UM DESCRITOR ROBUSTO E EFICIENTE DE PONTOS DE INTERESSE: DESENVOLVIMENTO E APLICAÇÕES

ERICKSON RANGEL DO NASCIMENTO

UM DESCRITOR ROBUSTO E EFICIENTE DE PONTOS DE INTERESSE: DESENVOLVIMENTO E APLICAÇÕES

Tese apresentada ao Programa de Pós-Graduação em Ciência da Computação do Instituto de Ciências Exatas da Universidade Federal de Minas Gerais como requisito parcial para a obtenção do grau de Doutor em Ciência da Computação.

Orientador: Mario Fernando Montenegro Campos

Belo Horizonte
Agosto de 2012

ERICKSON RANGEL DO NASCIMENTO

ON THE DEVELOPMENT OF A ROBUST, FAST AND LIGHTWEIGHT KEYPOINT DESCRIPTOR AND ITS APPLICATIONS

Thesis presented to the Graduate Program in Computer Science of the Federal University of Minas Gerais in partial fulfillment of the requirements for the degree of Doctor in Computer Science.

ADVISOR: MARIO FERNANDO MONTENEGRO CAMPOS

Belo Horizonte
August 2012

Nascimento, Erickson Rangel do
N244d Um descritor robusto e eficiente de pontos de interesse: desenvolvimento e aplicações / Erickson Rangel do Nascimento. — Belo Horizonte, 2012
xxvi, 86 f. : il. ; 29cm

Tese (doutorado) — Universidade Federal de Minas Gerais — Departamento de Ciência da Computação.
Orientador: Mario Fernando Montenegro Campos.

1. Computação - Teses. 2. Visão Computacional - Teses. I. Orientador. II. Título.

CDU 519.6*84(043)

UNIVERSIDADE FEDERAL DE MINAS GERAIS
INSTITUTO DE CIÊNCIAS EXATAS
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

FOLHA DE APROVAÇÃO

Um descritor robusto e eficiente de pontos de interesse: desenvolvimento e aplicações

ERICKSON RANGEL DO NASCIMENTO

Tese defendida e aprovada pela banca examinadora constituída pelos Senhores:

Prof. Mario Fernando Montenegro Campos - Orientador
Departamento de Ciência da Computação - UFMG

Prof. Flávio Luis Cardeal Pádua
Departamento de Computação - CEFET/MG

Renato Cardoso Mesquita
Prof. Renato Cardoso Mesquita
Departamento de Engenharia Elétrica - UFMG

Prof. Thomas Maurice Lewiner
Departamento de Matemática - PUC-RJ

William Robson Schwartz
Prof. William Robson Schwartz
Departamento de Ciência da Computação - UFMG

Belo Horizonte, 21 de agosto de 2012.

To my parents, Pekison and Maria Geralda, my sisters, Tamarakajalgina and Kellyuri, my beloved Marcela and my tutor and dear friend Cleber Resende (in memoriam), who believed in me and taught me to face the challenges with passion and humility.

Acknowledgments

A doctorate is a work of numerous individuals providing indispensable support, collaborating with their skills and goodwill. In fact, it is not a work of only one person. Thus, I would like to seize this chance to express my gratitude towards all that helped me in my research work.

I would like to express my very great appreciation to my advisor Prof. Mario F. M. Campos for his valuable and constructive suggestions during the development of this thesis. His great advice on how to do high quality research significantly contributed for the results of this work and I will always take with me.

I would also like thank the thesis committee members, professors Flavio L. C. Pádua, Renato C. Mesquita, Thomas M. Lewiner and William R. Schwartz for their suggestions to the improvement of this research work. Special thanks should be given to Prof. William for his assistance in the object recognition experiments.

Many thanks go to all VeRLab members for creating a pleasant working atmosphere. I would like to specially acknowledge the collaboration and prolific discussions with Gabriel L. Oliveira, Antônio W. Vieira, Vilar C. Neto and Armando A. Neto. My thanks are also extended to James Milligan and Emily LeBlanc for their very careful review of this text.

My grateful thanks to colleagues, professors and staff of the Computer Science Department of the UFMG, specially Renata, Sonia, Sheila, Linda and Tulia, who always were available to help me with the paperwork with a minimum of obstacles.

I wish to thank my parents, Pekison and Maria Geralda, and my sisters Tamarakajalgina and Kellyuri, for always believing in me. To my family, my deepest gratitude.

Finally, I want to thank my beloved Marcela, for her continuous support and affection wich help me to maintain my sanity. This work would not be possible without your support.

This research was supported by CNPq and CAPES.

"Olhe o mundo."
(Cleber Gonçalves Resende)

Resumo

Diferentes metodologias para reconhecimento de objetos, reconstrução e alinhamento tridimensional, possuem no cerne de seu desenvolvimento o problema de correspondência. Devido à ambiguidade em nosso mundo e à presença de ruídos nos processos de aquisições de dados, obter correspondências de qualidade é um dos maiores desafios em Robótica e Visão Computacional. Dessa maneira, a criação de descritores que identifiquem os elementos a serem correspondidos e que sejam capazes de gerar pares correspondentes corretamente é de grande importância.

Nesta tese, introduzimos três novos descritores que combinam de maneira eficiente aparência e informação geométrica de images RGB-D. Os descritores apresentados neste trabalho são largamente invariantes a rotação, mudanças de iluminação e escala. Além disso, para aplicações cujo principal requisito é o baixo consumo computacional em detrimento de alta precisão na correspondência, a invariância a rotação e escala podem ser facilmente desabilitadas sem grande perda na qualidade de discriminância dos descritores.

Os resultados dos experimentos realizados nesta tese demonstram que nossos descritores, quando comparados a três descritores padrões da literatura, SIFT, SURF (para images com texturas) e Spin-Images (para dados geométricos) e ao estado da arte CSHOT, foram mais robustos e precisos.

Foram também realizados experimentos com os descritores em duas aplicações distintas. Nós os utilizamos para a detecção e reconhecimento de objetos sob diferentes condições de iluminação para a construção de mapas com informações semânticas e para o registro de múltiplos mapas com profundidade e textura. Em ambas as aplicações, nossos descritores demonstraram-se mais adequados do que outras abordagens, tendo sido superiores em tempo de processamento, consumo de memória, taxa de reconhecimento e qualidade do registro.

Palavras-chave: Visão Computacional, Descritores, Pontos de Interesse, Imagens RGB-D.

Abstract

At the core of a myriad of tasks such as object recognition, tridimensional reconstruction and alignment resides the critical problem of correspondence. Due to the ambiguity in our world and the presence of noise in the data aquisition process, performing high quality correspondence is one of the most challenging tasks in robotics and computer vision. Hence, devising descriptors, which identify the entities to be matched and that are able to correctly and reliably establish pairs of corresponding points is of central importance.

In this thesis, we introduce three novel descriptors that efficiently combine appearance and geometrical shape information from RGB-D images, and are largely invariant to rotation, illumination changes and scale transformations. For applications that demand speed performance in lieu of a sophisticated and more precise matching process, scale and rotation invariance may be easily disabled. Results of several experiments described here demonstrate that as far as precision and robustness are concerned, our descriptors compare favorably to three standard descriptors in the literature, namely: SIFT, SURF (for textured images) and Spin-Images (for geometrical shape information). In addition, they outperfom the state-of-the-art CSHOT, which, as well as our descriptors, combines texture and geometry.

