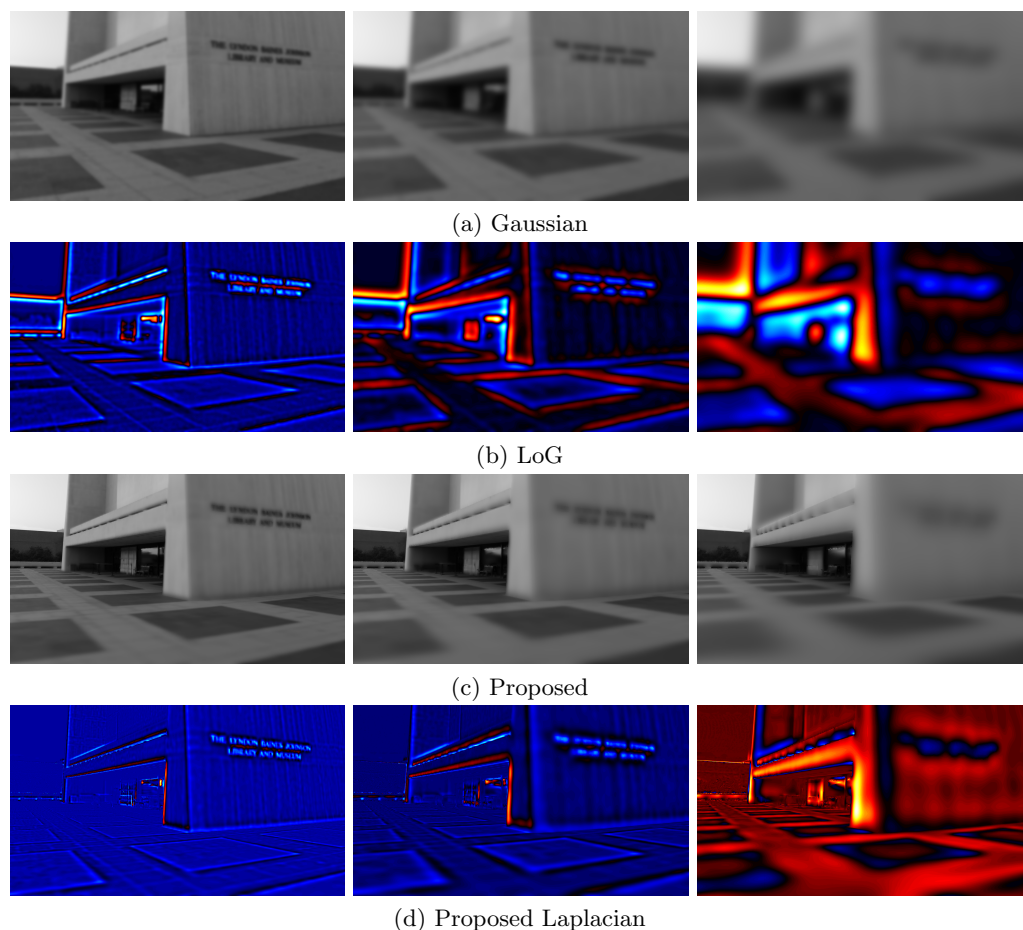


Figure 6.2 – An example of the proposed scale space and Laplacian on a real RGBD image compared to the Gaussian scale space and the corresponding Laplacian. Images in each row present obtained with different levels of smoothing: $\sigma = 5, 10$ and 25 for the Gaussian scale space and $\sigma = 0.1, 0.2$ and 0.5 for the proposed one. Bigger views of images (c) are shown in Fig. 6.1.

This allows for a *viewpoint-coherent* filter behavior, as observed in filtered images (see Section 6.4.1).

Filtering intrinsic to a surface is naturally formalized as a diffusion process on manifolds. These processes are classically described through the *Laplace-Beltrami operator* [109, 126, 127]. The scene surface parametrization \vec{r} may be regarded as a mesh with boundaries having a regular local topology but very irregular vertex spatial density. Thus, it is possible in theory to establish a diffusion process over such a mesh by using the Laplace-Beltrami operator. An important property of the surface-intrinsic characteristics of the Laplace-Beltrami operator is that the diffusion process is, in principle, “viewpoint-covariant”, i.e., its action in a given point of the scene surface is completely independent of the camera position and orientation. Nevertheless, as we aim at designing a scale space, such an approach exhibits two important difficulties.

First, such a filter might not necessarily engender a scale space in the parametrization

domain (image plane). One may easily verify that the proposed Laplacian $\mathcal{D}_{uu} + \mathcal{D}_{vv}$ acts exactly as the Laplace-Beltrami operator at points where the tangent plane to \vec{r} is parallel to the camera plane. However, in general the proposed continuous Laplacian is not equivalent to Laplace-Beltrami operator. This is partially a consequence of the absence of a term containing mixed derivative f_{uv} in the analytic expression defining the operator (6.9). A Laplacian could be defined differently, but some coefficients of the operator matrix \mathcal{A} (especially corresponding to the absent mixed derivative term) would become negative, and neither the scale space properties nor the numerical stability would be ensured by referring to [104].

Second, the Laplace-Beltrami operator is known to be hardly discretizable. Specifically, numerous discrete Laplacian operators do not converge to the Laplace-Beltrami operator, but satisfy some other desired properties and are largely used in practice [126, 128]. As for our Laplacian, it is easy to verify that, referring to the results in [128], it satisfies *locality* (LOC), *linear precision* (LIN), and *positive weights* (POS) properties.

6.2.5 GPU implementation of the proposed filter

As mentioned before, computing the filter output consists in an iterative process according to Eq. (6.11). Since the operator matrix \mathcal{A} is sparse, it is possible to parallelize the filtering process, as the value of a given pixel at iteration $n + 1$ depends only on a few pixels at iteration n . This allows to compute the designed diffusion process on GPU in a very efficient way. For our experiments in this work, we implemented the designed numerical scheme using OpenGL utilities. Our implementation is outlined in the following.

We first allocate several textures to store the input image (T_{in}), the output image (T_{out}) and the nonzero entries of the operator matrix \mathcal{A} . More precisely, there are only five non-zero entries in each line of \mathcal{A} , forming the defined discrete Laplacian operator support, situated at left, right, top, bottom and center pixel positions with respect to the current position u, v . In our implementation, we compute these coefficients in a single CPU pass on the input image, and assign them to five separate single-channel textures.

The rendering is performed into an off-screen pixel buffer bound to the output image texture. The updating step (Eq. (6.11)) is implemented in the fragment shader: the Laplacian is computed using the stored coefficients, and then weighted by the time step τ and added to the image. After the rendering, we swap the textures T_{in} and T_{out} . This is performed without any time-consuming pixel transfer, simply by rebinding the two textures in a crosswise manner. The rendering step is repeated until the target level of smoothing σ is reached. Then the pixel data can be read back from GPU memory and transmitted to the application.

It worth noticing that the described process makes use of the standard graphic pipeline and does not require any advanced GPGPU¹ technology such as CUDA, which is hardware

¹General-Purpose computing on Graphics Processing Units

vendor-specific. Consequently, the designed scale space may be rendered on any OpenGL-compliant graphic hardware. Due to wide applicability of OpenGL, our approach could perform efficiently on a large spectrum of devices, including modern smartphones, tablets and even drones (equipped with a depth sensor).

6.3 Proposed detector

In this section, we use the scale space described above in order to design a novel RGBD keypoint detector. A keypoint detector mainly consists of three parts: (i) initial keypoint candidates selection criteria selecting a set of locations with corresponding scales in the input image, (ii) a candidate filtering, aimed at rejecting candidates that are likely less repeatable, and (iii) an accurate localization procedure of remaining keypoints. We describe in detail each step in the following.

6.3.1 Candidates selection

Similarly to the popular SIFT detector [22], the initial keypoint candidates in our proposed detector are selected as local extrema of the Laplacian operator. The SIFT detector uses the classic image Laplacian in (2.11), approximated by a difference of Gaussians, i.e., by subtracting consecutive levels of the scale space. In our case, the proposed Laplacian operator (6.7) is used. We do not need to approximate it by taking differences of the smoothed images, as we simulate the diffusion process where the Laplacian is computed explicitly at each iteration.

However, the main difference with respect to the SIFT detection criterion is that *we look only for spatial local extrema at each scale*, i.e., over variables u, v , and not for the local extrema along both spatial and scale coordinates, i.e., over u, v and σ . Indeed, in our experiments we found that keypoint candidates issued from extrema along the σ axis are generally unstable. A possible reason for that is related to the intrinsic nature of our proposed scale space: the smoothing injected into the image is spatially varying, so that σ represents a scale with respect to the scene geometry, and not the scale in the image plane. On the other hand, local minima and maxima of our Laplacian (6.7) with respect only to spatial image variables u, v turn out to be very repeatable, and reveal distinctive blob-like structures on the scene surface. Such a setting, where the keypoints are searched on different scale levels independently, is a variation of the *multiscale detector* proposed by [25].

More precisely, we search for keypoints in a multiscale representation obtained in a similar way to [50], by progressively smoothing and subsampling the input image. We construct a set of smoothed images of levels $\sigma_0, 2\sigma_0, 4\sigma_0, \dots, 2^{M-1}\sigma_0$. Here σ_0 is a constant, its value is set manually according to the depth measurement unit used in the depth map. Each subsequent image is subsampled by two in each dimension with respect to the

previous one: it reveals larger scale structures and allows to reduce the computation time. The number of levels M is limited by the image size. In our experiments we keep $M = 5$, which is enough to detect blobs on a large variety of scales.

6.3.2 Candidates filtering

A common practice to reduce the number of poorly repeatable keypoints is to threshold a keypoint score, keeping only candidates with highest scores. In a similar way, we keep only those initial candidates that have a Laplacian operator response greater in absolute value than a threshold.

Once the initial candidates are selected, we apply Harris cornerness measure [39] similarly to ORB [41] and CenSurE [113]. This technique allows to filter out the keypoints localized on the edges that are likely to be unstable: they can move along the edge when the camera position changes.

6.3.3 Accurate localization

In order to localize keypoints with subsample precision, we apply the accurate localization procedure presented in [60], reducing it from three dimensions (u, v, σ) to two. More precisely, let L be the Laplacian response, (u, v) a candidate point, (u^*, v^*) an accurately localized local extremum, $\vec{\delta} = (u^* - u, v^* - v)^T$. We develop the Taylor expansion of $L(u^*, v^*)$ with respect to (u, v) :

$$L(u^*, v^*) \approx L + (L_u \ L_v) \vec{\delta} + \frac{1}{2} \vec{\delta}^T \begin{pmatrix} L_{uu} & L_{uv} \\ L_{uv} & L_{vv} \end{pmatrix} \vec{\delta}. \quad (6.13)$$

L and its derivatives on the right side of the equation above are taken at point (u, v) . Deriving (6.13) and exploiting the fact that (u^*, v^*) is a local extremum, i.e., $L_u|_{u^*, v^*} = L_v|_{u^*, v^*} = 0$, we obtain:

$$\vec{\delta} = - \begin{pmatrix} L_u \\ L_v \end{pmatrix} \begin{pmatrix} L_{uu} & L_{uv} \\ L_{uv} & L_{vv} \end{pmatrix}^{-1}. \quad (6.14)$$

Similarly to a known SIFT implementation [80], we apply this procedure iteratively, cumulating the offset and reinterpolating the derivatives of L . If after a fixed number of iterations the displacement $\vec{\delta}$ remains large, the keypoint candidate is considered as unstable and rejected.

After the keypoints are detected, in order to be able to use standard descriptors, we derive their on-screen scale. We consider keypoint k as a sphere of radius σ_k , situated on the scene surface. σ_k is simply equal to the scale level where the keypoint is detected. Assuming that its center is projected on the screen at point (u_k, v_k) , obtained from the accurate localization procedure, we apply the pinhole camera model to get the output

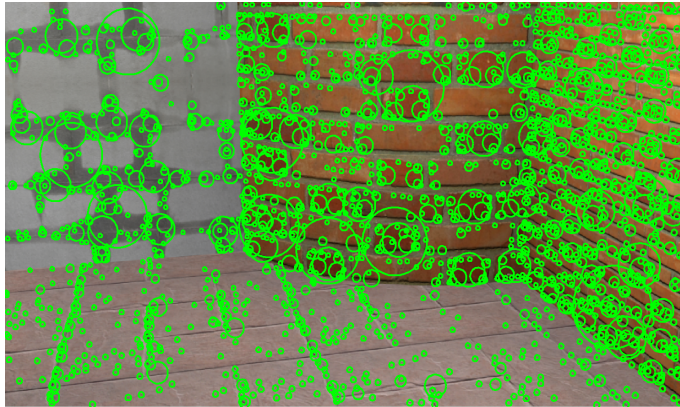


Figure 6.3 – Keypoints detected using the proposed method in an image of *Bricks* sequence.

(on-screen) keypoint scale (similarly to Eq. (4.2)):

$$s_k = \frac{\sigma_k W}{2D(u_k, v_k) \tan \frac{\omega}{2}}. \quad (6.15)$$

The set of triples $\{(u_k, v_k, s_k)\}_k$ constitutes the detector output and is sent to the descriptor extraction stage. An example of detected keypoints in an image from *Bricks* sequence is given in Fig. 6.3. We notice that the dominant direction estimation and the consequent rotational normalization of the patches, required to have in-plane rotation-invariant descriptors, are performed on the descriptor side.

6.4 Experiments

6.4.1 Viewpoint-covariant filter behavior illustration

The goal of defining a new Laplacian operator was to render the smoothing process intrinsic to the surface and reduce its dependence on the camera position. Since this behavior is not straightforward to formalize, we begin the experimental part with an illustration of this effect.

Specifically, we measure pixelwise similarity between a filtered reference view and a reprojection of filtered test view. The simulation follows the scheme in Fig 6.4. A scene is captured from two different positions obtaining two views (I_1, D_1) and (I_2, D_2) . As the camera positions and orientations are known and depth-maps are available, we may reproject the first view texture image on the second camera plane, i.e., reconstruct the second view from the first one. The corresponding image is referred to as $I_{1 \rightarrow 2}$. When no filter is applied, $I_{1 \rightarrow 2}$ is close to I_2 at pixels whose 3D origins are present in both images. The testimony of the viewpoint covariant filter behavior will consist in a limited difference between $I_{1 \rightarrow 2}$ and I_2 , when the corresponding filter is applied to both input views.

We compare our filter with Gaussian and Perona and Malik’s filters. The amount of

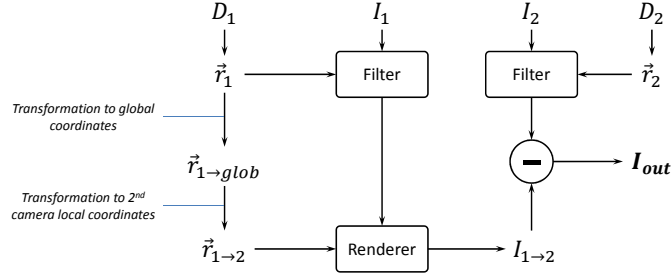


Figure 6.4 – Test setting to assess the viewpoint covariance of a given smoothing filter

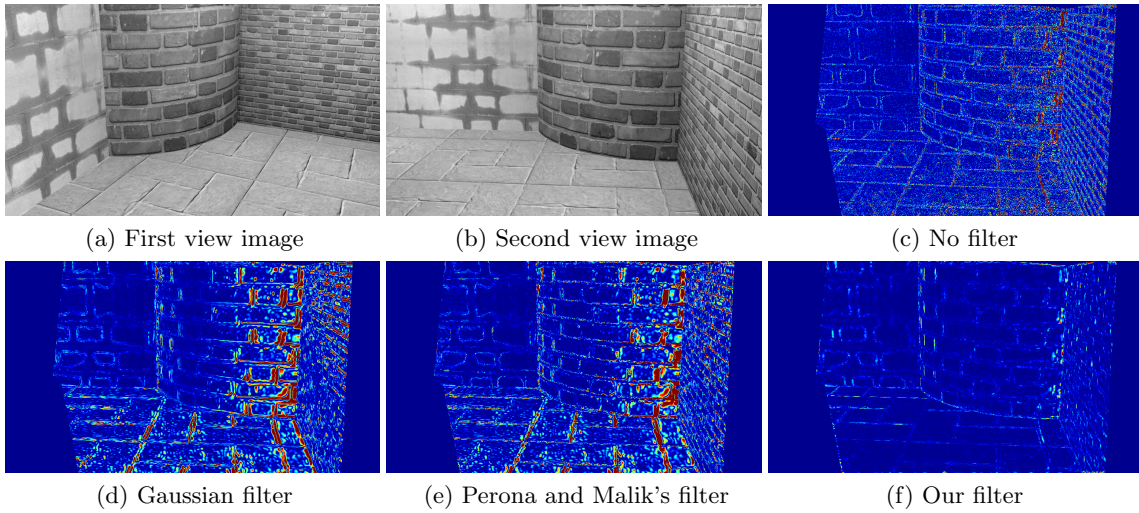


Figure 6.5 – Two views I_1 and I_2 of *Bricks* sequence and the reconstruction difference $I_{out} = |I_2 - I_{1 \rightarrow 2}|$ computed using the test setting on Fig. 6.4 without filter (c), with Gaussian filter (d), Perona and Malik filter (e), and our filter (f).

smoothness, as well as all the other filter parameters, are set up experimentally in such a way that the filtered images are visually similar. To reduce sampling and depth quantization effects, instead of pixel-by-pixel difference, we take a set of neighboring points in $I_{1 \rightarrow 2}$ and compute the minimal difference with the corresponding pixel value in I_2 . More precisely, we take 8 points on a circle of 1 pixel radius and its center. As $I_{1 \rightarrow 2}$ is sampled from a scattered point set obtained by the reprojection, we may interpolate it in the desired way. An example of the filtering process is reported in Fig. 6.5 for the *Bricks* content, where one can see that filtered images from different viewpoints match better with each other than those obtained through conventional smoothing processes. This illustrates pictorially why the proposed scale space yields a certain viewpoint change robustness. In the following we discuss more quantitatively this property for a keypoint detection scenario.

6.4.2 Repeatability evaluation

We then proceed to the detector evaluation, beginning with the mid-level evaluation protocol as described in Section 2.5.1, focusing on the repeatability part (i.e. no descriptors are involved in this experiment).

We compare the proposed detector to the standard SIFT detector (*VLFeat* [80] implementation) and to Viewpoint Invariant Patches [82] (original authors' implementation), which incorporates a keypoint detector that uses the depth map. Three RGBD test sequences are used, representing different content, containing significant viewpoint position changes: *Bricks* (20 images), *Graffiti* and *House* (25 images, see Section 2.5.2.1 for more details). As in the preceding chapter, the repeatability score of each detector is computed for two values of the overlap error threshold $\eta = 0.5$ and $\eta = 0.25$. The results of this experiment are shown in Fig. 6.6.

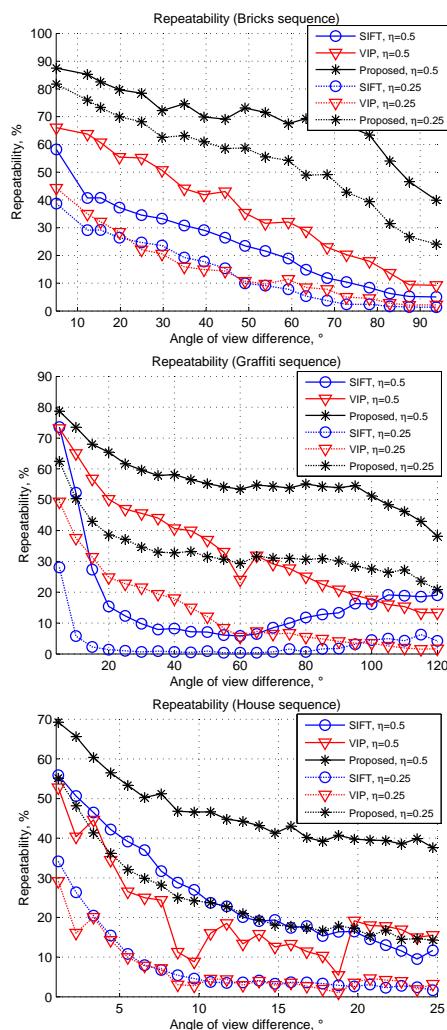


Figure 6.6 – Repeatability score on synthetic RGBD sequences in function of angle of view difference between reference and test images.

It can be observed that, for both values of the overlap η , the proposed detector clearly outperforms the two other approaches. Moreover, even in the tighter condition $\eta = 0.25$ our proposed detector demonstrates a comparable or better repeatability to the two other detectors, even when those are matched using the more tolerant value $\eta = 0.5$.

It is worth noticing that in this experiment the number of keypoints detected by SIFT and our proposed method remain comparable (vary between 1000 and 2500 depending on the input image), however VIP detects generally more keypoints (up to 5000).

6.4.3 Scene recognition using Kinect images

In this section, we analyze the performance of the proposed RGBD detector in the scene recognition scenario requiring repeatable local features. We use the dataset presented in Section. 2.5.2.2. The problem is, e.g., for a mobile robot or a drone, to recognize the location (room) where it is situated, solely using visual sensors data and prior knowledge, i.e., a database of local features representing different locations.

This problem may be reduced to a simple classification task. In order to classify a given image I with respect to a set of references $\mathcal{R} = \{(I_k, l_k)\}_{k=1}^K$, where l_k represent the ground truth class label (i.e., room number), we simply look for an index k^* of an image from \mathcal{R} that represents the best match against I . The best match is the one that maximizes an image similarity score, which is computed as follows.

We detect keypoints in both images and match their corresponding descriptors. The descriptors are matched testing all descriptor pairs: for each given descriptor from the first image we pick the closest descriptor from the second image. If the number of closely matching descriptors (those that have a distance less than a given matching selectivity threshold t) is large enough, then the two images are assumed visually similar. Thus, to recognize the location, we select the most similar image and take its label.

Specifically, let $N_{feat}(I)$ denote the number of features extracted from image I and $N_{matches}(I, I_k, t)$ the number of matching descriptor pairs having the inter-descriptor distance less than a threshold t . Then, the image-level similarity score is given by

$$J(I, I_k, t) = \frac{N_{matches}(I, I_k, t)}{N_{feat}(I) + N_{feat}(I_k) - N_{matches}(I, I_k, t)}. \quad (6.16)$$

The best match with respect to the given set of references \mathcal{R} is the one maximizing J :

$$k^*(I, t) = \arg \max_k \{J(I, I_k, t)\}. \quad (6.17)$$

The label l_{k^*} is then attributed to I . If the ground truth label of I is equal to l_{k^*} , the image is classified correctly, i.e., the location is correctly recognized.

Differently to the previous experiment, here we involve complete feature extraction pipelines (containing both detector and descriptor). We compare the following local feature extraction methods, representing well-known techniques to deal with out-of-plane rotations:

- original VIP features [82],
- standard SIFT features (*VLFeat* [80] implementation, referred to as DOG+SIFT),
- SIFT descriptors undergoing affine normalization [116], bootstrapped with SIFT keypoints (*VLFeat* implementation, referred to as DOG+AFFINE),
- our proposed detector with standard SIFT descriptors (referred to as PROPOSED+SIFT),
- SIFT descriptors undergoing affine normalization [116], bootstrapped with our proposed detector (referred to as PROPOSED+AFFINE).

To keep the comparison fair, for all the detectors we keep at most 1000 keypoints with the highest scores (Laplacian response). All input parameters of all the methods keep their default values.

All the descriptors are represented by 128-dimensional numerical vectors. There are two options to measure the inter-descriptor similarity:

1. simple Euclidean norm of inter-descriptor difference taken as a vector;
2. ratio of Euclidean distances to the 1st closest and the 2nd closest descriptor, as proposed in [22].

As previously, for each method we simply use the option that performs better: the first one is used with DOG+AFFINE and PROPOSED+AFFINE, the second one is used for the rest. For a fair comparison between the tested methods, we perform the experiment for a set of matching selectivity threshold values t , as there is no reason that different features will perform equally well with the same threshold. Here we present only the best result of each method over all the used values of t .

We conduct two experiments. First, we match all the images against each other computing confusion matrices. This allows to classify each given image with respect to all the others, so that the reference set is different for each input and consists of 74 remaining images. The portion of correctly classified images per method in this setting is reported in Fig. 6.7 (left bars, referred to as *complete*). Then we switch to a more practical scenario. We randomly select a single image per location, forming a reference set \mathcal{R} of 15 images, and then classify all the remaining images with respect to the given reference set. The obtained recognition accuracy is also shown in Fig. 6.7 (right bars, referred to as *Single ref.*). In order to avoid the influence of the random reference selection, we repeat the experiment 1000 times.

Our proposed detector achieves a higher recognition accuracy in both the experiments. Affine normalization compensates the perspective distortions on the descriptor computation stage, yielding improved performance compared to the unnormalized SIFT descriptors. For qualitative comparison, an additional illustration of matching using these descriptors is given in Fig. 6.8: keypoints detected with the proposed detector generally provide more

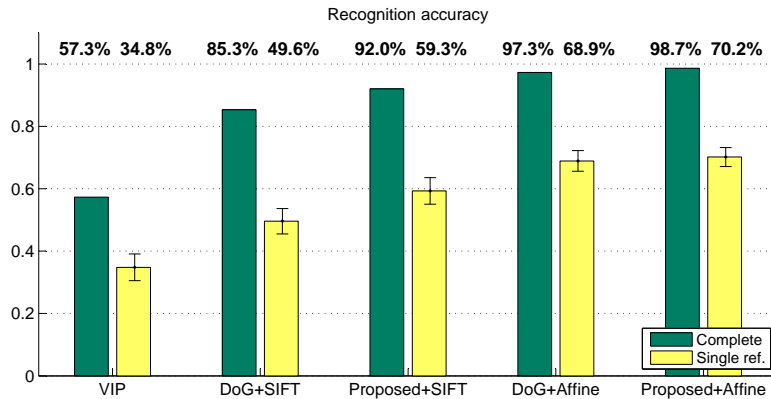


Figure 6.7 – Accuracy of scene recognition on the images of Fig. 2.6. The left bars (*complete*) are computed by matching a query image to all the remaining 74 images in the dataset. In the *single reference* classification, instead, each image is classified using a set of 15 randomly selected reference images (one per class). In this case the reported results are the average over 1000 repetitions, corresponding standard deviation is displayed.

consistent and regular correspondences. Moreover, in spite of the noise present in depth maps and their incompleteness (some areas have undefined depth, which is a common problem of infrared depth sensors), our proposed approach is able to detect repeatable keypoints. However, the degraded depth map quality is probably the reason for the limited performance of VIP.

In this experiment we also report that the keypoint detection time taken by our proposed detector averaged over all the 75 images is about 0.42 seconds². It is nearly half of the average computation time of VLFeat SIFT detector, which is implemented in a single thread on CPU, but uses vectorial processor instructions in order to speed up the processing.

6.5 Conclusion

In this chapter we have proposed a multiscale representation and a keypoint detector for RGBD images. First, we have proven that the proposed multiscale representation is causal in the image plane, i.e., it engenders a scale space. Second, since the generation of this scale space corresponds to an approximated diffusion along the surfaces of the scene, the resulting keypoints have a higher stability to large viewpoint changes than conventional, isotropic scale spaces. Finally, the proposed diffusion scheme is numerically stable, linear in the input texture image, and can be efficiently computed on GPU using OpenGL.

These properties have been leveraged to design a novel multiscale detector, which offers a significant gain in terms of keypoint repeatability with respect to viewpoint position changes, both on synthetic and real RGBD images, in a computational time comparable to alternative conventional detectors such as VLFeat SIFT.

²Run on a Windows 7 machine with 12-core 3.5 GHz Intel Xeon CPU, 16 GB RAM and NVidia Quadro K620 graphic card.

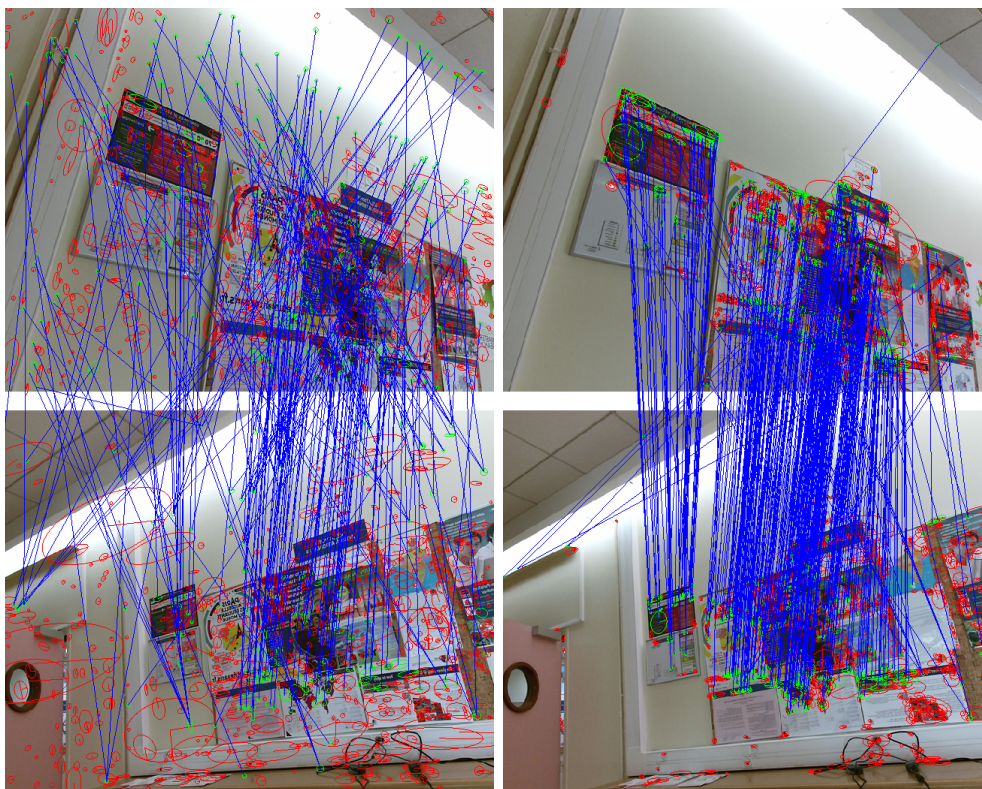


Figure 6.8 – Raw (putative) feature matches between two RGBD images from *Board* scene obtained with affine-covariant descriptors on top 1000 keypoints in each image. Left: SIFT detector (243 matches), right: the proposed detector (419 matches).