We use these new descriptors to detect and recognize objects under different illumination conditions to provide semantic information in a mapping task. Furthermore, we apply our descriptors for registering multiple indoor textured depth maps, and demonstrate that they are robust and provide reliable results even for sparsely textured and poorly illuminated scenes. In these two applications we compare the performance of our descriptors against the standard ones in the literature and the state-of-the-art. Experimental results show that our descriptors are superior to the others in processing time, memory consumption, recognition rate and alignment quality.

Keywords: Computer Vision, Descriptors, Keypoints, RGB-D Images.

List of Figures

List of Tables

Acronym List

EDVD Enhanced Descriptor for Visual and Depth Data

BASE Binary Appearance and Shape Elements

BRAND Binary Robust Appearance and Normal Descriptor

CSHOT Color-SHOT

SHOT Signature of Histograms of Orientations

VOSCH Voxelized Shape and Color Histograms

SLAM Simultaneous Localization And Mapping

SIFT Scale Invariant Feature Transform

SURF Speeded-Up Robust Features

BRIEF Binary Robust Independent Elementary Features

ROC Relative Operating Characteristic

ICP Iterative Closest Point

LIDAR Light Detection And Ranging

PCA Principal Component Analysis

LBP Local Binary Patterns

AUC Area Under Curve

BoF Bag of Features

PLS Partial Least Squares

SAC Sampled Consensus-Initial Alignment

FOV Field of View

IMU Inertial Measurement Unit

Contents

Chapter 1

Introduction

AT THE HEART OF NUMEROUS TASKS both in robotics and computer vision resides the crucial problem of correspondence. Methodologies for building accurate tridimensional models of scenes, Simultaneous Localization And Mapping (SLAM), tracking, and object recognition and detection algorithms are some examples of techniques in which the correspondence plays a central role in the pipeline process. Among these methodologies, we are particularly interested in 3D model building and object recognition. Methodologies for 3D model building usually have to handle alignment and registration issues finding a set of corresponding points in two different views. The learning algorithms used in object detection and recognition rely on selecting corresponding points to reduce data dimensionality, which makes data intensive model building a manageable task.

The correspondence problem consists of organizing pairs of entities, *e.g.* pixels in images X and Y, according to a similarity function. The aim is to find a relation f_c which determines for every element $x \in X$ a correspondent element $y \in Y$. Formally, $\forall x \in X$, $\sum g(y, f_c(x)) = 0$, where g is a similarity function and y is the corresponding element of x.

Due to ambiguity in our world and the presence of noise in the data aquisition process, finding out f_c is one of the most challenging tasks in robotics and computer vision. For example, in the correspondence of pixels in two images, the same world point may be different under distinct imaging conditions and different points may present identical appearance when observed from different viewpoints. Futhermore, image noise may severely interfere with the correspondence process.

The correspondence task, or matching process, can be broken down into three main procedures (Figure 1.1):

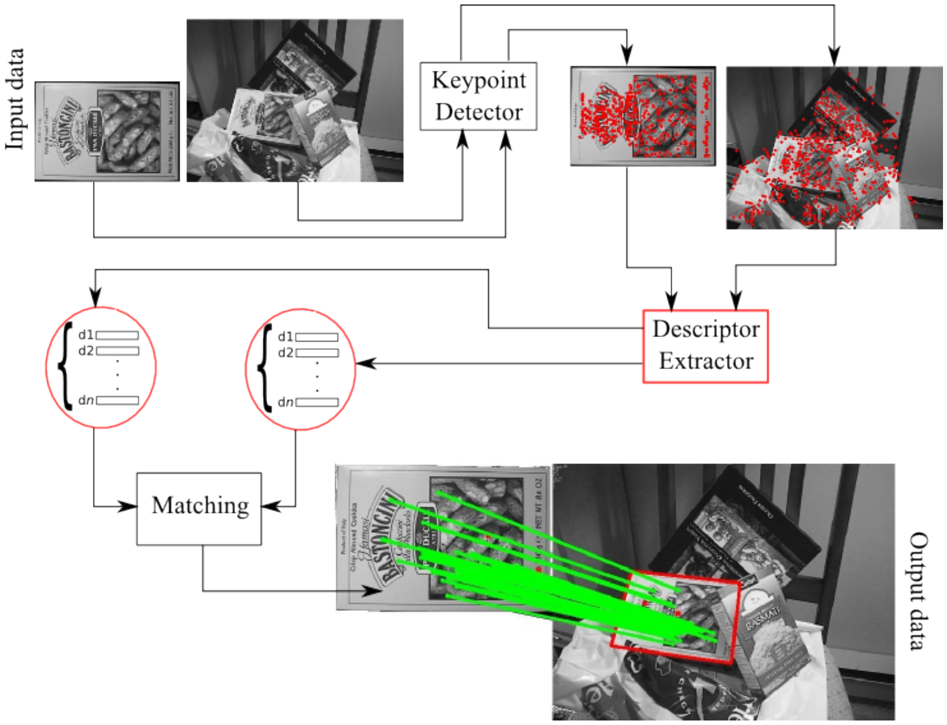


Figure 1.1. Matching feature descriptors. For each detected keypoint, a signature is computed called a descriptor. Theses descriptors are matched with another set of keypoint descriptors in a different image.

- Detect and select a set of interest points, which we will call *keypoints*: Here we use a keypoint detector. Keypoint detectors look for points in images with properties such as repeatibility, which may create less ambiguity and are in discriminative regions. There is a vast amount of literature about keypoint detectors [Harris and Stephens, 1988; Lowe., 2004; Bay et al., 2008; Rosten et al., 2010; Agrawal et al., 2008] and the development of algorithms will not be the focus of this work;

- Compute a signature, commonly called *descriptor*, for each keypoint: This step computes an identification for the keypoints detected in the previous step. Such identification is generated based on a *local* analysis of the region around the keypoint and is represented by n-dimensional vector. This descriptor is then used to compute the similarity distance between the keypoints;

- Find the nearest neighbor in descriptor space: This step is accomplished by comparing the descriptors using some similarity distance, *e.g.* Euclidian distance, Manhattan distance or Hamming distance.

It is clear that even a perfect similarity distance combinated with the best keypoint detector will not compensate for a descriptor with poor discriminative characteristics. Hence, devising descriptors that are able to correctly and reliably establish pairs of corresponding points is of central importance.

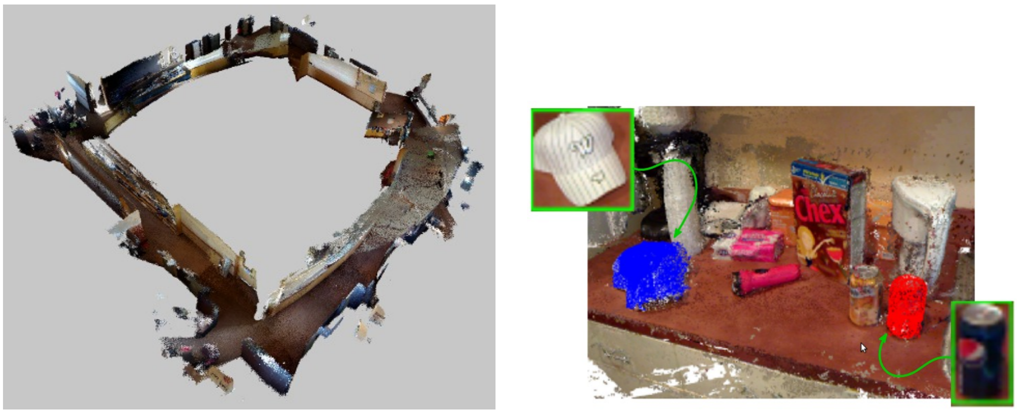
In this thesis, we focus on the creation and analysis of feature descriptors for keypoints in color images and range images. The proposed methodology is based on the concern of how to reach the best possible descriptor. Thus, the main problem of this work can be defined by the question:

Problem 1.1 (Thesis Problem). *How to design and build a robust descriptor for keypoints in range and visual data?*

A robust feature descriptor or any approximation of such descriptor shares a common set of properties. These properties yield the requirements for an ideal descriptor which must compute strong discriminative signatures, providing unique identification for keypoints independent of the viewpoint and illumination conditions. In this work we elected the set Π of eight properties:

- π_0: Robustness to noise;
- π_1: Scale invariance;
- π_2: Rotation invariance;
- π_3: Illumination invariance;
- π_4: Robustness to textureless scenes;
- π_5: Low processing time to compute;
- π_6: Low processing time to compare;
- π_7: Low memory consumption;
- π_8: Keypoint detection independence.

The properties in set Π have been used as a design guide and in the evaluation and development process of descriptors during this work. These properties have a strong relation with each step depicted in Figure 1.1. In particular, the π_8 property is linked to two important steps in correspondence problem: keypoint detection and keypoint description. Although this thesis addresses the keypoint description step, it is crucial for descriptor algorithms to have high independence of keypoint detection, which can avoid adding noise from keypoint detection to description and



(a) Result from [Henry et al., 2010] (b) Result from [Lai et al., 2011a]

Figure 1.2. Examples of works using geometrical and intensity information for (a) Indoor Environment Reconstruction and (b) Object Detection and Recognition.

allows to use any of detector algorithms proposed every year. The reasons we chose the others properties will be detailed in the next section.

1.1 Motivation

The matching of descriptors is at the core of a myriad of applications in computer vision and robotics. Three-dimensional alignment (Figure 1.2 (a)), SLAM, tracking, detection and recognition of objects (Figure 1.2 (b)) and structure from motion are some of the applications that rely on feature point matching methods. Hence, it is of great importance to the success of these systems to develop robust, invariant and discriminative descriptors, since rotation, illumination and the viewpoint are not fixed. Moreover, the data of these systems is noisy.

Additionally, the requirements from the online application for limited hardware, as mobile phones and embedded systems, are not reached at an acceptable level by the state-of-the-art descriptors. This leads to the demand for descriptors that are robust, fast to compute and match, and memory efficient.

Although available 3-D sensing techniques have been available, such as techniques based on Light Detection And Ranging (LIDAR), time-of-flight (Canesta), and projected texture stereo (PR2 robot), they are still very expensive and demand a substantial engineering effort. With the recent introduction of fast and inexpensive RGB-D sensors (where *RGB* implies trichromatic intensity information and *D* stands for depth) the integration of synchronized intensity (color) and depth has become easier to obtain (Figure 1.3).

(a) (b) (c) (d)

Figure 1.3. Examples of recently released domestic tridimensional sensors: (a) Microsoft Kinect; (b) ASUS WAVI Xtion; (c) Minoru Webcam 3D; (d) 3D LG Optimus cell phone.

RGB-D systems output color images and the corresponding pixel depth information enabling the acquisition of both depth and visual cues in real-time. These systems have opened the way to obtain 3D information with unprecedented tradeoff of richness and cost. One such system is the Kinect [Microsoft, 2011], a low cost commercially available system that produces RGB-D data in real-time for gaming applications.

Robust, fast and low memory consumption descriptors that efficiently use the available information, like color and depth, will play a central role in the search of the optimal descriptor.

1.2 Thesis Goals and Contributions

In this thesis, we present three novel descriptors, which efficiently combine intensity and shape information to substantially improve discriminative power enabling enhanced and faster matching. We aim to advance in the task of building robust and efficient descriptors suitable for online applications. Experimental results presented later show that our approach is both robust and computationally efficient.

Moreover, we tested the descriptor capabilities in indoor environment alignment and object recognition and obtained better results when comparing with three other descriptors well known in literature.

The main contributions of this thesis can be summarized by the three developed RGB-D descriptors:

1. The Enhanced Descriptor for Visual and Depth Data (EDVD), which efficiently combines visual and shape information to substantially improve discriminative power, enabling high matching performance. Unlike most current methodologies, our approach includes in its design scale and rotation transforms in both image and geometrical domains;

2. Binary Appearance and Shape Elements (BASE) that, like EDVD, efficiently fuses visual and shape information to improve the discriminative power, but provides faster matching and lower memory consumption;

3. A fast, lightweight and robust feature point descriptor, called Binary Robust Appearance and Normal Descriptor (BRAND), which presents all the invariant properties of EDVD and is as fast as BASE with the same memory consumption.

In summary, our main contribution is to exploit the techniques to build a robust, fast and low memory consumption descriptor suitable for online 3D mapping and object recognition applications, which is able to work in modest hardware configurations with limited memory and processor use.

Portions of this work have been published in the following international peer reviewed journal, conference proceedings and workshop:

1. Nascimento, E. R.; Oliveira, G. L.; Vieira, A. W.; Campos, M. F. M.. *On The Development of a Robust, Fast and Lightweight Keypoint Descriptor*. Neurocomputing;

2. Nascimento, E. R; Schwartz W. R; Campos, M. F. M.. *EDVD - Enhanced Descriptor for Visual and Depth Data*. IAPR International Conference on Pattern Recognition (ICPR), 2012, Tsukuba - Japan;

3. Nascimento, E. R.; Oliveira, G. L.; Campos, M. F. M.; Vieira, A. W. and Schwartz, W. R. *BRAND: A Robust Appearance and Depth Descriptor for RGB-D Images*, in IEEE Intl. Proc. on Intelligent Robots and Systems (IROS), 2012, Vilamoura - Algarve - Portugal;

4. Nascimento, E. R.; Schwartz, W. R.; Oliveira, G. L.; Vieira, A. W.; Campos, M. F. M.; Mesquita, D. B.. *Appearance and Geometry Fusion for Enhanced Dense 3D Alignment*, in XXV Conference on Graphics, Patterns and Images (SIBGRAPI), 2012, Ouro Preto - Minas Gerais - Brazil;

5. Nascimento, E. R.; Oliveira, G. L.; Vieira, A. W.; Campos, M. F. M.. *Improving Object Detection and Recognition for Semantic Mapping with an Extended Intensity and Shape based Descriptor.* In: IROS 2011 workshop - Active Semantic Perception and Object Search in the Real World (ASP-AVS-11), 2011, San Francisco - California - USA.

1.3 Organization of the Thesis

This thesis is organized as follows. The next chapter is intended to provide the reader with an overview of the main and more recent techniques in the creation of descriptors; it reviews methods to build image and geometrical descriptors where the most relevant algorithms are discussed in detail. In Chapter 3, we describe the methodology to build approximations of a robust descritor according to the properties in Set Π and present an analysis of its capabilities in comparison to other descriptors in Chapter 4. Following in Chapter 5, we present the use of our descriptor in indoor environment reconstruction and semantic mapping applications. Chapter 6 concludes with a discussion about the limitations and contributions of this work, and highlights future research directions.

Chapter 2

Related Work

In spite of all the adversities in the correspondence of pixels in images, like noise and ambiguity, much progress towards estimating an aproximated correspondence relation f_c (defined in Chapter 1) has been made in the last decade for color images and range images.