Chapter 7

$O(1)$ accurate image smoothing operator for non-uniform multiscale representations

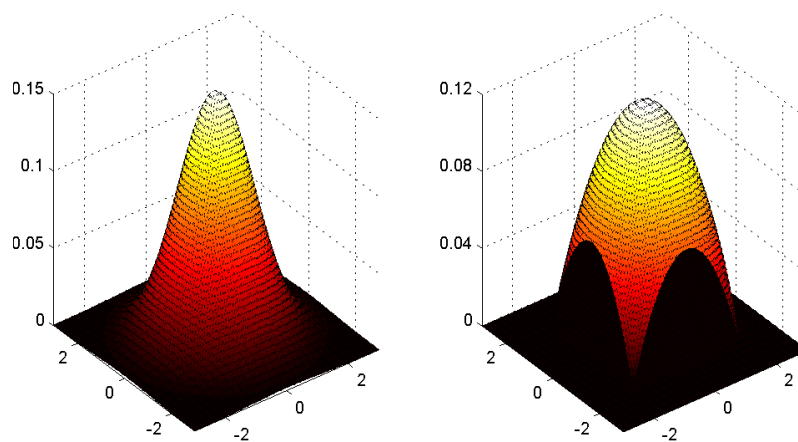


Figure 7.1 – Bi-dimensional Gaussian filter kernel of unit variance (left) compared to the proposed filter kernel (right). The latter provides a more accurate (closer to the Gaussian) output than the box filter, but the convolution may still be computed in $O(1)$ operations as the kernel surface is polynomial.

7.1 Overview

Low pass image filtering (smoothing) is a basic operation in many image processing applications, including image matching through local features. For instance, Gaussian smoothing is used to generate a scale space and detect interesting points in a scale-invariant manner [22]. While this approach can provide very stable keypoints, its computation may not be feasible when power or battery resources are limited. As an alternative, one can use simpler and faster smoothing techniques such as the box filter [67], although this can imply

lower feature matching performance under some transformations. The trade-off between computational efficiency and accuracy of the smoothing filter is therefore a key factor in the stability and repeatability of extracted features.

In this chapter we continue to investigate the blob detection in RGBD images by proposing an alternative blob detection approach, which does not require a diffusion process simulation that can be unaffordable in practice. To this end, we first introduce a fast and accurate general purpose image smoothing filter based on integral images. It has the same computational complexity as the box filter, i.e., the response is computed in constant time at any image point and any smoothing level. However, our proposed filter provides improved rotational invariance and a better approximation of Gaussian smoothing, which allows for better performance compared to the box filter in several tested keypoint detection scenarios. Then, we use this filter to design a blob detector for RGBD images, aimed at improved repeatability under viewpoint changes.

7.2 Image smoothing operators for keypoint detection

Image smoothing is a well-explored research area. Some recent works are focused on complexity reduction, e.g. [129]. In this work we focus on image smoothing complexity in context of salient visual point detection.

Gaussian smoothing

Bi-dimensional Gaussian filter is one of the most commonly used image smoothing operators. To smooth an input image $f(x, y)$ up to a given smoothing level σ , one would typically compute the following quantity:

$$f_{out}(x, y, \sigma) = \frac{1}{2\pi\sigma^2} \iint_{\mathbb{R}^2} f(x, y) e^{-\frac{(x-u)^2 + (y-v)^2}{2\sigma^2}} dudv. \quad (7.1)$$

Thanks to the exponential decay, the integral may be contracted to a reasonably compact support, for example applying the well known rule of 3σ [22]. Moreover, the convolutional kernel is separable, allowing to replace the two-dimensional convolution by two simple ones.

$$f_{out}(x, y, \sigma) \approx \frac{1}{2\pi\sigma^2} \int_{-s}^{+s} e^{-\frac{(y-v)^2}{2\sigma^2}} \left[\int_{-s}^{+s} f(x, y) e^{-\frac{(x-u)^2}{2\sigma^2}} du \right] dv. \quad (7.2)$$

Thus, the numerical filter, obtained by replacing the integrals by integer summations, has *linear complexity* in function of σ for computing the response in a given spatial point: discrete version of $f_{out}(x, y, \sigma)$ at a given point (x, y, σ) may be computed in $O(\sigma)$ operations.

The separability described by Eq. (7.2) does not generally hold for spatially vary-

ing σ , i.e., if $\sigma = \sigma(x, y)$. This is a particular kind of *isotropic non-uniform multiscale representations*. The computational complexity then raises to $O(\sigma^2)$.

We recall in the following two important properties of the Gaussian filter:

Relation to the heat diffusion equation. It is known that Eq (7.1) is the solution of the differential problem for the heat diffusion equation. The input image becomes the initial condition of this problem, and the amount of injected smoothing σ is related to the diffusion time. This enables to establish a set of important properties, allowing to use this filter to engender a proper scale space [104]. The latter may be understood as a representation of the internal image structure at multiple scales. This kind of representations is predominant in vision problems, such as keypoint detection for image matching [50].

Rotational invariance. The Gaussian convolutional kernel is radially symmetric. This implies perfect invariance of the filter response to in-plane image rotations.

A notable successful application of the Gaussian smoothing motivated by these properties is SIFT image features [22]. Similarly, the Gaussian scale space is employed in the MPEG Compact Descriptors for Visual Search standard [37]. However, due to the linear computational complexity, this filter has been systematically criticized [67, 130–133], and approximate solutions such as the box filter have become popular.

Box filter

The box filter response is given by the following expression:

$$f_{out}(x, y, s) = \frac{1}{s^2} \int_{x-\frac{s}{2}}^{x+\frac{s}{2}} \int_{y-\frac{s}{2}}^{y+\frac{s}{2}} f(u, v) dudv \quad (7.3)$$

The filter thus may be seen as a convolution with a kernel taking a constant value within the rectangular support $\Omega = [-\frac{s}{2}, \frac{s}{2}] \times [-\frac{s}{2}, \frac{s}{2}]$, i.e., taking the average image value in Ω . Here s represents the scale parameter, or the amount of smoothing required on the output. Integral image technique [68] allows to avoid the explicit computation of the integral, providing *constant computational complexity*, i.e., independent of the amount of smoothing s , provided that the integral image has been precomputed. This principle is illustrated on Fig. 7.2. Due to its simplicity and efficiency, the box filter is largely used in different applications and scenarios. Some complex image filters, such as the guided filter [108] or some bilateral filter variant [134], employ the box filter. In the vision applications, the box filter is often used to approximate the time-consuming Gaussian scale space in SIFT-like detectors. Thus, [130] proposes to select the keypoint candidates in a *Difference-of-Mean* (DoM) image pyramid built with the box filtering instead of the original *Difference-of-Gaussian* pyramid. In a similar way integral images [131] and their generalizations [132] are combined with SIFT detection strategy. Speeded Up Robust Features (SURF) [67] also make use of the box filtering for the keypoint detection. Center Surrounded Extrema

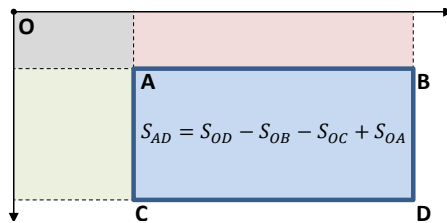


Figure 7.2 – Integral image principle [68]. For a given function, the integral over any rectangle AD may be computed immediately, if for any point X on the plane the integral over OX is known. The latter ones are precomputed once during the filter initialization stage, and form the integral image (the rectangles are denoted by their diagonals).

(CenSurE) detector [113] uses slanted integral images to detect the interesting points without computing explicitly a pyramidal representation. BRISK [46] employs the box smoothing to compute the descriptor.

An issue when using the box filter in vision problems comes from its sensitivity to rotations. Due to the sharp corners of the rectangular convolutional kernel, rotating the input image of 45° may drastically change the distribution of interesting points with respect to the structure of the image. For instance, it is known that SURF features suffer from limited rotational invariance, which is partially due to the use of box filtering [67].

As for non-uniform multiscale representations, when the smoothing becomes spatially varying, it does not imply any additional computational complexity as in case of the Gaussian filter, since the output response complexity does not depend on the kernel support size. However, due to inaccuracy of the box filter, such multiscale representations might be inherently inaccurate. This leaves a room for an accurate smoothing filter with a low computational complexity that does not depend on σ , allowing for efficient isotropic non-uniform multiscale representations.

There are several approaches that propose a trade-off between the box and the Gaussian filters in terms of computational cost and accuracy. In *repeated integration* technique [135] a higher order smoothing is achieved by applying the box filter several times, which is shown to be equivalent to a convolution of the input image with a more smooth kernel. This, however, requires to reinitialize the filter after each pass, and does not easily generalize to spatially varying kernels. *Symmetric weighted integral images* [136] provide a more accurate smoothing using a piecewise linear kernel computed with 5 integral images. This approach is applied to the keypoint detection in [133], and is compared to the Gaussian and box filters. *Stacked integral image* approach [137] uses the standard box filter (and thus a single integral image) to approximate more complex kernel shapes by summing its piecewise responses computed over different regions. A similar idea in application to the feature detection is discussed in [138]. *Kernel integral images* [139] is a general framework of computing convolutions with smooth kernels using integral images. An approximation of 2D Gaussian smoothing is proposed with 9 integral images. Our proposed filter is a particular case of the kernel integral images, providing a close approximation of the Gaussian with

5 integral images (if no subpixel precision needed, only 4 are necessary). We also show how to interpolate the integral images properly to achieve the subpixel precision, i.e., to compute the exact convolution continuously for fractional pixel position and support size.

7.3 The proposed filter design

7.3.1 Filter kernel definition

The key idea of the proposed approach consists in approximating the Gaussian kernel by a function that can be computed using *image moments*, which are efficiently represented by the integral images.

Let us consider the following function:

$$K(x, y) = A - B(x^2 + y^2), \quad (7.4)$$

where A and B are constants. It is straightforward to show that the convolution F of the image with this kernel may be decomposed as follows:

$$\begin{aligned} F(x, y, s) &= \iint_{\Omega} f(u, v) K(x - u, y - v) dudv \\ &= [A - B(x^2 + y^2)] I_1(x, y, s) - BI_{x^2+y^2}(x, y, s) \\ &\quad + 2B [xI_x(x, y, s) + yI_y(x, y, s)] \end{aligned} \quad (7.5)$$

The integrals $I_{(\cdot)}$ denote *image moments*. Specifically, I_1 is the zero-order moment that is equivalent to the box filter output, I_x and I_y are first-order moments with respect to x and y , and $I_{x^2+y^2}$ is the sum of two second-order moments:

$$I_x(x, y, s) = \iint_{\Omega} u f(u, v) dudv \quad (7.6)$$

$$I_y(x, y, s) = \iint_{\Omega} v f(u, v) dudv \quad (7.7)$$

$$I_{x^2+y^2}(x, y, s) = \iint_{\Omega} [u^2 + v^2] f(u, v) dudv. \quad (7.8)$$

The main point of using K as the filter kernel is that all the image moment integrals may be computed in constant time using the integral image technique: an integral image is precomputed for each image moment and is then used to obtain the required value. Therefore, the convolution F may be computed in $O(1)$ operations for any x , y and s .

To design a suitable filter whose response is expected to be close to the Gaussian filter, we need to choose proper values of A and B . This is done assuming that

1. K must be nonnegative within the support Ω , in order to have a smoothing filter,

2. $\iint_{\Omega} K(u, v) dudv = 1$,
3. there is a linear relation between the scale parameter and the standard deviation of the Gaussian filter, e.g. $s = C\sigma$.

To satisfy the first constraint we simply set $A = \frac{Bs^2}{2}$. The second one then gives directly the kernel normalization constant: $\iint_{\Omega} K(u, v) dudv = As^2 - \frac{Bs^4}{6} = \frac{Bs^4}{3}$. Dividing the kernel by this value and applying $s = C\sigma$ we obtain the filter kernel expression:

$$K(x, y) = \frac{3}{2s^2} - \frac{3}{s^4}(x^2 + y^2) = \frac{3}{C^2\sigma^2} \left(\frac{1}{2} - \frac{x^2 + y^2}{C^2\sigma^2} \right). \quad (7.9)$$

We finally tune the constant C to minimize the total squared difference between K and the Gaussian kernel, i.e. in order to assure the response close to the Gaussian one, obtaining $C \approx 3.5$. The resulting kernel K is shown in Figure 7.1.

7.3.2 Continuous response computation

To achieve subsample precision with the designed filter, a specific interpolation has to be applied to the integral images. The interpolation coefficients are derived assuming that the input image is a piecewise-constant function, taking a constant value at each pixel position. We show then, that the exact value of each image moment in this case may be obtained through a linear interpolation with proper weights of the four closest neighbors of each vertex of the support Ω .

Specifically, let $J(u, v)$ be the integral image corresponding to a given image moment $I(u, v) = \iint_{\Omega} m(u, v) f(u, v) dudv$, $x = x_0 + \alpha$, $y = y_0 + \beta$, where $x_0 \in \mathbb{Z}$, $y_0 \in \mathbb{Z}$, and $0 < \alpha, \beta \leq 1$. According to the principle of integral images, the following relations take place:

$$\begin{cases} J(x_0 + \alpha, y_0) = J(x_0, y_0) + \int_{x_0}^{x_0 + \alpha} \int_0^{y_0} m(u, v) f(u, v) dudv \\ J(x_0, y_0 + \beta) = J(x_0, y_0) + \int_0^{x_0} \int_{y_0}^{y_0 + \beta} m(u, v) f(u, v) dudv \\ J(x_0 + \alpha, y_0 + \beta) = J(x_0 + \alpha, y_0) + J(x_0, y_0 + \beta) - J(x_0, y_0) + \int_{x_0}^{x_0 + \alpha} \int_{y_0}^{y_0 + \beta} f(u, v) m(u, v) dudv \end{cases} \quad (7.10)$$

Using the assumption that the input image f is a piecewise constant function, i.e., $f(x_0 + \alpha, y_0 + \beta) \equiv f(x_0, y_0)$, we develop the double integral from the first equation as follows

(the integrals in the other two equations might be developed similarly):

$$\int_{x_0}^{x_0+\alpha} \int_0^{y_0} m(u, v) f(u, v) dudv = \int_0^{y_0} f(x_0, v) \left[\int_{x_0}^{x_0+\alpha} m(u, v) du \right] dv = \quad (7.11)$$

$$= \sum_{k=0}^{y_0-1} f(x_0, k) \int_k^{k+1} \int_{x_0}^{x_0+\alpha} m(u, v) dvdu. \quad (7.12)$$

The last double integral does not depend on the input image and might be easily computed analytically for all the image moments:

$$I = I_1 : \quad \int_k^{k+1} \int_{x_0}^{x_0+\alpha} m(u, v) dvdu = \int_k^{k+1} \int_{x_0}^{x_0+\alpha} dvdu = \alpha \quad (7.13)$$

$$I = I_x : \quad \int_k^{k+1} \int_{x_0}^{x_0+\alpha} m(u, v) dvdu = \int_k^{k+1} \int_{x_0}^{x_0+\alpha} u dvdu = \alpha \frac{2x_0 + \alpha}{2} \quad (7.14)$$

$$I = I_y : \quad \int_k^{k+1} \int_{x_0}^{x_0+\alpha} m(u, v) dvdu = \int_k^{k+1} \int_{x_0}^{x_0+\alpha} v dvdu = \alpha \frac{2k + 1}{2} \quad (7.15)$$

$$I = I_{x^2} : \quad \int_k^{k+1} \int_{x_0}^{x_0+\alpha} m(u, v) dvdu = \int_k^{k+1} \int_{x_0}^{x_0+\alpha} u^2 dvdu = \alpha \frac{3x_0^2 + 3x_0\alpha + \alpha^2}{3} \quad (7.16)$$

$$I = I_{y^2} : \quad \int_k^{k+1} \int_{x_0}^{x_0+\alpha} m(u, v) dvdu = \int_k^{k+1} \int_{x_0}^{x_0+\alpha} v^2 dvdu = \alpha \frac{3k^2 + 3k + 1}{3} \quad (7.17)$$

These expressions plugged into the initial system of equations (7.10) provide us with the following key observation: each double integral in Eq. (7.10) may be split into two factors, where the first one depends only on the image and the current pixel location (x_0, y_0) , and the second one depends on (x_0, y_0) , α and β , but not on the input image f .

To proceed, let us take the image moment I_x (the corresponding integral image is then denoted by J_x). Since x_0 and y_0 are integer, $J(x_0, y_0)$, $J(x_0 + 1, y_0)$, $J(x_0, y_0 + 1)$ and $J(x_0 + 1, y_0 + 1)$ are computed and stored on the filter initialization phase according to the classic integral image technique. By virtue of continuity of the image moment, Eq. (7.10)

gives the following relations when $\alpha = \beta = 1$:

$$\left\{ \begin{array}{l} J_x(x_0 + 1, y_0) = J_x(x_0, y_0) + \phi_f(x_0, y_0) \left(\alpha \frac{2x_0 + \alpha}{2} \right) \Big|_{\alpha=1} \\ J_x(x_0, y_0 + 1) = J_x(x_0, y_0) + \psi_f(x_0, y_0) \beta \Big|_{\beta=1} \\ J_x(x_0 + 1, y_0 + 1) = J_x(x_0 + 1, y_0) + J_x(x_0, y_0 + 1) - J_x(x_0, y_0) + \\ \quad + f(x_0, y_0) \left(\alpha \beta \frac{2x_0 + \alpha}{2} \right) \Big|_{\alpha=1, \beta=1} \end{array} \right. \quad (7.18)$$

Here $\phi_f(x_0, y_0)$, $\psi_f(x_0, y_0)$ and $f(x_0, y_0)$ represent the content-dependent factors issued from the double integration in the initial system of equations (7.10). We derive them from the Eq. (7.18), since all the remaining quantities there are known:

$$\phi_f(x_0, y_0) = 2 \frac{J_x(x_0 + 1, y_0) - J_x(x_0, y_0)}{2x_0 + 1}, \quad (7.19)$$

$$\psi_f(x_0, y_0) = J_x(x_0, y_0 + 1) - J_x(x_0, y_0), \quad (7.20)$$

$$f(x_0, y_0) = 2 \frac{J_x(x_0 + 1, y_0 + 1) - J_x(x_0 + 1, y_0) - J_x(x_0, y_0 + 1) + J_x(x_0, y_0)}{2x_0 + 1}. \quad (7.21)$$

These expressions are again plugged in Eq. (7.10):

$$\left\{ \begin{array}{l} J_x(x_0 + \alpha, y_0) = J_x(x_0, y_0) + \frac{\alpha(2x_0 + \alpha)}{2x_0 + 1} [J_x(x_0 + 1, y_0) - J_x(x_0, y_0)] \\ J_x(x_0, y_0 + \beta) = J_x(x_0, y_0) + \beta [J_x(x_0, y_0 + 1) - J_x(x_0, y_0)] \\ J_x(x_0 + \alpha, y_0 + \beta) = J_x(x_0 + \alpha, y_0) + J_x(x_0, y_0 + \beta) - J_x(x_0, y_0) + \\ \quad + \alpha \beta \frac{2x_0 + \alpha}{2x_0 + 1} [J_x(x_0 + 1, y_0 + 1) - J_x(x_0 + 1, y_0) - J_x(x_0, y_0 + 1) + J_x(x_0, y_0)]. \end{array} \right. \quad (7.22)$$

We notice that the last equation provides us with the interpolated integral image value: $J(x_0 + \alpha, y_0 + \beta) = J(x, y)$. After plugging the first two equations This equation can finally be rewritten in the following form, representing the bilinear interpolation of $J(x_0, y_0)$, $J(x_0 + 1, y_0)$, $J(x_0, y_0 + 1)$ and $J(x_0 + 1, y_0 + 1)$:

$$J(x, y) = [(1 - \xi)J(x_0, y_0) + \xi J(x_0 + 1, y_0)] (1 - \eta) + \quad (7.23)$$

$$+ [(1 - \xi)J(x_0, y_0 + 1) + \xi J(x_0 + 1, y_0 + 1)] \eta, \quad (7.24)$$

where the weights ξ and η are functions of x , y , α and β . We omitted x in the subscripts, this relation is valid for all the image moments. The resulting weights ξ and η are given in Table 7.1. Using them within Eq. (7.24) allows to compute the filter response for any pixel position and support size (not necessarily aligned with the pixel grid).

Image moment	ξ	η
I_1	α	β
I_x	$\alpha \frac{2x_0 + \alpha}{2x_0 + 1}$	β
I_y	α	$\beta \frac{2y_0 + \beta}{2y_0 + 1}$
I_{x^2}	$\alpha \frac{3\alpha x_0 + 3x_0^2 + \alpha^2}{3x_0^2 + 3x_0 + 1}$	β
I_{y^2}	α	$\beta \frac{3\beta y_0 + 3y_0^2 + \beta^2}{3y_0^2 + 3y_0 + 1}$

Table 7.1 – Interpolation coefficients for integral images of the image moments.

7.3.3 Viewpoint-covariant scale space approximation and detector

Keeping in mind the goal of improved keypoint repeatability under viewpoint position changes, we apply the designed filter to construct an efficient multiscale representation and an associated keypoint detector.

A scale space is typically engendered by progressively applying an image smoothing operator to the input image. In Chapter 6, we formalized the scale space through a diffusion process on a manifold, whose internal metric is given by the depth map, and the texture represents the initial data. Although that approach proves to satisfy scale space requirements, it has the drawback of requiring a computationally expensive iterative simulation of the diffusion process. In this work, we propose a simpler and faster approximation, which adapts the intensity of smoothing locally by using depth information.

The proposed scale space approximation is based on the following observation. A texture image corresponds to the projection of objects in the scene onto the camera plane, followed by a sampling at the pixel granularity. As this sampling is uniform on the camera plane, the scene surfaces are sampled non uniformly. Similarly to [102], we leverage this simple observation to construct an approximated scale space, by varying locally the amount of smoothing, i.e., we vary the smoothing quantity from pixel to pixel as a function of the distance given by the depth map, so that *the further a given pixel is, the less it is smoothed*. More precisely, assume that we can smooth the input image up to a given smoothing quantity $\sigma(x, y)$ at each pixel location (x, y) . Then, let us assume that $\sigma(x, y)$ depends on the depth map $D(x, y)$ in the following way:

$$\sigma(x, y) = \frac{\hat{\sigma}}{D(x, y)}. \quad (7.25)$$

$\hat{\sigma}$ is a constant value (in x, y image variables) representing a scale on the surface, or *spatial scale*, whereas σ is the corresponding scale in the image plane or *projected scale*. Using the pinhole camera model, it is straightforward to show that the projection on the camera

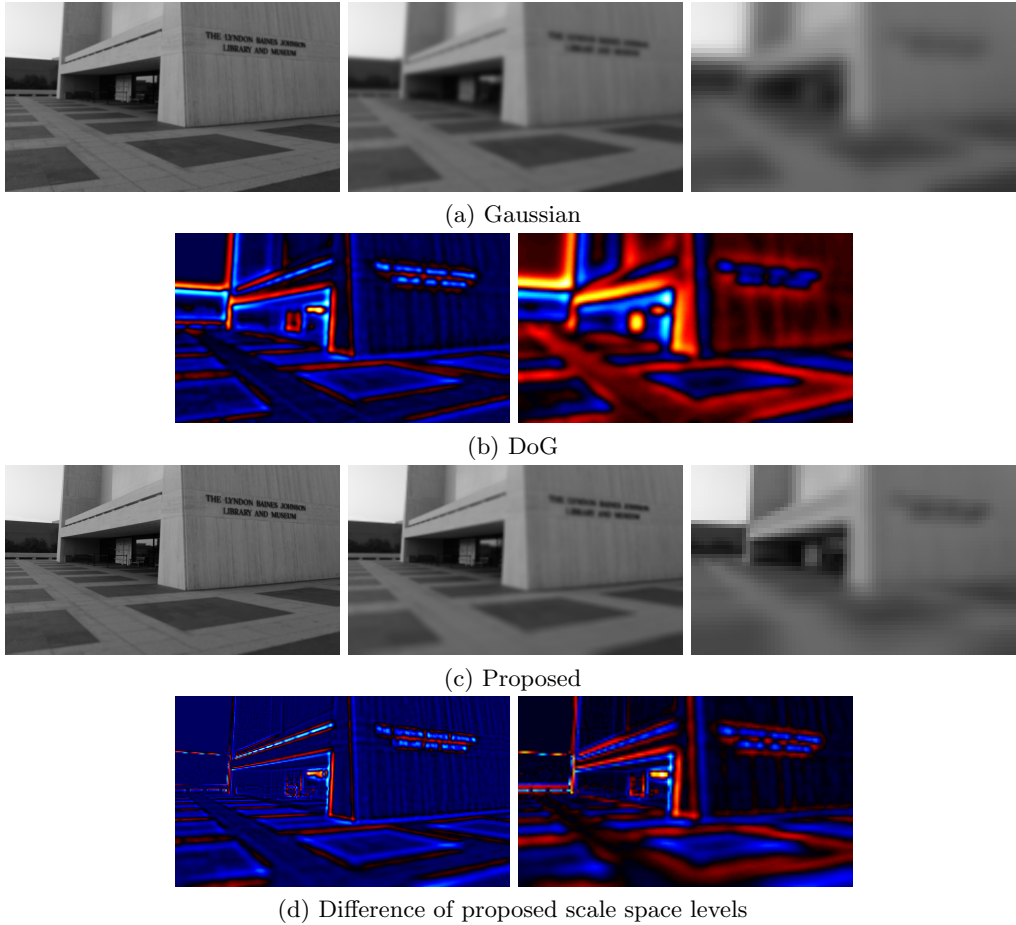


Figure 7.3 – Qualitative comparison of the Gaussian and the proposed multiscale representations for an RGBD image from the *LIVE1* dataset. In the standard Gaussian scale space σ is constant within each image. In the proposed multiscale representation σ varies, but $\hat{\sigma}$ remains constant. Images in rows (b) and (d) obtained by subtracting adjacent smoothed images from rows (a) and (c).

plane of an object of characteristic spatial size $\hat{\sigma}$ is of size $\frac{\hat{\sigma}}{D(x,y)}$ pixels independently on the observer position. Thus, the smoothing quantity $\sigma(x,y)$ injected into the image becomes related to the surface and varies accordingly when the camera moves.

By progressively smoothing the original texture image I using a set $\{\hat{\sigma}_k\}_k$ of increasing $\hat{\sigma}$ values, it is possible to build a multiscale representation $I_k = I(\hat{\sigma}_k)$ that demonstrates the described approximated viewpoint-covariant behavior. The choice of $\hat{\sigma}_k$ and the structure of I_k are discussed below. A visual example is given in Fig. 7.3 (a,c). The desired viewpoint-covariant behavior of the proposed approach might be observed on larger scales.

Once the multiscale representation is defined, the remaining part of keypoint detection consists in three steps: candidates selection, candidates filtering, and accurate localization. We reuse the detection scheme described in 6.3. Specifically, candidates selection is performed as follows:

1. The detection begins by setting $\hat{\sigma} = \hat{\sigma}_0$, a parameter tuned manually as a function of

the depth measurement unit, which mainly depends on the used depth sensor (e.g., Kinect or LIDAR).

2. Next, we construct the pyramidal image representation analogous to [50], conventionally used in numerous scale-invariant detectors, including SIFT. It is based on a combination of smoothing and subsampling steps: we compute $I(\hat{\sigma}_k)$ and $I(2\hat{\sigma}_k)$ as described in the previous section and then downsample the input image by a factor of two horizontally and vertically. Here, k is an integer representing the octave index (counted from zero), and $\hat{\sigma}_k = 2^k \hat{\sigma}_0$. The smoothing filter we use allows to avoid explicit downsampling of the texture image, however, the depth map is properly filtered and resampled to avoid aliasing.
3. Finally, we compute the differences $J_k = I(2\hat{\sigma}_k) - I(\hat{\sigma}_k)$, which are analogous to DoG. It is known that local extrema of DoG reveal visual details of different scales and are repeatable under various transformations [49]. Based on this, our technique consists in taking the local extrema of J_k that should reveal visual details of a given spatial scale in octave k . An illustration is given in Fig. 7.3 (b,d). Distinctive red and blue blobs in the example images contain local maxima and minima of J_k that are taken as initial candidates.
4. The same accurate localization procedure is run for each detected keypoint candidate, as described in Section 6.3.

Keypoints detected on all octaves are put together and sent to the detector output. Each keypoint is thus characterized by its location on the image plane and its visual scale σ obtained according to Eq. (7.25) and interpolated properly after the accurate localization process.