Methodologies for keypoint detectors, descriptor creation and matching have been proposed in the last decade with great success. In this chapter we review the related literature on the creation of descriptors. Due to the enormous quantity of work that has been published on the subject, we will focus on those with a direct relation with ours.

2.1 Keypoint Detection

As seen in Figure 1.1, the first step in the matching process is keypoint detection. Thus, we present in this section a discussion of the main concepts present in detector algorithms.

The main goal of detectors is to assign a saliency score to each pixel of an image. This score is used to select a small subset of pixels that present as properties [Tuytelaars and Mikolajczyk, 2008]:

- Repeatability: The selected pixels should be stable under several image pertubations;
- Distinctiveness: The neighborhood around each keypoint should have intesity pattern with strong variantions;
- Locality: The features should be a function of local information;

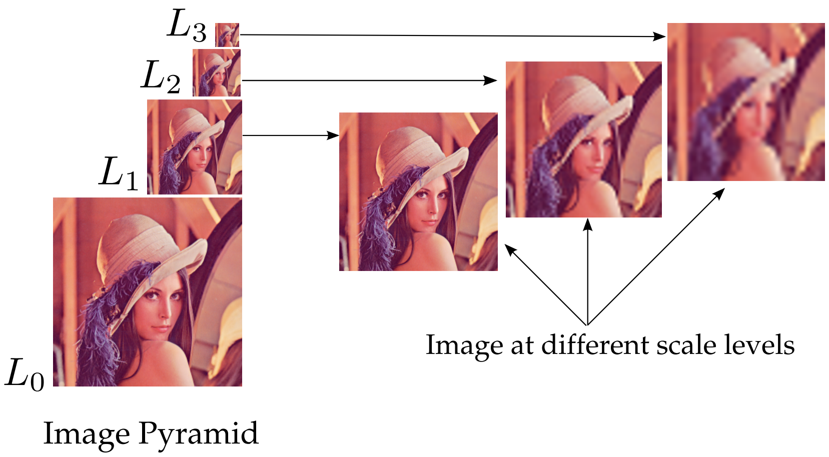


Figure 2.1. The original image L_0 is repeatedly subsampled and smoothed generating a sequence of reduced resolution images L_1, L_2 and L_3 in different scale levels.

- Accurately localizable: The localization process should be less error-prone with respect to scale and shape;
- Efficient: Low processing time.

Corner detection was used in earlier techniques to detect keypoints [Zhang et al., 1995; Schmid and Mohr, 1997], however, corner detection approaches have a limited performance since they generally examine the image at only a single scale.

The more recent detector algorithms are designed to detect the same keypoints in different scalings of an image. Using the scale-space representation [Lindeberg, 1994], these algorithms are able to provide scale invariance in image features. Futhermore, the scale-space representation is useful in reducing the noise.

The scale-space representation L for an image I is defined as

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y), \tag{2.1}$$

where $*$ is the 2D convolution operator and

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{\frac{x^2+y^2}{2\sigma^2}} \tag{2.2}$$

is a Gaussian kernel.

Generally, the detector methodologies implement the scale-space as an image pyramid. In each level of the pyramid a smoothed and sub-sampled representation of the image is stored (see Figure 2.1). Thus, keypoint detection is performed comparing the maxima and minima of a pixel response in the scale-space fuction in the pyramid. In addition to detection, this process determines the scale of each keypoint found.

In order to provide invariance to rotation transformations, detector algorithms estimate the characteristic direction of the region around each keypoint. This direction is called *Canonical Orientation*, and it is defined by the pattern of gradients in the keypoint's neighborhood.

2.2 Descriptor Extraction

The approaches for assembling descriptors for keypoints can be categorized based on the type of data acquired from the scene. That is, data may be composed of textured images or of depth images.

In the last years, textured images have been the main choice. They provide a rich source of information which naturally ushered the use of texture based descriptors in several methods despite their inherent complexity. Computer Vision literature presents numerous works on using different cues for correspondence based on texture [Lowe., 2004; Bay et al., 2008; Calonder et al., 2010; Leutenegger et al., 2011; Rublee et al., 2011]. Virtually all of these techniques are based on the analysis of the distribution of local gradients.

Scale Invariant Feature Transform (SIFT) and Speeded-Up Robust Features (SURF) are the most popular image descriptors algorithms. Thanks to their descriminative power and speed, they became standard for several tasks such as keypoint correspondence and object recognition. For these reasons we chose both as competitors with the methods proposed in this thesis and we detail them in the next sections. We also present Binary Robust Independent Elementary Features (BRIEF) descriptor since our methodology was partially inspired on it.

2.2.1 SIFT Descriptor

Lowe, in his landmark paper [Lowe., 2004], presents the keypoint descriptor called SIFT. Although Lowe proposed SIFT to be used in object recognition applications, due to the high discriminative power and stability, his algorithm became the most used keypoint descriptor in a myriad of other tasks.

The whole creation process of SIFT is illustrated in Figure 2.2. The first step is to compute, for each pixel around the keypoint location, the gradient magnitude

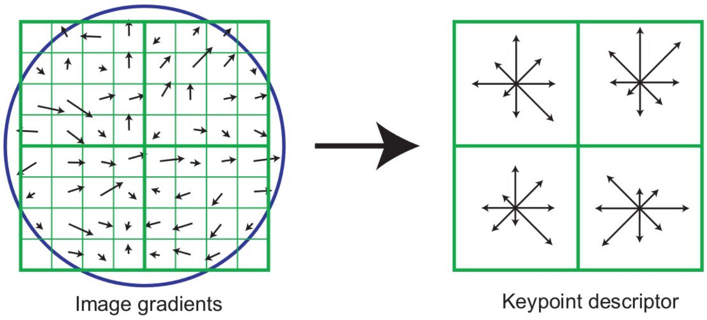


Figure 2.2. Computing a SIFT descriptor. First, the magnitudes and orientations of local gradients are computed around the keypoint. The magnitudes are weighted by a Gaussian window (blue circle) and accumulated in 4 orientation histograms. Each histogram has 8 bins of orientation corresponding to a subregion. Then, the bins for all histograms are concatenated to form the descriptor. Unlike the example in this image, the standard implementation of SIFT uses a 16×16 sample on the left and 4×4 histograms on the right. Illustration taken from [Lowe., 2004].

and orientation. The magnitude $m(x, y)$ of the pixel (x, y) is given by:

$$m(x,y) = \sqrt{[I(x+1,y) - I(x-1,y)]^2 + [I(x,y+1) - I(x,y-1)]^2}, \tag{2.3}$$

and its orientation $\theta(x, y)$ is estimated as:

$$\theta(x,y) = \arctan\left[\frac{I(x,y+1) - I(x,y-1)}{I(x+1,y) - I(x-1,y)}\right], \tag{2.4}$$

where $I(x, y)$ is the intensity of pixel (x, y) of the smoothed and subsampled image in the level of the scale-space pyramid where the keypoint was detected.

A region of 16×16 pixels, centred in the keypoint localization, is subdivided in 4×4 subregions. These 16 subregions are rotated relative to the canonical orientation computed for the keypoint. For each subregion, a histogram with 8 orientation bins is computed. The magnitute values for all gradients inside of the region are weighed by a Gaussian window and accumulated into the orientation histograms.

The 8 bins of all 16 histograms are concatenated forming the 128-vector, which after normalization, represents the SIFT descriptor. The whole procedure makes the descriptor scale and rotation invariant thanks to scale-space and the canonical orientation, and due to normalization the descriptors are partially robust to illumination changes.

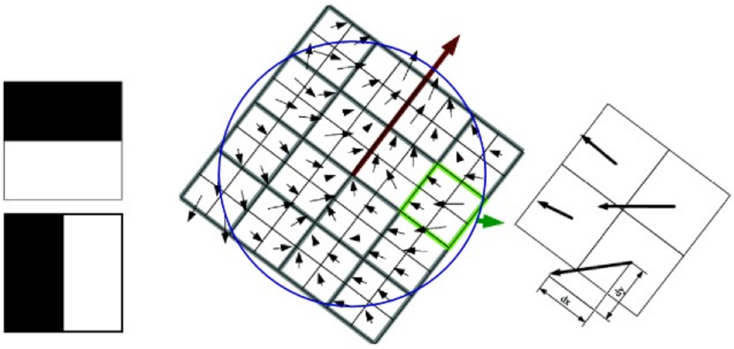


Figure 2.3. Computing a SURF descriptor. Two Haar wavelet filters (left) are used to estimate the local gradients inside of an oriented quadratic grid centred in the keypoint position (middle). The wavelet responses are weighted with a Gaussian (blue circle) and, for each 2×2 sub-region (right), is computed the sums of dx, $|dx|$, dy and $|dy|$ relatively to the canonical orientation (red arrow). The final descriptor is composed of 4×4 vectors $\mathbf{v} = (\sum dx, \sum dy, \sum |dx|, \sum |dy|)$ concatenated. Illustration taken from [Bay et al., 2008].

2.2.2 SURF Descriptor

Although SIFT brings forth discriminative descriptors, it has a high processing cost. To overcome this issue, Bay et al. [2008] propose the faster algorithm SURF. This algorithm can be seen as an approximation of SIFT and shares the idea of using histograms based on local gradients. Despite the approximations in descriptor creation, there is no significant loss in robustness or rotation and scale invariance.

Like SIFT, the creation process of SURF descriptor consists of centering a square region in the keypoint location. The keypoint scale in scale-space is used to determine the size of this region and the canonical orientation of its direction. However, differently from SIFT which uses pixel differences to compute the gradients, SURF creates the local gradient distribution using Haar wavelet responses in horizontal and in vertical directions (the Haar wavelet filters used are shown in Figure 2.3).

SURF computes, in both x and y directions, four sums: i) sum of gradients $\sum dx$; ii) sum of gradients $\sum dy$; iii) sum of absolute gradients $\sum |dx|$ and iv) sum of absolute gradients $\sum |dy|$ (see 2.3). These sums are computed for each one of the 4×4 sub-regions and concatenated forming 16 vectors $\mathbf{v} = (\sum dx, \sum dy, \sum |dx|, \sum |dy|)$. The final SURF descriptor is produced by concatenating all the 16 vectors, creating a 64-dimensional vector.

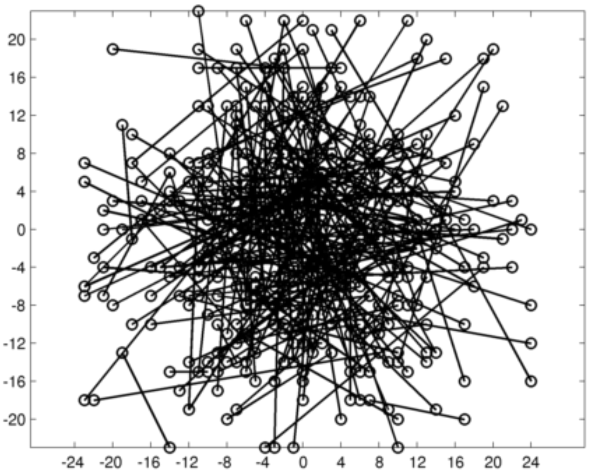


Figure 2.4. Patch **P** with 48×48 pixels indicating 256 sampled pairs of pixel locations used to construct the binary feature.

2.2.3 BRIEF Descriptor

Despite the high discriminative power of SIFT and SURF, they suffer with slow match and high processing time and memory consumption (vectors with 128 and 64 floats respectively). Hence, these algorithms are not feasible in applications where it is necessary to store millions of descriptors or have real-time constraints.

Dimensionality reduction methodologies, such as Principal Component Analysis (PCA) [Ke and Sukthankar, 2004], Linear Discriminant Embedding (LDE) [Hua et al., 2007], algorithms based on L1-norm-based [Pang and Yuan, 2010] and [Pang et al., 2010], and quantization techniques that convert floating-point coordinates into integers coded on fewer bits are used by some of the approaches to solve the descriptor dimensionality problem. However, those techniques involve further post-processing, usually with high computation charge, of a long descriptor which is already costly to compute. Furthermore, PCA and LDE methodologies may lead to overfitting and reduce the performance.

More recently, several compact descriptors, such as Calonder et al. [2010], Leutenegger et al. [2011], Rublee et al. [2011], Ambai and Yoshida [2011], Kembhavi et al. [2011] and Choi et al. [2012] have been proposed employing ideas similar to those used by Local Binary Patterns (LBP) [Ojala et al., 1996]. Those descriptors are computed using simple intensity difference tests, which have small memory consumption and modest processing time in the creation and matching processes.

The use of binary strings as descriptors has been used with promising results and one successful example of this methodology is the BRIEF descriptor [Calonder et al., 2010]. As this work is inspired by BRIEF we will detail its methodology.

Assembling binary descriptors In order to generate a string of bits, BRIEF's approach consists of computing individual bits by comparing the intensities of pairs of points in a neighborhood around each detected keypoint. Similar to SIFT and SURF descriptors, the BRIEF methodology estimates for every keypoint a gradient field.

The pairs are selected by a patch $\mathbf{P}$ of size $S \times S$, centred at a keypoint position. This patch is smoothed to reduce sensitivity, increase stability and repeatibility. Calonder et al. [2010] tested five configurations to build a spatial arrangement of the patch and the best results were reached by using an isotropic Gaussian distribution $(\mathbf{X}, \mathbf{Y})$ $i.i.d.$ $\mathcal{N}(0, \frac{S^2}{25})$. Figure 2.4 illustrates this arrangement. Each pair of pixels are indicated with line segments. This pairs distribution is created with random function, however, is fixed for all keypoints and it is centred at keypoint's location.

For all positions in a set of $(\mathbf{x}, \mathbf{y})$-locations, defined by the distribution, the following function is evaluated:

$$f(\mathbf{P}, \mathbf{x}, \mathbf{y}) = \begin{cases} 1 & \text{if } p(\mathbf{x}) < p(\mathbf{y}) \\ 0 & \text{otherwise,} \end{cases} \tag{2.5}$$

where $p(\mathbf{x})$ is the pixel intensity at position $\mathbf{x} = (u, v)^T$ in the patch $\mathbf{P}$.

The final descriptor is encoded as a binary string computed by:

$$b(\mathbf{P}) = \sum_{i=1}^{256} 2^{i-1} f(\mathbf{P}, \mathbf{x}_i, \mathbf{y}_i). \tag{2.6}$$

One of the main disadvantages of BRIEF is the lack of invariance to scaling and rotation transform, differently from SIFT and SURF, BRIEF algorithm does not compute the canonical orientation. Nevertheless, according to the authors, BRIEF has shown be invariant to rotation of small degrees. Also, there are cases that the image orientation can be estimated using other sensors, such as a mobile phone equipped with Inertial Measurement Unit (IMU) and a robot that knows its attitude [Calonder et al., 2010].