7.4 Experiments and discussion

In this section we compare the proposed approach notably to the original box filter. Our test data consists of several natural images acquired with a DSLR camera. In all our experiments, the box filter support size is defined as 2.6σ , which is a known convention when one tries to approximate the Gaussian. [46] As for the implementations of SIFT features and the separable convolution with Gaussian kernel, we take *VLFeat* library. [80]

7.4.1 Qualitative assessment

We first evaluate the proposed approach visually. Some filtered images are presented in Fig. 7.4. “Phantom contours” along the real edges in the box filter output appear due to the kernel discontinuity near the support boundaries; their displacement is therefore related to the support size s . Typically, this kind of visual artifacts of low-pass filtering are

undesirable. As the proposed filter has continuous falloff to the corners of Ω , the "phantom contours" are attenuated in the resulting image. However, they do not disappear completely as some discontinuities are present near middle points on the sides of the square support region (see Fig. 7.1).



Figure 7.4 – An input image fragment of 600*600 pixels and filter outputs for $\sigma = 20$.

7.4.2 Computational time

In this section we evaluate the computational time of the proposed approach. The results for different values of σ and different input image resolutions are presented in Table 7.2. For each filter we present the smoothing time t_S and the initialization time t_I . The latter comprises the integral images computation.

7.4.3 Rotational invariance compared to Gaussian scale space

As it is discussed before, the invariance of the filter response to image rotations is a very desirable property in some cases. In this experiment we study the rotational invariance of

¹To have a fair comparison, we disabled the use of SSE2 instructions in *VLFeat* library in these tests.

Image resolution	σ	Gauss. ¹	Box		Proposed	
		t_S , ms	t_I , ms	t_S , ms	$t_{I.}$, ms	t_S , ms
1.1 Mpix	1	86	13	34	34	124
	2	134				
	4	237				
	8	433				
	16	865				
4.5 Mpix	1	418	49	135	128	494
	2	611				
	4	994				
	8	1823				
	16	3543				
10.1 Mpix	1	968	110	300	282	1102
	2	1381				
	4	2264				
	8	4084				
	16	7905				

Table 7.2 – Computation times for different filters in function of input image resolution and smoothing level σ . Each value is averaged over 10 repetitions. Tested on a Windows 7 machine with 12-core 3.5 GHz Intel Xeon CPU, 16 GB RAM.

our proposed filter. We proceeded as follows:

- an input image H is first smoothed to a level σ and stored to H_0 ,
- H is then rotated by angle a , smoothed with the same value of σ and then rotated back giving H_a ,
- H_0 is finally compared to H_a in a pixelwise manner. The difference between the images is evaluated in terms of PSNR. To avoid sampling artifacts at this point, σ is chosen large enough.

The experiment is repeated for several values of a from a given range.

A filter stable response to the image rotations will imply the rotated smoothed image H_a close to the smoothed image without rotations H_0 . The more invariant a filter is, the closer the two images should be. As it is explained before, the Gaussian filter reveals perfect rotational invariance.

We perform this experiment on a set of 5 different images (indoor and outdoor photos of 4.5 megapixels), smoothing each of them with 3 values of σ : 5, 10 and 20, resulting in 15 cases per each value of the rotation angle a . The averaged results in function of a are presented in Fig. 7.5. The proposed filter clearly demonstrates a more stable response to image rotations than the box filter. Moreover, obtained PSNR values justify the visual similarity between the Gaussian and the proposed filter, which is illustrated in Fig. 7.4.

We then test the proposed filter in a keypoint detection scenario. Similarly to [130, 131] we replace the Gaussian scale space in the SIFT keypoint detector by image pyramids generated using our proposed filter and the box filter. Following a classic local feature

evaluation procedure [20], we study the repeatability of detected local features obtained with different smoothing operators by means of *matching score*, i.e. ratio between the number of correctly matched features between two given images and the minimal number of detected features for these two images. We evaluate the features repeatability with respect to the in-plane image rotations. The resulting matching scores obtained with several different image sequences are presented on Fig. 7.6.

The proposed filter demonstrates stable repeatability gain with respect to the box filtering. The performance could be further improved by properly tuning the detector to the filter output.

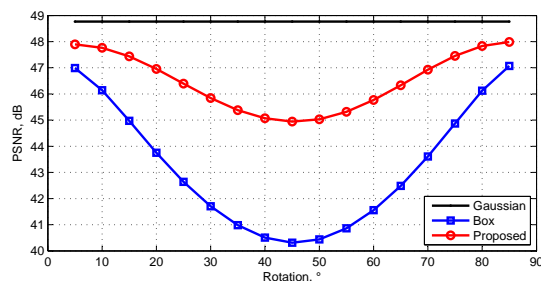


Figure 7.5 – Rotational invariance of the proposed filter vs the box filter. An image is rotated, smoothed and rotated back. The result is then compared to an image smoothed without rotations. Averaged results of 5 images and 3 levels of σ per each rotation angle are shown.

7.4.4 Viewpoint covariant multiscale representation

Following the mid-level feature evaluation protocol described in Section 2.5.1, we test the viewpoint-covariant multiscale approximation and the detector based on it proposed in Section 7.3.3.

The evaluation is performed on two artificial RGBD sequences both containing significant viewpoint position and scale changes: *Bricks* with 20 images and *House* with 25 images (see Section 2.5.2.1). The proposed detector performance is compared to SIFT (VLFeat [80] implementation) and VIP (original implementation). The resulting repeatability scores as a function of the angle of view difference between test and reference views are shown in Fig. 7.7.

The proposed detector demonstrates better overall repeatability except for large scale changes in *House* sequence, where SIFT performance is about 10 points better. A particularly higher accuracy is achieved on *Bricks*, as for tighter η our proposed detector demonstrates a significant gain up to 45° of rotation. At larger angles, the proposed scheme is outperformed by VIP on *Bricks* and SIFT on *House*, but the difference is at most 10 points. Furthermore, VIP detector fails on *House* sequence due to a more complex geometry, that may not be represented well by dominant planes. It is worth noticing that the number of detected keypoints by SIFT and the proposed detector are comparable and of order of 1000 to 2000,

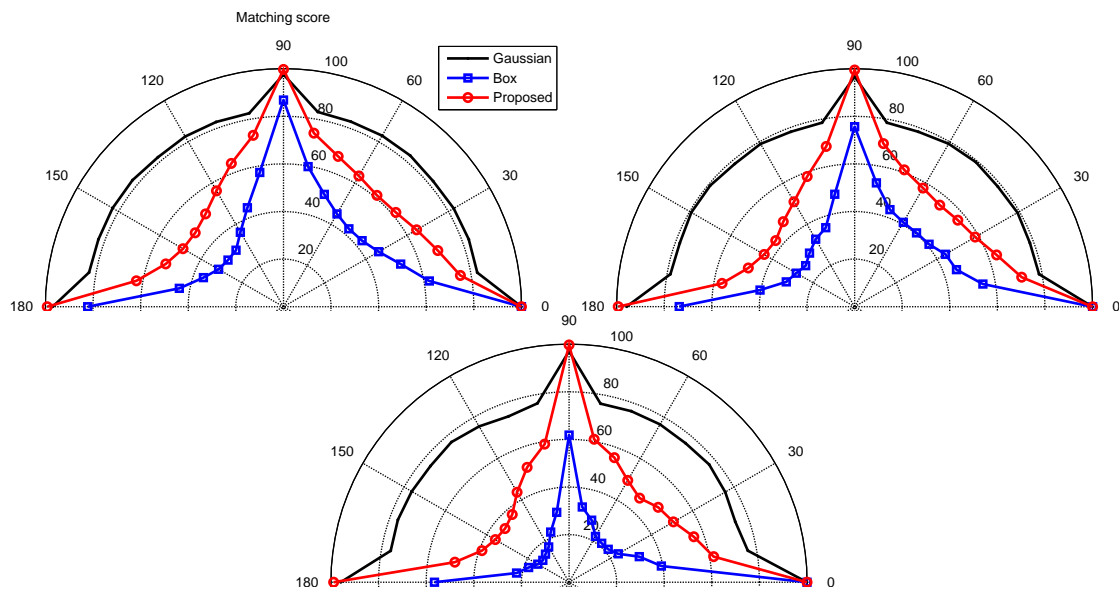


Figure 7.6 – Matching scores subject to in-plane rotations achieved by SIFT features detected with different filters. Three different image sequences of 1296*864 pixels representing in-plane rotations are matched against the corresponding upright images.

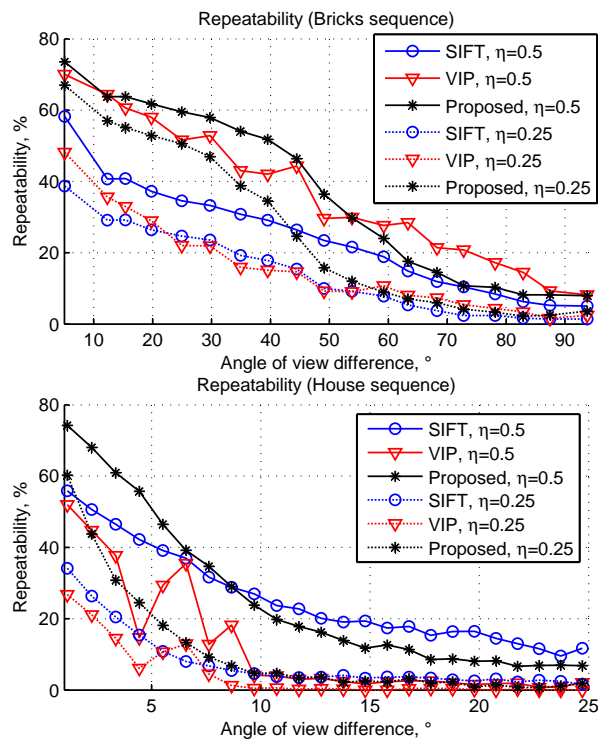


Figure 7.7 – Keypoint repeatability obtained with different detectors on two synthetic RGBD sequences.

whereas VIP exhibits 2 or 3 times more keypoints. A visual example of repeated keypoints is given in Fig. 7.8.

7.5 Conclusion

In this chapter we have proposed an efficient and accurate image smoothing operator providing a good trade-off between fast box filtering and classic Gaussian smoothing. Based on the integral images, our proposed filter inherits the ability of the box filter to compute the response in a constant time at any given point (x, y, σ) . This makes the approach particularly useful not only in feature matching applications, but also in cases where a non-structured smoothing is required, for example a non-uniform spatially adaptive filtering. A viewpoint-covariant multiscale representation and a keypoint detector based on such a filtering process are also proposed and tested, demonstrating improved keypoint repeatability with respect to viewpoint position changes.

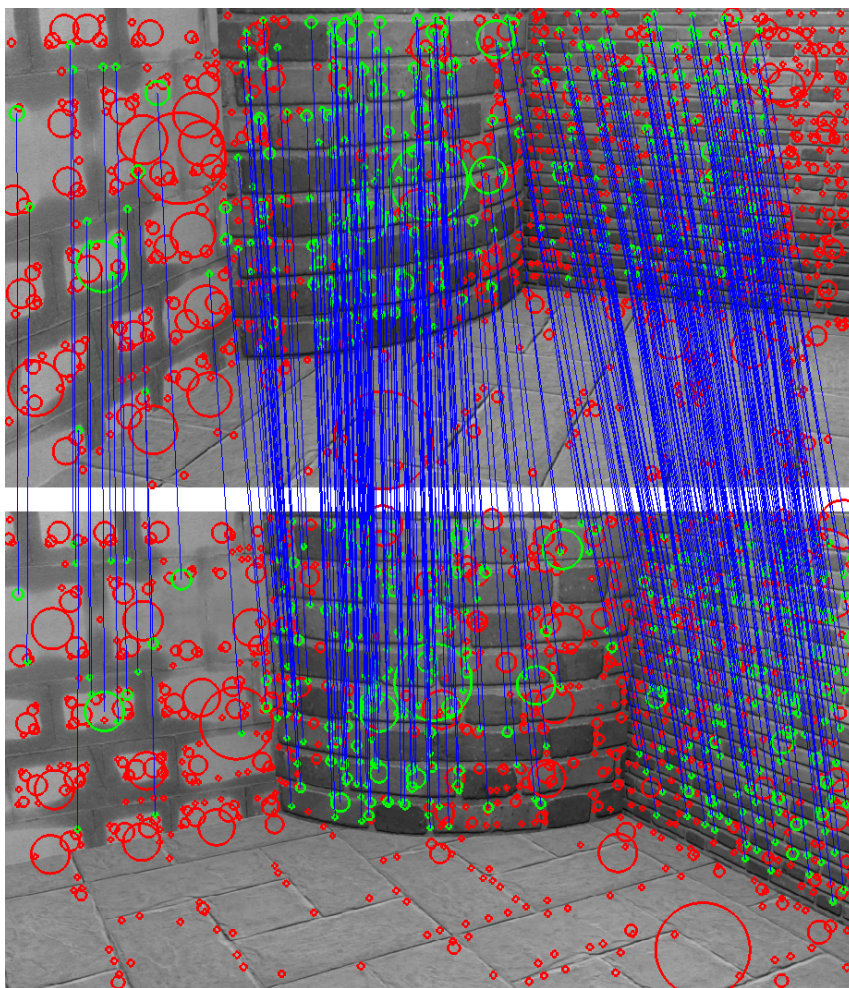


Figure 7.8 – Repeated keypoints of the proposed detector in two views of *Bricks* scene. 1257 and 1109 keypoints are detected in each image, 34.9% repeated for $\eta = 0.5$.

Chapter 8

Conclusion

8.1 Summary

In this thesis we have proposed a set of tools for local visual features extraction from RGBD (texture+depth) images. Based on a systematic study of different visual deformation classes from the local features extraction point of view, we have first identified the most challenging and important geometric deformation for the local features in traditional imaging, that we refer to as *viewpoint position changes*. Including arbitrary movements of the camera, this term combines in-plane translations, rotations and scale changes covered by conventional local features with perspective distortions caused by out-of-plane rotations. The feature locality aspect allows to consider these deformations as occurring locally in the scene, which renders them equivalent to *rigid 3D deformations* of the observed objects.

The robustness of local features to this kind of visual content deformations is of interest in the majority of practical scenarios based on image matching, where no constraints can be imposed a priori on the evolution of the observer position and orientation within the surrounding. At the same time, as we have seen from the analysis of the state of the art, traditional local features often demonstrate limited robustness to this class of visual distortions. However, when the photometric information sent to the input of a feature extraction algorithm is complemented by a geometrical description of the scene, the visual features present in the photometric part might be discovered in a more robust way to the changes in the viewing position. In this thesis, we have proposed several techniques that attempt to involve the depth map in a way to render the texture features more robust to viewpoint position changes.

We have used different experimental settings to evaluate our proposed approaches. A standard feature repeatability and discriminability evaluation procedure was revisited taking into account the complementary depth information (Section 2.5.1), and has then been used in the experimental validation of all the proposed contributions. To assess the performance of the proposed feature extraction techniques in realistic conditions, we tested visual odometry and scene recognition tasks on real images acquired with Kinect

RGBD sensors (experimental parts in Chapters 5, 6 and 7). A number of state-of-the-art techniques of keypoint detection and feature extraction, both from traditional and texture+depth imaging, have been involved in the experiments to provide the performance baseline [22, 25, 46, 82, 100, 113].

Specifically, the following contributions are detailed in this thesis.

1. We investigated how a local planar normalization of texture descriptor patches impacts the performance of local features under out-of-plane rotations. Several such techniques have been proposed in the literature [82, 101, 102], allowing to compensate for the perspective distortions in the descriptor computation stage, and to render the resulting local feature more stable under perspective distortions. We developed an alternative deterministic derivative-free approach of local planar normalization and tested it within the conventional SIFT [22] and BRISK [46] descriptors, analyzing the discriminability of the resulting features [C1].
 2. We developed a generic technique of binary descriptor pattern mapping onto the scene surface defined by the depth map, allowing to render a surface-intrinsic binary feature describing the photometric information given by the texture map. The proposed approach is based on a parametrization of a binary descriptor sampling pattern through the geodesic distance on the surface. This technique is tested within BRISK binary descriptor [46] and compared to other texture-only descriptors and a texture+depth binary descriptor [C2].
 3. We designed a complete feature extraction pipeline for texture+depth content, dubbed TRISK. The proposed approach consists of a corner detector and a binary descriptor, both using the depth information to extract keypoints and their binary signatures from the texture map, in such a way to be robust to arbitrarily complex viewpoint changes. The proposed detector is an extended version of AGAST [45], a recent corner detection approach, and uses the depth information to correct perspective distortions when performing the corner test. The designed pipeline is evaluated in two scenarios, including a visual odometry experiment involving a popular RGBD dataset acquired with Kinect sensor [C8].
 4. We then addressed blob detection in texture+depth images, proposing a scale space formulation for the texture image that exploits the surface metric given by the depth map. This is done by means of a Laplacian-like operator, defining a non-uniform diffusion process for the texture image. We showed that the proposed diffusion process satisfies the maximum principle, which implies that the resulting multiscale representation is causal. This legitimates the use of the proposed scale space in the context of keypoint detection. We also showed experimentally that the proposed multiscale representation exhibits a *viewpoint-covariant behavior*, i.e., the response of the filter engendering the scale space remains stable in a given physical point on the
-

scene surface when the camera position changes. Moreover, the proposed diffusion is performed with a stable numerical scheme. We implemented it using GPU, which allowed substantial saving of computational time [C3, C7].

5. Based on the proposed scale space, we designed a multiscale blob detector for texture+depth images. The proposed approach allows for a substantially more robust keypoint detection under viewpoint position changes, while remaining covariant to other conventional visual deformations (translation, in-plane rotation and scale). The proposed detector is tested in conjunction with conventional descriptors in two different scenarios, including a scene recognition application involving an RGBD dataset acquired with Kinect 2 sensor [C7].
6. We proposed an image smoothing operator, suitable for efficient and accurate spatially varying smoothing. Based on the integral images technique [68] and having the same computational complexity as the popular box filter, the proposed smoothing filter is more accurate both visually and from the rotation invariance point of view, due to compensated discontinuities on boundaries of its convolutional kernel. We tested its performance in several different cases, including a simple scenario of rotational-covariant keypoint detection in conventional images, comparing it to the efficient box filter and to the accurate Gaussian filter, both extensively used in different keypoint detectors [C4].
7. Using the proposed smoothing operator, we designed an alternative multiscale keypoint detector for texture+depth images, which also performs well under significant changes in viewpoint and requires an affordable computational effort, since it does not need to simulate a diffusion process [C5].

The corresponding published articles are reported in List of Publications.

8.2 Future research directions

The increasing availability of RGBD content in different application areas poses new challenges and opens up new perspectives for future research. In this concluding section, we describe several possible extensions of this thesis. We begin with other feature invariance classes from our initial classification in Table 2.1 that are relevant to be addressed in the context of RGBD feature representations.

Invariance by design to unconstrained local rigid scene deformation

As we have seen in some of the experiments (e.g., in Fig. 5.5 and 6.6), involving the depth map into the keypoint detection and/or descriptor computation allows to get close to the *invariance by design* to viewpoint position changes, as it is defined in Section 2.1. However,

there are still some cases where the ultimate goal of designing local features that are fully invariant by design to arbitrary viewpoint position changes and, consequently, local rigid content deformations is still challenging.

The design principles we have used exhibit several limitations with respect to the characteristics of the content given in input, that remain to be overcome in future. Local planar normalization of descriptor patches does not generally perform well in presence of fine details in the observed surface. As we have shown, such a technique is an attractive trade-off that can significantly increase the discriminability of resulting features with a modest computational effort, but may cause a loss of their repeatability. The alternative normalization technique not limited to planar or very smooth surfaces developed in Chapter 4 is computationally demanding and may be sensitive to the depth map quality. Hence, the design of alternative efficient and reliable descriptor patch normalization techniques is considered as a perspective towards the invariance by design. As a concrete example, a computationally efficient fitting of a finite-order smooth surface for blob-like keypoints, or a piecewise constant surface for corners might serve as a basis for the descriptor stabilization allowing further performance improvements.

Invariance to isometric and general elastic surface deformations

Another possible extension of the contributions proposed in this thesis is to address further geometrical visual deformations **G-VI** and **G-VII** in Table 2.1. While these classes are less frequent in practice compared to the viewpoint changes, there are several approaches in the literature addressing such a kind of visual distortions for RGB and RGBD images [98, 140, 141]. Similarly to the viewpoint changes and rigid scene deformations, the depth information seems a valuable complement to the texture map to render invariant features in presence of non-rigid deformations.

As a more concrete example, the descriptor normalization technique proposed in Chapter 4 has been designed without any rigidity constraints, which means that it could be useful to match surfaces under non-rigid isometric distortions (**G-VI**). However, the feature invariance in this case is harder to assess experimentally. For this reason we leave this for future research.

Invariance to illumination conditions

Lighting condition changes are arguably the most challenging and practically interesting class of visual distortions amongst the photometric transformations in Table 2.1. This might include a very broad class of different effects in which the illumination impacts the visual feature detection: shadows, surface specularities, white balance, dynamic range issues, etc. While there is an ongoing work concerning the feature stability under different lighting for conventional images [140, 142] and in High Dynamic Range (HDR) imaging [143, 144], the geometrical information in RGBD images can be helpful when dealing with illumination

changes, as the geometry is by definition invariant to any kind of photometric changes. However, the illumination changes in images reveal a composite phenomenon, and involving the depth information into feature extraction to cope with the illumination changes is not a trivial task.

Together with further feature invariance classes in the classification given in Table 2.1, several broader perspectives for future work on local features in RGBD imaging might be outlined.

Cross-format interoperability of features

In some application areas, such as visual search, feature representations allowing to match instances of different kinds of content, e.g., RGBD images against multi-view content or meshes, would be beneficial. In particular, in RGBD imaging the following question deserves a discussion: how can we match an RGBD image against an RGB image or vice versa, i.e., how the depth map can be used in the matching process if it is available only in the reference or only in the test image? In most, if not all, practical scenarios, despite the increasing availability of RGBD images, the conventional RGB content will still remain dominating. From this perspective, the ability to involve the complementary geometrical information in the matching process, even if it is present only on one side, is of a high practical interest.

Involving temporal dimension

Since the depth map might be viewed as a complementary information providing an additional dimension, it is natural to consider the temporary dimension as one more additional input dimension in video-based representations. Depth maps are often considered as 2.5D images, since they describe 2D manifolds embedded in 3D space, and texture maps are then functions defined on these manifolds. From this perspective, adding the temporal dimension allows for a “3.5D” content representation. Going beyond the depth maps we can consider more general 4D content representations. While the temporal dimension has been involved in the feature extraction both on the detection and the description stages [105, 145] and used to compress local features extracted from video [146], such “3.5D” and 4D representations has not yet been studied.

Using machine learning techniques instead the handcrafted design

Recent progress in machine learning and its applications in various problems of vision have revealed substantial advantages of automatically learned feature representations over the manually designed ones [147]. Machine learning techniques have already been successfully applied in the image matching pipeline to boost the detector repeatability [142] and the descriptor discriminability [41, 73–75]. As for RGBD content, its increasing availability opens

up new perspectives for different machine learning models in the context of RGBD image matching, since the amount of training data grows with an increasing speed. Therefore, learning of appropriate feature representations robust under different geometrical and photometrical informations from RGBD images is another promising direction for future research.

List of Publications

- [C1] M. Karpushin, G. Valenzise, and F. Dufaux, “Local visual features extraction from texture+depth content based on depth image analysis,” in *Proceed. of IEEE Intern. Conf. Image Proc.*, (Paris, France), October 2014.
- [C2] M. Karpushin, G. Valenzise, and F. Dufaux, “Improving distinctiveness of BRISK features using depth maps,” in *Proceed. of IEEE Intern. Conf. Image Proc.*, (Québec city, Canada), September 2015.
- [C3] M. Karpushin, G. Valenzise, and F. Dufaux, “A scale space for texture+depth images based on a discrete Laplacian operator,” in *IEEE Intern. Conf. on Multimedia and Expo*, (Torino, Italy), July 2015.
- [C4] M. Karpushin, G. Valenzise, and F. Dufaux, “An image smoothing operator for fast and accurate scale space approximation,” in *Proceed. of IEEE Intern. Conf. Acoust., Speech and Sign. Proc.*, (Shanghai, China), March 2016.
- [C5] M. Karpushin, G. Valenzise, and F. Dufaux, “Keypoint detection in RGBD images based on an efficient viewpoint-covariant multiscale representation,” in *Proceed. of Europ. Sign. Proc. Conf.*, (Budapest, Hungary), EURASIP, August 2016.
- [C6] M. Karpushin, G. Valenzise, and F. Dufaux, “Good Features to Track for RGBD images,” in *Proceed. of IEEE Intern. Conf. Acoust., Speech and Sign. Proc.*, 2017, **submitted**.
- [C7] M. Karpushin, G. Valenzise, and F. Dufaux, “Keypoint detection in RGBD images based on an anisotropic scale space,” *IEEE Trans. Multimedia*, vol. 18, no. 9, pp. 1762 – 1771, 2016.
- [C8] M. Karpushin, G. Valenzise, and F. Dufaux, “TRISK: A local features extraction framework for texture+depth content matching,” *IEEE Trans. Pattern Anal. Machine Intell.*, 2016, **submitted**, currently under revision.
-

Glossary

AGAST Adaptive and Generic Accelerated Segment Test

ALP A Low-degree Polynomial

AST Accelerated Segment Test

BRAND Binary Robust Appearance and Normal Descriptor

BRIEF Binary Robust Independent Elementary Feature

BRISK Binary Robust Invariant Scalable Keypoints

BRISKOLA BRISK Optimized for Low-power ARM

CARD Compact and Real-time Descriptor

CDVA Compact Descriptors for Video Analysis

CDVS Compact Descriptors for Visual Search

CSHOT Color SHOT

DAFT Depth-Adaptive Feature Transform

DoG Difference-of-Gaussians

FAIR-SURF Fully Affine Invariant SURF

FAST Features from Accelerated Segment Test

FINDDD Fast Integral Normal 3D

FPFH Fast Point Feature Histogram

FREAK Fast REtinA Keypoint

GFTT Good Features to Track

GLOH Gradient Location and Orientation Histogram

HDR High Dynamic Range

HOG Histogram of Oriented Gradients

LATCH Learned Arrangements of Three patCH codes

LD-SIFT Local Depth SIFT

LoG Laplacian of Gaussian

Mesh-LBP Mesh Local Binary Patterns

MSER Maximally Stable Extremal Region

NARF Normal Aligned Radial Feature

ORB Oriented FAST and Rotated BRIEF

PCA principal component analysis

PDE partial differential equation

PFH Point Feature Histogram

PIN Perspectively Invariant Normal features

RANSAC random sample consensus

ROC receiver operating characteristics

SAR synthetic aperture radar

SHOT Signature of Histograms of Orientations

SIFT Scale Invariant Feature Transform

SLAM simultaneous localization and mapping

SURF Speeded Up Robust Features

SUSAN the Smallest Univalve Segment Assimilating Nucleus

VIP Viewpoint Invariant Patches

Bibliography

- [1] L. Gomes, “When will Google’s self-driving car really be ready? It depends on where you live and what you mean by ”ready”,” *IEEE Spectrum*, vol. 53, no. 5, pp. 13–14, 2016.
 - [2] A. Geiger, P. Lenz, and R. Urtasun, “Are we ready for autonomous driving? The KITTI vision benchmark suite,” in *Proceed. of IEEE Intern. Conf. on Comp. Vision and Pattern Rec.*, Providence, Rhode Island, USA, June 2012.
 - [3] H. Chen, X.-m. Wang, and Y. Li, “A survey of autonomous control for UAV,” in *Proceed. of IEEE Intern. Conf. on Artif. Intelligence and Comp. Intelligence*, Shanghai, China, November 2009.
 - [4] A. Gil, O. M. Mozos, M. Ballesta, and O. Reinoso, “A comparative evaluation of interest point detectors and local descriptors for visual SLAM,” *Machine Vision and Applications*, vol. 21, no. 6, pp. 905–920, 2010.
 - [5] C. Schmid and R. Mohr, “Local grayvalue invariants for image retrieval,” *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 19, no. 5, pp. 530–534, 1997.
 - [6] H. Jegou, F. Perronnin, M. Douze, J. Sánchez, P. Perez, and C. Schmid, “Aggregating local image descriptors into compact codes,” *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 34, no. 9, pp. 1704–1716, 2012.
 - [7] S. Husain and M. Bober, “Robust and scalable aggregation of local features for ultra large-scale retrieval,” in *Proceed. of IEEE Intern. Conf. Image Proc.*, Paris, France, 2014.
 - [8] M. Brown and S. Süssstrunk, “Multi-spectral SIFT for scene category recognition,” in *Proceed. of IEEE Intern. Conf. on Comp. Vision and Pattern Rec.*, Colorado Springs, USA, 2011.
 - [9] S. Wang, H. You, and K. Fu, “BFSIFT: A novel method to find feature matches for SAR image registration,” *IEEE Geoscience and Remote Sensing Letters*, vol. 9, no. 4, pp. 649–653, 2012.
 - [10] I. Tosić and K. Berkner, “3D keypoint detection by light field scale-depth space analysis,” in *Proceed. of IEEE Intern. Conf. Image Proc.*, Paris, France, October 2014.
 - [11] A. Kovnatsky, M. M. Bronstein, A. M. Bronstein, and R. Kimmel, “Photometric heat kernel signatures,” in *Proceed. of Intern. Conf. on Scale Space and Variational Methods in Comp. Vision*. Ein-Gedi, Israel: Springer, May 2011.
 - [12] A. Zaharescu, E. Boyer, K. Varanasi, and R. Horaud, “Surface feature detection and description with applications to mesh matching,” in *Proceed. of IEEE Intern. Conf. on Comp. Vision and Pattern Rec.*, Miami, USA, June 2009.
 - [13] B. Steder, R. B. Rusu, K. Konolige, and W. Burgard, “Point feature extraction on 3D range scans taking into account object boundaries,” in *Proceed. of IEEE Intern. Conf. on Rob. and Autom.*, Shanghai, China, May 2011.
-