2.3 Geometrical Descriptor Extraction

In nearly all approaches mentioned in previous section, feature descriptors are estimated from images alone, and they seldomly use other information such as geometry. As a consequence, common issues concerning real scenes such as variation in illumination and textureless objects may dramatically decrease the performance of techniques that are based only on texture.

With the growing availability of inexpensive, real time depth sensors, depth images are becoming increasingly popular and many new geometrical-based descriptors are proposed each year. The methodologies presented by Rusu et al. [2008a], Rusu et al. [2008b], Rusu et al. [2008c], Rusu et al. [2008d], Rusu et al. [2009], Steder et al. [2010], Tombari et al. [2010] and Steder et al. [2011] are some of the most recent approaches.

As in the case of textured images, region matching on geometrical data is most advantageous. However, due to the geometrical nature of the data, effective descriptors tend to present higher complexity, and large ambiguous regions may become a hinderance to the correspondence process.

Geometrical descriptors take advantage of the matrix like structures of depth images which have low discriminative power and are less useful. Nevertheless, information of such descriptors is most relevant for textureless scene regions where texture based descriptors are doomed to fail.

In order to define robust descriptors for depth data, large amounts of data are necessary to encompass sufficient information and to avoid ambiguities. Spin-Image [Johnson and Hebert, 1999] is the most popular and used algorithm of such descriptors.

2.3.1 Spin-Image

Johnson proposed in his paper [Johnson and Hebert, 1999] to represent a surface with a set of images enconding global properties. He created a view-independent descriptor, where an object-oriented coordinate system is fixed on the surface and does not change when viewpoint changes.

The object-oriented coordinate system is defined by a three-dimensional point $\mathbf{p}$ and its normal $\mathbf{n}$. It is usually trivial to estimate the direction of the normal for each point on the surface as the vector perpendicular to the surface in that point.

The origin of object-oriented coordinate systems is defined by a keypoint location $\mathbf{p}$. A tangent plane $\mathcal{P}$ to the point $\mathbf{p}$ is oriented perpendicularly to the surface normal $\mathbf{n}$. Using the point $\mathbf{p}$ and its normal $\mathbf{n}$, the algorithm defines the line $\mathcal{L}$, which together with the plane $\mathcal{P}$ determine a cylindrical coordinate system $\mathcal{O}$ without the polar angle coordinate, since it is not possible to determine this coordinate using just the point and its normal surface. Thus, in this cylindrical coordinate system, a point $\mathbf{x}$ is represented using α and β coordinates, where α is the (non-negative) perpendicular distance to line $\mathcal{L}$ and β is the signed perpendicular distance to plane $\mathcal{P}$. Figure 2.5 depicts the creation of this cylindrical coordinate system.

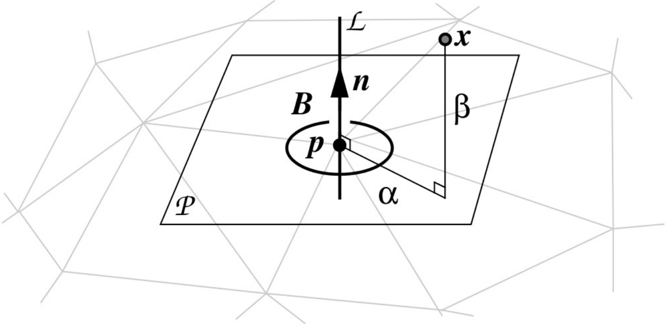


Figure 2.5. Creation of a cylindrical coordinate system based on point $\mathbf{p}$ and the normal surface $\mathbf{n}$ on this point. A point $\mathbf{x}$ can be represented in this cylindrical coordinate system using coordinates α and β. The plane $\mathcal{P}$ defines the object-oriented coordinate system. Illustration taken from [Johnson and Hebert, 1999].

After creation of the coordinate system, the next step is to generate a 2D image called a spin map. First, all point $\mathbf{x} \in \mathbb{R}^3$ in the point cloud are projected onto the cylindrical coordinate system $\mathcal{O}$, using the projection function $S_O : \mathbb{R}^3 \to \mathbb{R}^2$

$$S_O(\mathbf{x}) \to (\alpha, \beta) = \left(\sqrt{||\mathbf{x} - \mathbf{p}||^2 - \langle \mathbf{n}, (\mathbf{x} - \mathbf{p})\rangle^2}, \langle \mathbf{n}, \mathbf{x} - \mathbf{p}\rangle\right), \qquad (2.7)$$

where $\langle . \rangle$ is the dot product.

Then, the spin map is assembled by an accumulating schema. Using a 2D histogram with discrete bins to α and β values, the methodology updates these bins according to the projected points in the cylindrical coordinate system. This accumulaton process is shown in Figure 2.6.

By using a local coordinate system attached to the object surface and oriented along the keypoint normal to build the spin maps, the Spin-Image algorithm provides robustness and invariance to rotation transformation. Since the signature is independent of the viewpoint. It is well suited for depth maps and meshes in general.

Even though the geometrical descriptors, such as Spin-Image are accurate, they present high computational cost and memory consumption, and constructing a single descriptor for general raw point clouds or range images involves complex geometric operations.

2.4 Fusing Image and Geometrical Information

The combination of visual (from texture images), and geometrical shape (from depth information) cues has been adopted by some recent works as an alternative to im-

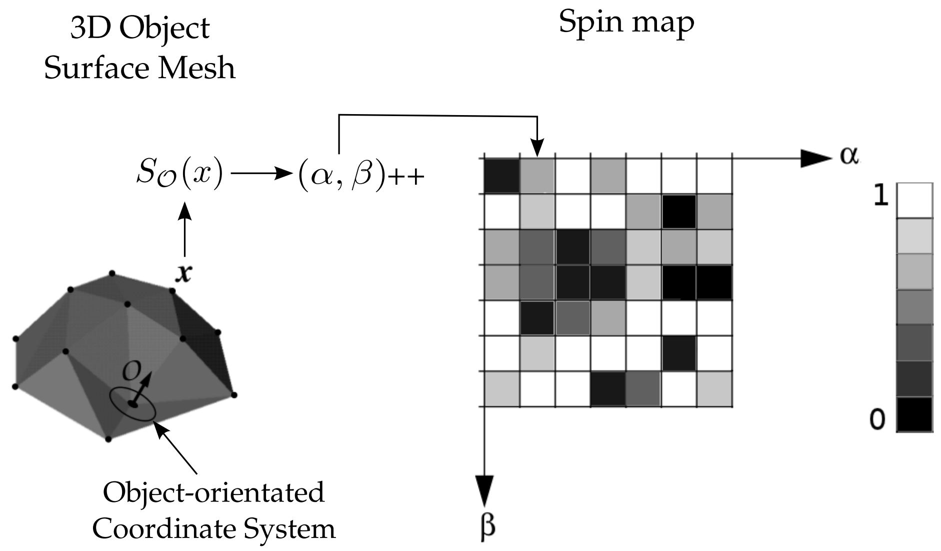


Figure 2.6. Spin-Image creation using 2D histogram. After the projection of a point x in the coordinate system $\mathcal{O}$, the resulting 2D point is accumulated into a discrete bin. Illustration based on [Johnson and Hebert, 1999].

prove object detection and recognition rate. The fusion of appearance and geometry information, which has shown a very promising approach for object recognition, is still in its opening movements. For example, Lai et al. [2011a,b] and Henry et al. [2010] combine both sources of information, but they applied well-known descriptors for each type of data, such as SIFT for texture and Spin-Image for shape and then concatenate both to form a new signature. However, as far as efficacy is concerned, Lai et al. [2011b] have shown that the combination of intensity and depth information outperforms approaches using either intensity or depth alone.

Most likely, the main reason many descriptors have not used shape information can be partially explained by the fact that, until recently, object geometry could not be easily or quickly obtained so as to be combined with image feature data.

In the last few years, the combination of multiple cues is becoming a popular approach for the design of descriptors. Zaharescu et al. [2009] proposed the MeshHOG descriptor using texture information of 3D models as scalar functions defined over 2D manifolds. Tombari et al. [2011] presented the Color-SHOT (CSHOT) descriptor based on an extension of their shape only descriptor Signature of Histograms of Orientations (SHOT) [Tombari et al., 2010] to incorporate texture. The authors compared CSHOT against MeshHOG and reported that CSHOT approach outperforms MeshHOG in processing time and accuracy. In the case of global descriptor, Kanezaki et al. [2011] presented the Voxelized Shape and Color Histograms (VOSCH) descriptor, which by combining depth and texture, was able to increase

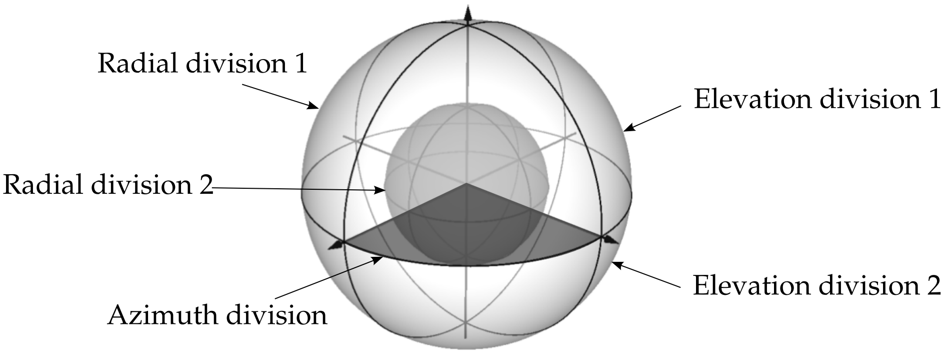


Figure 2.7. Isotropic Spherical Grid used by CSHOT descriptor. The space is partioned in 32 sectors: 4 azimuth divisions (the standard implementation uses 8 divisions), 2 elevation divisions and 2 radial divisions. Illustration based on [Tombari et al., 2011].

the recognition rate in cluttered scenes with obstruction. Since CSHOT is the state-of-the-art of shape and texture descriptor, we choose it as the main competitor of our methodology and we detail its construction process in next section.

2.4.1 CSHOT Descriptor

CSHOT signatures are composed of two concatenated histograms, one contains the geometric features and other with the texture information enconded. An isotropic spherical grid is overlaid onto each keypoint location. This grid has 32 sectors that divides the space in 8 azimuth divisons, 2 elevation divisions and 2 radial divisons (Figure 2.7 illustrates this spherical grid with 4 azimuth divisons).

For each sector, two local histograms are computed, one based on geometrical features and one with texture information. In the former, the algorithm accumulates into histograms bins according to the geometric metric

$$f(K, P) = \langle N_K, N_P \rangle, \tag{2.8}$$

where K is the keypoint, P represents a generic vertex belonging to the spherical support around K, N_K and N_P are the normals of keypoint and generic vertex, respectively and $\langle . \rangle$ is the dot product.

The accumulation in the texture histograms is performed using color triplets in CIELab space and the metric is based on the L_1 norm, given by

$$l(R_K, R_P) = \sum_{i=1}^{3} \|R_K(i) - R_P(i)\|, \tag{2.9}$$

where, R_K and R_P are the CIELab representation of the RGB triplet of the keypoint K and a generic vertex P in spherical support.

The standard implementation of CSHOT uses 11 bins for geometrical histograms and 31 bins for texture histograms. Since 32 histograms are computed for each cue, the resulting signature is a 1344-length vector.

In this work we take a similar approach to the problem. Our technique builds a descriptor which simultaneously takes into account both sources of information to create a unique representation of a region simultaneously considering texture and shape.

Our method aims at fusing visual (texture) and shape (geometric) information to enrich the discriminative power of our matching process to registration. On one hand, image texture information can usually provide better perception of object features, on the other hand depth information produced by 3D sensors is less sensitive to lighting conditions. Our descriptor brings forth the advantages of both texture and depth information. Moreover, it uses less memory space, since it was designed as a bit string, and less processing and matching time due to the low cost computations needed.

2.5 Descriptors Rating based on the Π Set

Table 2.1 summarizes the assigned properties of the descriptors described in this section. We give an individual rating with respect to the eight properties of Π set. Recall the properties described in Chapter 1, the Π set is composed of eight properties:

1. Robustness to noise;
2. Scale invariance;
3. Rotation invariance;
4. Illumination invariance;
5. Robustness to textureless scenes;
6. Low processing time to compute;
7. Low processing time to compare;
8. Low memory consumption;

	SURF	SIFT	Spin-Image	CSHOT	BRIEF
Robustness to noise	●	●	–	●	○
Scale invariance	●	●	–	●	–
Rotation invariance	●	●	●	●	–
Illumination invariance	○	○	●	●	–
Robustness to textureless scenes	–	–	●	●	–
Low time to compute	●	–	–	–	●
Low time to compare	○	○	–	–	●
Low memory consumption	–	–	–	–	●
Detection independence	○	○	–	–	○

Table 2.1. Descriptors rating based on the properties of an robust descriptor.

9. Keypoint detection independence.

We rate each descriptor of this chapter using the following criteria:

– : Descriptor does not have implemented any algorithm to cover the property;

○ : The property is implemented using an approximation;

● : Descriptor has a robust implementation of the property.

Table 2.1 shows a clear trade-off between computational efficiency and discrimination power. This trade-off is highlighted mainly in the comparison of the fast descriptor BRIEF and the robust descriptor SIFT. The independence of keypoint detection rate is base on the results shown in Chapter 4.

Chapter 3

A Computational Approach to Creation of Keypoint Descriptors

As observed in the preceding chapter, on the one hand, the use of texture information results in highly discrimative descriptors, on the other hand, binary strings and depth information from range images can reduce the cost of the descriptor creation and matching steps and provide descriptors robust to lack of texture and illumination changes.

Unlike tradional descriptors, such as SIFT, SURF and Spin-Image, that use only texture or geometry information, this chapter presents three novel descriptors to encode visual and shape information. However, differently from the work of Tombari et al. [Tombari et al., 2011], our algorithms provide robust, fast and lightweight signatures for keypoints. The methodology used by our descriptors uses the best information of both worlds in an efficient and low cost way.

All algorithms presented in this thesis receive as inputs a data pair (I, D), which denotes the output of a RGB-D sensor and a list $\mathcal{K}$ of detected keypoints. For each pixel x, $I(\mathbf{x})$ provides the intensity and $D(\mathbf{x})$ the depth information. Furthermore, we estimated a normal surface for all x as a map N, where $N(\mathbf{x})$ is efficiently estimated by Principal Component Analysis (PCA) over the surface defined by the depth map.

3.1 General Methodology

In this section we detail the methodology used to design three new descriptors. The stages of this methodology are illustrated in Figure 3.1.