- [14] T.-W. R. Lo and J. P. Siebert, "Local feature extraction and matching on range images: 2.5D SIFT," *Comp. Vision and Image Understanding*, vol. 113, no. 12, pp. 1235–1250, 2009.
- [15] J. Han, L. Shao, D. Xu, and J. Shotton, "Enhanced computer vision with Microsoft Kinect sensor: A review," *IEEE Trans. on Cybernetics*, vol. 43, no. 5, pp. 1318–1334, 2013.
- [16] H. Gonzalez-Jorge, B. Riveiro, E. Vazquez-Fernandez, J. Martínez-Sánchez, and P. Arias, "Metrological evaluation of Microsoft Kinect and ASUS Xtion sensors," *Measurement*, vol. 46, no. 6, pp. 1800–1806, 2013.
- [17] F. Dufaux, B. Pesquet-Popescu, and M. Cagnazzo, *Emerging Technologies for 3D Video: Creation, Coding, Transmission and Rendering*. John Wiley & Sons, 2013.
- [18] D. Nistér, "An efficient solution to the five-point relative pose problem," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 26, no. 6, pp. 756–770, 2004.
- [19] A. Redondi, L. Baroffio, M. Cesana, and M. Tagliasacchi, "Compress-then-analyze vs. analyze-then-compress: Two paradigms for image analysis in visual sensor networks," in *Proceed. of IEEE Worksh. Multim. Sign. Proc.*, 2013, pp. 278–282.
- [20] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 27, no. 10, pp. 1615–1630, 2005.
- [21] A. Canclini, M. Cesana, A. Redondi, M. Tagliasacchi, J. Ascenso, and R. Cilla, "Evaluation of low-complexity visual feature detectors and descriptors," in *Proceed. of IEEE Intern. Conf. on Dig. Signal Proc.*, Fira, Santorini, Greece, July 2013.
- [22] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Intern. J. of Comp. Vision*, vol. 60, no. 2, pp. 91–110, 2004.
- [23] J.-M. Morel and G. Yu, "ASIFT: A new framework for fully affine invariant image comparison," *SIAM Journal on Imaging Sciences*, vol. 2, no. 2, pp. 438–469, 2009.
- [24] P. Moreels and P. Perona, "Evaluation of features detectors and descriptors based on 3D objects," *Intern. J. of Comp. Vision*, vol. 73, no. 3, pp. 263–284, 2007.
- [25] K. Mikolajczyk and C. Schmid, "An affine invariant interest point detector," in *Proceed. of Europ. Conf. on Comp. Vision*. Copenhagen, Denmark: Springer, May 2002.
- [26] F. Fraundorfer and H. Bischof, "A novel performance evaluation method of local detectors on non-planar scenes," in *Proceed. of IEEE Intern. Conf. on Comp. Vision and Pattern Rec.*, San Diego, USA, May 2005.
- [27] Z. Wang, L.-Y. Duan, J. Lin, T. Huang, W. Gao, and M. Bober, "Component hashing of variable-length binary aggregated descriptors for fast image search," in *Proceed. of IEEE Intern. Conf. Image Proc.*, Paris, France, 2014.
- [28] Z. Liu, H. Li, W. Zhou, R. Hong, and Q. Tian, "Uniting keypoints: Local visual information fusion for large-scale image search," *IEEE Trans. Multimedia*, vol. 17, no. 4, pp. 538–548, 2015.
- [29] U. L. Altintakan and A. Yazici, "Towards effective image classification using class-specific codebooks and distinctive local features," *IEEE Trans. Multimedia*, vol. 17, no. 3, pp. 323–332, 2015.
- [30] B. Kitt, A. Geiger, and H. Latégahn, "Visual odometry based on stereo image sequences with RANSAC-based outlier rejection scheme," in *Proceed. of IEEE Intelligent Vehicles Symposium*, San Diego, CA, USA, 2010.

- [31] F. Endres, J. Hess, N. Engelhard, J. Sturm, D. Cremers, and W. Burgard, "An evaluation of the RGB-D SLAM system," in *Proceed. of IEEE Intern. Conf. on Rob. and Autom.*, St. Paul, MN, USA, May 2012.
- [32] S. N. Sinha, J.-M. Frahm, M. Pollefeys, and Y. Genc, "Feature tracking and matching in video using programmable graphics hardware," *Machine Vision and Applications*, vol. 22, no. 1, pp. 207–217, 2011.
- [33] H. Fradi and J.-L. Dugelay, "Spatial and temporal variations of feature tracks for crowd behavior analysis," *Journal on Multimodal User Interfaces*, pp. 1–11, 2015.
- [34] J. Heinly, E. Dunn, and J.-M. Frahm, "Comparative evaluation of binary features," in *Proceed. of Europ. Conf. on Comp. Vision*. Firenze, Italy: Springer, October 2012.
- [35] Y. Guo, M. Bennamoun, F. Sohel, M. Lu, J. Wan, and N. M. Kwok, "A comprehensive performance evaluation of 3d local feature descriptors," *Intern. J. of Comp. Vision*, pp. 1–24, 2015.
- [36] D. Mukherjee, Q. J. Wu, and G. Wang, "A comparative experimental study of image feature detectors and descriptors," *Machine Vision and Applications*, vol. 26, no. 4, pp. 443–466, 2015.
- [37] ISO/IEC JTC 1/SC 29/ WG 11, "ISO/IEC CD 15938-13 compact descriptors for visual search," ISO/IEC, Sapporo, Japan, MPEG document N14681, July 2014.
- [38] —, "CDVA: Requirements," ISO/IEC, Valencia, Spain, MPEG document N14509, March 2014.
- [39] C. Harris and M. Stephens, "A combined corner and edge detector." in *Alvey vision conference*, vol. 15, Manchester, UK, 1988, p. 50.
- [40] H. P. Moravec, "Obstacle avoidance and navigation in the real world by a seeing robot rover." DTIC Document, Tech. Rep., 1980.
- [41] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: an efficient alternative to SIFT or SURF," in *Proceed. of IEEE Intern. Conf. on Comp. Vision*, Barcelona, Spain, November 2011.
- [42] J. Shi and C. Tomasi, "Good features to track," in *Proceed. of IEEE Intern. Conf. on Comp. Vision and Pattern Rec.*, Seattle WA, USA, June 1994.
- [43] S. M. Smith and J. M. Brady, "SUSAN – a new approach to low level image processing," *Intern. J. of Comp. Vision*, vol. 23, no. 1, pp. 45–78, 1997.
- [44] E. Rosten and T. Drummond, "Fusing points and lines for high performance tracking," in *Proceed. of IEEE Intern. Conf. on Comp. Vision*, Beijing, China, October 2005.
- [45] E. Mair, G. D. Hager, D. Burschka, M. Suppa, and G. Hirzinger, "Adaptive and generic corner detection based on the accelerated segment test," in *Proceed. of Europ. Conf. on Comp. Vision*. Crete, Greece: Springer, September 2010.
- [46] S. Leutenegger, M. Chli, and R. Y. Siegwart, "BRISK: Binary robust invariant scalable keypoints," in *Proceed. of IEEE Intern. Conf. on Comp. Vision*, Barcelona, Spain, November 2011.
- [47] T. Lindeberg, "Scale-space theory: A basic tool for analyzing structures at different scales," *Journal of applied statistics*, vol. 21, no. 1-2, pp. 225–270, 1994.

- [48] J. J. Koenderink, "The structure of images," *Biological cybernetics*, vol. 50, no. 5, pp. 363–370, 1984.
- [49] T. Lindeberg, "Feature detection with automatic scale selection," *Intern. J. of Comp. Vision*, vol. 30, no. 2, pp. 79–116, 1998.
- [50] T. Lindeberg and L. Bretzner, "Real-time scale selection in hybrid multi-scale representations," *Scale Space Methods in Computer Vision*, pp. 148–163, 2003.
- [51] K. Mikolajczyk and C. Schmid, "Indexing based on scale invariant interest points," in *Proceed. of IEEE Intern. Conf. on Comp. Vision*, vol. 1, 2001, pp. 525–531.
- [52] A. E. Johnson and M. Hebert, "Using spin images for efficient object recognition in cluttered 3D scenes," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 21, no. 5, pp. 433–449, 1999.
- [53] S. Lazebnik, C. Schmid, and J. Ponce, "Sparse texture representations using affine-invariant neighborhoods," in *Proceed. of IEEE Intern. Conf. on Comp. Vision and Pattern Rec.*, Madison, Wisconsin, USA, June 2003.
- [54] L. Van Gool, T. Moons, and D. Ungureanu, "Affine/photometric invariants for planar intensity patterns," in *Proceed. of Europ. Conf. on Comp. Vision*. Cambridge, UK: Springer, 1996.
- [55] J. J. Koenderink and A. J. van Doorn, "Representation of local geometry in the visual system," *Biological cybernetics*, vol. 55, no. 6, pp. 367–375, 1987.
- [56] F. Mindru, T. Moons, and L. Van Gool, "Recognizing color patterns irrespective of viewpoint and illumination," in *Proceed. of IEEE Intern. Conf. on Comp. Vision and Pattern Rec.*, 1999.
- [57] M. Varma and A. Zisserman, "Classifying images of materials: Achieving viewpoint and illumination independence," in *Proceed. of Europ. Conf. on Comp. Vision*. Copenhagen, Denmark: Springer, May 2002.
- [58] C. Schmid, "Constructing models for content-based image retrieval," in *Proceed. of IEEE Intern. Conf. on Comp. Vision and Pattern Rec.*, Kauai, HI, USA, December 2001.
- [59] F. Schaffalitzky and A. Zisserman, "Multi-view matching for unordered image sets, or "How do i organize my holiday snaps?"," in *Proceed. of Europ. Conf. on Comp. Vision*, ser. Lecture Notes in Computer Science. Copenhagen, Denmark: Springer, May 2002.
- [60] M. Brown and D. G. Lowe, "Invariant features from interest point groups." in *Proceed. of British Machine Vision Conf.*, Cardiff, UK, September 2002.
- [61] Y. Ke and R. Sukthankar, "PCA-SIFT: A more distinctive representation for local image descriptors," in *Proceed. of IEEE Intern. Conf. on Comp. Vision and Pattern Rec.*, Washington, DC, USA, June 2004.
- [62] S. A. Winder and M. Brown, "Learning local image descriptors," in *Proceed. of IEEE Intern. Conf. on Comp. Vision and Pattern Rec.*, Minneapolis, Minnesota, USA, June 2007.
- [63] B. Rister, G. Wang, M. Wu, and J. R. Cavallaro, "A fast and efficient SIFT detector using the mobile GPU," in *Proceed. of IEEE Intern. Conf. Acoust., Speech and Sign. Proc.*, Vancouver, Canada, May 2013.
- [64] G.-R. Kayombya, "SIFT feature extraction on a smartphone GPU using OpenGL ES 2.0," Ph.D. dissertation, Massachusetts Institute of Technology, 2010.
- [65] C. Wu, "SiftGPU: A GPU implementation of scale invariant feature transform (SIFT)," <http://cs.unc.edu/~ccwu/siftgpu>, 2007.

- [66] H. Fassold and J. Rosner, "A real-time GPU implementation of the SIFT algorithm for large-scale video analysis tasks," in *SPIE/IS&T Electronic Imaging*. International Society for Optics and Photonics, 2015, pp. 940 007–940 007.
- [67] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (SURF)," *Comp. Vision and Image Understanding*, vol. 110, no. 3, pp. 346–359, 2008.
- [68] F. C. Crow, "Summed-area tables for texture mapping," *ACM SIGGRAPH computer graphics*, vol. 18, no. 3, pp. 207–212, 1984.
- [69] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, "BRIEF: Binary robust independent elementary features," in *Proceed. of Europ. Conf. on Comp. Vision*. Crete, Greece: Springer, September 2010.
- [70] T. Ojala, M. Pietikäinen, and D. Harwood, "A comparative study of texture measures with classification based on featured distributions," *Pattern recognition*, vol. 29, no. 1, pp. 51–59, 1996.
- [71] L. Baroffio, A. Canclini, M. Cesana, A. Redondi, and M. Tagliasacchi, "Briskola: BRISK optimized for low-power ARM architectures," in *Proceed. of IEEE Intern. Conf. Image Proc.*, Paris, France, October 2014.
- [72] A. Alahi, R. Ortiz, and P. Vandergheynst, "FREAK: Fast retina keypoint," in *Proceed. of IEEE Intern. Conf. on Comp. Vision and Pattern Rec.*, Providence, Rhode Island, USA, June 2012.
- [73] T. Trzcinski, M. Christoudias, P. Fua, and V. Lepetit, "Boosting binary keypoint descriptors," in *Proceed. of IEEE Intern. Conf. on Comp. Vision and Pattern Rec.*, Portland, Oregon, USA, June 2013.
- [74] M. Ambai and Y. Yoshida, "CARD: Compact and real-time descriptors," in *Proceed. of IEEE Intern. Conf. on Comp. Vision*, Barcelona, Spain, 2011.
- [75] G. Levi and T. Hassner, "LATCH: learned arrangements of three patch codes," in *Proceed. of IEEE Winter Conf. on Applications of Comp. Vision*, Lake Placid, NY, USA, March 2016. [Online]. Available: <http://www.openu.ac.il/home/hassner/projects/LATCH>
- [76] L. Wolf, T. Hassner, and Y. Taigman, "Descriptor based methods in the wild," in *Proceed. of ECCV Workshop on faces in 'real-life' images*. Marseille, France: Springer, 2008.
- [77] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Van Gool, "A comparison of affine region detectors," *Intern. J. of Comp. Vision*, vol. 65, no. 1-2, pp. 43–72, 2005.
- [78] K. Mikolajczyk, C. Schmid *et al.*, "Comparison of affine-invariant local detectors and descriptors," in *Proceed. of Europ. Sign. Proc. Conf.* Vienna, Austria: EURASIP, 2004.
- [79] T. Lindeberg and J. Gårding, "Shape-adapted smoothing in estimation of 3-D shape cues from affine deformations of local 2-D brightness structure," *Image and vision computing*, vol. 15, no. 6, pp. 415–434, 1997.
- [80] A. Vedaldi and B. Fulkerson, "VLFeat: An open and portable library of computer vision algorithms," in *Proceed. of Intern. Conf. on Multimedia*, ser. MM '10. New York, USA: ACM, 2010.
- [81] A. Baumberg, "Reliable feature matching across widely separated views," in *Proceed. of IEEE Intern. Conf. on Comp. Vision and Pattern Rec.*, vol. 1, 2000, pp. 774–781.

- [82] C. Wu, B. Clipp, X. Li, J.-M. Frahm, and M. Pollefeys, “3D model matching with viewpoint-invariant patches (VIP),” in *Proceed. of IEEE Intern. Conf. on Comp. Vision and Pattern Rec.*, Anchorage, Alaska, USA, June 2008.
- [83] Y. Pang, W. Li, Y. Yuan, and J. Pan, “Fully affine invariant SURF for image matching,” *Neurocomputing*, vol. 85, pp. 6–10, 2012.
- [84] ISO/IEC JTC 1/SC 29/ WG 11, “ISO/IEC CD 15938-14 reference software, conformance and usage guidelines for compact descriptors for visual search,” ISO/IEC, Warsaw, Poland, MPEG document N15371, June 2015.
- [85] —, “CDVA: Requirements,” ISO/IEC, Strasbourg, France, MPEG document N15040, October 2014.
- [86] F. Perronnin and C. Dance, “Fisher kernels on visual vocabularies for image categorization,” in *Proceed. of IEEE Intern. Conf. on Comp. Vision and Pattern Rec.*, Minneapolis, Minnesota, USA, June 2007.
- [87] ISO/IEC JTC 1/SC 29/ WG 11, “Compact descriptors for visual search, test model 12,” ISO/IEC, Strasbourg, France, MPEG document N14961, October 2014.
- [88] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in *Proceed. of IEEE Intern. Conf. on Comp. Vision and Pattern Rec.*, San Diego, USA, May 2005.
- [89] T. Hou and H. Qin, “Efficient computation of scale-space features for deformable shape correspondences,” in *Proceed. of Europ. Conf. on Comp. Vision.* Crete, Greece: Springer, September 2010.
- [90] N. Werghi, S. Berretti, and A. Del Bimbo, “The mesh-LBP: a framework for extracting local binary patterns from discrete manifolds,” *IEEE Trans. Image Processing*, vol. 24, pp. 220–235, 2015.
- [91] I. Sipiran and B. Bustos, “Harris 3D: a robust extension of the harris operator for interest point detection on 3D meshes,” *The Visual Computer*, vol. 27, no. 11, pp. 963–976, 2011.
- [92] T. Darom and Y. Keller, “Scale-invariant features for 3-D mesh models,” *IEEE Trans. Image Processing*, vol. 21, no. 5, pp. 2758–2769, 2012.
- [93] Y. Guo, M. Bennamoun, F. Sohel, M. Lu, and J. Wan, “3D object recognition in cluttered scenes with local surface features: A survey,” *IEEE Trans. Image Processing*, vol. 36, pp. 2270–2287, 2014.
- [94] R. B. Rusu, Z. C. Marton, N. Blodow, and M. Beetz, “Persistent point feature histograms for 3D point clouds,” in *Proceed. of Intern. Conf. on Intelligent Autonomous Systems.* Baden-Baden, Germany: IOS Press, July 2008.
- [95] R. B. Rusu, N. Blodow, and M. Beetz, “Fast point feature histograms (FPFH) for 3D registration,” in *Proceed. of IEEE Intern. Conf. on Rob. and Autom.*, Kobe, Japan, May 2009.
- [96] E. Wahl, U. Hillenbrand, and G. Hirzinger, “Surflet-pair-relation histograms: a statistical 3D-shape representation for rapid classification,” in *Proceed. of IEEE Intern. Conf. on 3D Digital Imaging and Modeling*, Banff, Alberta, Canada, October 2003.
- [97] F. Tombari, S. Salti, and L. Di Stefano, “Unique signatures of histograms for local surface description,” in *Proceed. of Europ. Conf. on Comp. Vision.* Crete, Greece: Springer, September 2010.

- [98] A. Ramisa, G. Alenya, F. Moreno-Noguer, and C. Torras, “FINDDD: A fast 3D descriptor to characterize textiles for robot manipulation,” in *Proceed. of IEEE Intern. Conf. on Intellig. Robots and Systems*, Tokyo, Japan, 2013.
- [99] F. Tombari, S. Salti, and L. Di Stefano, “A combined texture-shape descriptor for enhanced 3D feature matching,” in *Proceed. of IEEE Intern. Conf. Image Proc.*, Brussels, Belgium, September 2011.
- [100] E. R. do Nascimento, G. L. Oliveira, A. W. Vieira, and M. F. Campos, “On the development of a robust, fast and lightweight keypoint descriptor,” *Neurocomputing*, vol. 120, pp. 141–155, 2013.
- [101] K. Koser and R. Koch, “Perspectively invariant normal features,” in *Proceed. of IEEE Intern. Conf. on Comp. Vision*, Rio de Janeiro, Brazil, October 2007.
- [102] D. Gossow, D. Weikersdorfer, and M. Beetz, “Distinctive texture features from perspective-invariant keypoints,” in *Proceed. of IEEE Intern. Conf. on Pattern Rec.*, Tsukuba, Japan, November 2012.
- [103] J. Matas, O. Chum, M. Urban, and T. Pajdla, “Robust wide-baseline stereo from maximally stable extremal regions,” *Image and vision computing*, vol. 22, no. 10, pp. 761–767, 2004.
- [104] J. Weickert, *Anisotropic diffusion in image processing*. Teubner Stuttgart, 1998, vol. 1.
- [105] T. Lindeberg, “Generalized gaussian scale-space axiomatics comprising linear scale-space, affine scale-space and spatio-temporal scale-space,” *Journal of Mathematical Imaging and Vision*, vol. 40, no. 1, pp. 36–81, 2011.
- [106] P. Perona and J. Malik, “Scale-space and edge detection using anisotropic diffusion,” *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 12, no. 7, pp. 629–639, 1990.
- [107] C. Tomasi and R. Manduchi, “Bilateral filtering for gray and color images,” in *Proceed. of IEEE Intern. Conf. on Comp. Vision*, Bombay, India, January 1998.
- [108] K. He, J. Sun, and X. Tang, “Guided image filtering,” *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 35, no. 6, pp. 1397–1409, 2013.
- [109] G. Sapiro, *Geometric partial differential equations and image analysis*. Cambridge university press, 2006.
- [110] R. Unnikrishnan and M. Hebert, “Multi-scale interest regions from unorganized point clouds,” in *Proceed. of IEEE Intern. Conf. on Comp. Vision and Pattern Rec. workshops*, Anchorage, Alaska, USA, June 2008.
- [111] J. Digne, J.-M. Morel, C.-M. Souzani, and C. Lartigue, “Scale space meshing of raw data point sets,” in *Computer Graphics Forum*, vol. 30, no. 6. Wiley Online Library, 2011, pp. 1630–1642.
- [112] P. F. Alcantarilla, A. Bartoli, and A. J. Davison, “KAZE features,” in *Proceed. of Europ. Conf. on Comp. Vision*. Florence, Italy: Springer, October 2012.
- [113] M. Agrawal, K. Konolige, and M. R. Blas, “Censure: Center surround extremas for realtime feature detection and matching,” in *Proceed. of Europ. Conf. on Comp. Vision*. Marseille, France: Springer, 2008.
- [114] M. Gohara and D. Suter, “Feature detection with an improved anisotropic filter,” in *Proceed. of Asian Conf. on Comp. Vision*. Hyderabad, India: Springer, January 2006.

- [115] M. Heikkilä, M. Pietikäinen, and C. Schmid, "Description of interest regions with local binary patterns," *Pattern recognition*, vol. 42, no. 3, pp. 425–436, 2009.
- [116] K. Mikolajczyk and C. Schmid, "Scale & affine invariant interest point detectors," *Intern. J. of Comp. Vision*, vol. 60, no. 1, pp. 63–86, 2004.
- [117] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of rgb-d slam systems," in *Proceed. of IEEE Intern. Conf. on Intelligent Robot Systems*, Vilamoura, Algarve, Portugal, October 2012.
- [118] C.-C. Su, L. K. Cormack, and A. C. Bovik, "Color and depth priors in natural images," *IEEE Trans. Image Processing*, vol. 22, no. 6, pp. 2259–2274, 2013.
- [119] C.-C. Su, A. C. Bovik, and L. K. Cormack, "Natural scene statistics of color and range," in *Proceed. of IEEE Intern. Conf. Image Proc.*, Brussels, Belgium, 2011.
- [120] "LIVE1 dataset webpage," http://live.ece.utexas.edu/research/3dnss/live_color_plus_3d.html, accessed in Dec. 2014.
- [121] N. Yokoya and M. D. Levine, "Range image segmentation based on differential geometry: a hybrid approach," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 11, no. 6, pp. 643–649, 1989.
- [122] A. Spira and R. Kimmel, "An efficient solution to the eikonal equation on parametric manifolds," *Interfaces and Free Boundaries*, vol. 6, no. 3, pp. 315–328, 2004.
- [123] I. Mpiperis, S. Malassiotis, and M. G. Strintzis, "3-D face recognition with the geodesic polar representation," *IEEE Trans. Inform. Forensics Sec.*, vol. 2, no. 3, pp. 537–547, 2007.
- [124] R. B. Rusu and S. Cousins, "3D is here: Point cloud library (PCL)," in *Proceed. of IEEE Intern. Conf. on Rob. and Autom.*, Shanghai, China, May 2011.
- [125] P. J. Besl and N. D. McKay, "Method for registration of 3-D shapes," in *Robotics-DL tentative*. International Society for Optics and Photonics, 1992, pp. 586–606.
- [126] G. Xu, "Discrete Laplace–Beltrami operators and their convergence," *Computer Aided Geometric Design*, vol. 21, no. 8, pp. 767–784, 2004.
- [127] J. Digne and J.-M. Morel, "Numerical analysis of differential operators on raw point clouds," *Numerische Mathematik*, vol. 127, no. 2, pp. 255–289, 2014.
- [128] M. Wardetzky, S. Mathur, F. Kälberer, and E. Grinspun, "Discrete Laplace operators: no free lunch," in *Symposium on Geometry processing*, Barcelona, Spain, July 2007.
- [129] F. Hassan, N. Pax, and S. Khorbotly, "Reduced-latency architecture for image smoothing exponential filters," in *Proceed. IEEE Intern. Midwest Symposium on Circuits and Systems*, Fort Collins, Colorado, USA, August 2015.
- [130] M. Grabner, H. Grabner, and H. Bischof, "Fast approximated SIFT," in *Proceed. of Asian Conf. on Comp. Vision*, Hyderabad, India, January 2006.
- [131] L.-C. Chiu, T.-S. Chang, J.-Y. Chen, and N. Y.-C. Chang, "Fast SIFT design for real-time visual feature extraction," *Proceed. of IEEE Intern. Conf. Image Proc.*, September 2013.
- [132] K. G. Derpanis, E. T. Leung, and M. Sizintsev, "Fast scale-space feature representations by generalized integral images," in *Proceed. of IEEE Intern. Conf. Image Proc.*, San Antonio, TX, USA, September 2007.

- [133] L.-k. Liu, T. Nguyen, and S. H. Chan, "Do we really need Gaussian filters for feature point detection?" in *Proceed. of Europ. Sign. Proc. Conf.* Bucharest, Romania: EURASIP, August 2012.
- [134] B. Weiss, "Fast median and bilateral filtering," in *ACM SIGGRAPH 2006 Papers*, ser. SIGGRAPH '06, Boston, Massachusetts, 2006.
- [135] P. S. Heckbert, "Filtering by repeated integration," in *Proceed. of 13th Annual Conf. on Comp. Graphics and Interactive Techniques*, ser. SIGGRAPH '86. New York, NY, USA: ACM, 1986, pp. 315–321. [Online]. Available: <http://doi.acm.org/10.1145/15922.15921>
- [136] D. Marimon, "Fast non-uniform filtering with symmetric weighted integral images," in *Proceed. of IEEE Intern. Conf. Image Proc.*, Hong Kong, Hong Kong, September 2010.
- [137] A. Bhatia, W. E. Snyder, and G. Bilbro, "Stacked integral image," in *Proceed. of IEEE Intern. Conf. on Rob. and Autom.*, Anchorage, Alaska, USA, May 2010.
- [138] V. Fragoso, G. Srivastava, A. Nagar, Z. Li, K. Park, and M. Turk, "Cascade of box (CABOX) filters for optimal scale space approximation," in *Proceed. of IEEE Intern. Conf. on Comp. Vision and Pattern Rec. workshops*, Columbus, Ohio, USA, June 2014.
- [139] M. Hussein, F. Porikli, and L. Davis, "Kernel integral images: A framework for fast non-uniform filtering," in *Proceed. of IEEE Intern. Conf. on Comp. Vision and Pattern Rec.*, Anchorage, Alaska, USA, June 2008.
- [140] F. Moreno-Noguer, "Deformation and illumination invariant feature point descriptor," in *Proceed. of IEEE Intern. Conf. on Comp. Vision and Pattern Rec.*, Colorado Springs, USA, 2011.
- [141] H. Ling and D. W. Jacobs, "Deformation invariant image matching," in *Proceed. of IEEE Intern. Conf. on Comp. Vision*, Beijing, China, October 2005.
- [142] Y. Verdie, K. M. Yi, P. Fua, and V. Lepetit, "TILDE: A temporally invariant learned detector," in *Proceed. of IEEE Intern. Conf. on Comp. Vision and Pattern Rec.*, Boston, MA, USA, June 2015.
- [143] A. Rana, G. Valenzise, and F. Dufaux, "An evaluation of HDR image matching under extreme illumination changes," in *Proceed. of Intern. Conf. on Visual Comm. and Image Proc.*, Chengdu, China, November 2016.
- [144] B. Přebyl, A. Chalmers, and P. Zemčík, "Feature point detection under extreme lighting conditions," in *Proceed. of the 28th Spring Conf. on Computer Graphics*. ACM, 2013, pp. 143–150.
- [145] H. Mansour, S. Rane, P. T. Boufounos, and A. Vetro, "Video querying via compact descriptors of visually salient objects," in *Proceed. of IEEE Intern. Conf. Image Proc.*, Paris, France, 2014.
- [146] L. Baroffio, M. Cesana, A. Redondi, M. Tagliasacchi, and S. Tubaro, "Coding visual features extracted from video sequences," *IEEE Trans. Image Processing*, vol. 23, no. 5, pp. 2262–2276, 2014.
- [147] G. Antipov, S.-A. Berrani, N. Ruchaud, and J.-L. Dugelay, "Learned vs. hand-crafted features for pedestrian gender recognition," in *Proceed. of the 23rd Intern. Conf. on Multimedia*. Brisbane, Australia: ACM, October 2015.