Our methodology is composed of three main steps. In the first step, we com-

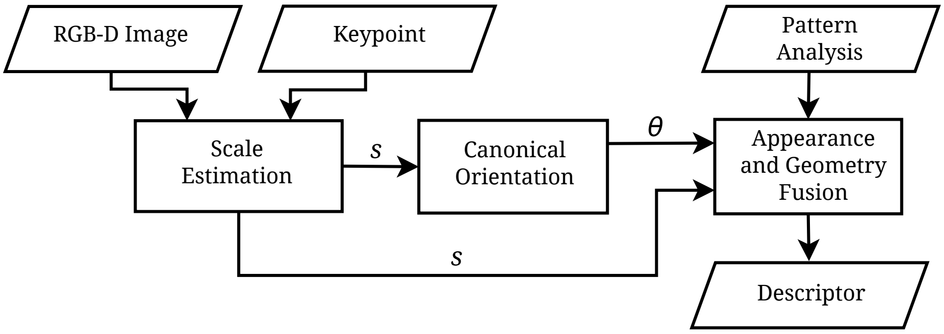


Figure 3.1. Methodology diagram. After computing the scale factor s using depth information from an RGB-D image, our methodology extracts a patch of the image in the RGB domain to estimate the canonical orientation θ of the keypoint. Finally, appearance and geometric information are fused together based on the features selected with a pattern analysis.

pute the scale factor using the depth information from RGB-D image. The scale factor is used in the next step (canonical orientation estimation) and in feature analysis in the keypoint's vicinity. In the canonical orientation estimating step, a patch in the RGB domain is extracted and used to estimate the characteristic angular direction of the keypoint's neighborhood. At last, we combine both appearance and geometric information to create keypoint descriptors that are robust, fast and lightweight.

3.1.1 Scale Assignment

Due to the lack of depth information in the images, approaches such as Lowe. [2004], Bay et al. [2008] and Leutenegger et al. [2011] use scale-space representation to localize keypoints at different scales. The image is represented by a multilevel, multiscale pyramid in which for each level the image is smoothed and sub-sampled.

Since RGB-D images are composed of color as well as depth information, instead of computing a pyramid and representing the keypoints in scale-space, we use the depth information of each keypoint to define the scale factor s of the patch to be used in the neighborhood analysis. In this way, patches associated with keypoints farther away from the camera will present smaller sizes.

The scale factor s is computed by the function:

$$s = \max\left(0.2, \frac{3.8 - 0.4\max(d_{\min}, d)}{3}\right), \tag{3.1}$$

which linearly scales the radius of a circular patch $\mathbf{P}$ from 9 to 24, and filters depths with values less than $d_{\min}$ (in this work we use $d_{\min} = 2$ meters).

3.1.2 Canonical Orientation Estimation

There are several algorithms to determine the canonical orientation of a keypoint. We tested three methods to be used in our descriptors: Intensity Centroid (IC), SURF-like (HAAR) and SIFT-like (BIN). Our choice was based on the stability and simplicity of the techniques, since they are robust and have small processing time.

Intensity Centroid (IC) The canonical orientation of a keypoint K can be estimated by a fast and simple method using geometric moments. The idea behind this is to build a vector from the keypoint's localization to the centroid patch defined by the moments in the region around the keypoint.

Rosin [1999] defines the moments of a patch of a image I as:

$$m_{pq} = \sum_{x,y} x^p y^q I(x, y). \tag{3.2}$$

Similar to Rublee et al. [2011], we compute the moments m using only pixels (x, y) remaining within a circular region of radius equal to the patch size.

The patch centroid C is determined by:

$$C = \left(\frac{m_{10}}{m_{00}}, \frac{m_{01}}{m_{00}} \right), \tag{3.3}$$

and the canonical orientation θ is given by the angle of the vector $\vec{KC}$:

$$\theta = \text{atan2}(m_{01}, m_{10}), \tag{3.4}$$

where atan2 is a implementation of the quadrant-aware version of the arctangent function.

SURF-like (HAAR) This algorithm identifies the direction of keypoints using a process more robust to noise based on Haar wavelets responses and a sliding orientation window. In order to estimate the canonical orientation θ, a circular neighbourhood of radius $6s$, where s is the scale at which the keypoint was detected, is centered around each keypoint.

Thus, for both x and y directions the Haar wavelet responses, with the wavelets sizes set to $4s$, are calculated. These responses are weighted with a Gaussian function centred at the keypoint. These values are plotted in a graph with he x direction response strength along the abscissa and the y direction response strength along the ordinate axis. Finally, a sliding orientation window of size $\pi/3$ is used to produce

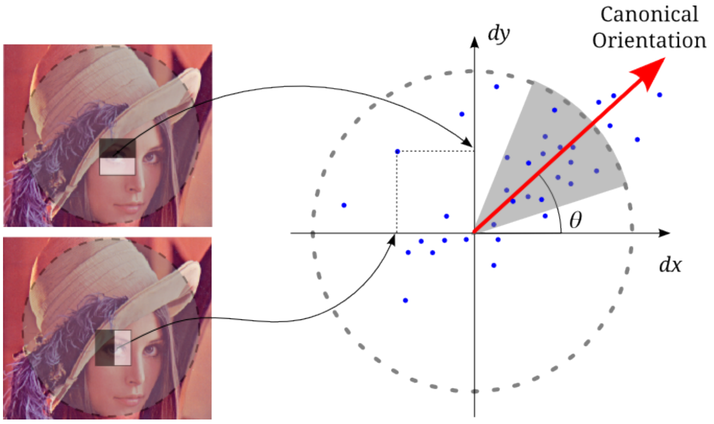


Figure 3.2. Computing SURF canonical orientation θ. In a circular neighbourhood of radius $6s$, two Haar wavelets filters with size $4s$ are used to compute the responses in x and y direction (left image). The responses are plotted in a graph (blue points) and summed. The largest vector (red vector) defines the canonical orientation (right image).

the keypoint's orientation. The responses in the x and y axes are added, yielding an orientation vector within the window. The canonical orientation is chosen as the vector with the largest magnitude. Figure 3.2 depicts this procedure.

SIFT-like (BIN) The third method is similar to the one used in the SIFT algorithm. A histogram with 36 directions is formed by taking values within a region around the keypoint's location. An accumulation is performed adding the values of $m(x, y)$ computed using Equation 2.3 and weighted by a Gaussian function around the keypoint. The orientation of each pixel $\theta(x, y)$ is computed by Equation 2.4. The highest peak of the histogram determines the canonical orientation of the local gradients. However, differently from SIFT's algorithm which includes more keypoints when there are other peaks within 80% of the highest, we pick only a single orientation.

3.1.3 Appearance and Geometry Fusion

The importance of combining shape and visual information comes from the possibility of creating descriptors robust to textureless objects, lack of illumination and scenes with ambiguous geometry.

Our fusion process is divided into three main steps: In the first step, to exploit the appearance information, we extract the visual features based on the direction of the gradient around a keypoint. The idea behind this step is similar to the one used by the Local Binary Patterns (LBP) [Ojala et al., 1996]. Then, we build a point cloud with the depth information and extract the features based on its normal surfaces.

Finally, we combine the result of this analysis in a unique vector which represents the signature of the keypoint.

In the next sections we will detail these steps and assemble three novel descriptors using our methodology as a design guide. The texture analysis step is shared by all the descriptors, therefore we will present it as follows.

Appearance Analysis The gradient directions are computed using simple intensity difference tests, which have small memory consumption and modest processing time. Given an image keypoint $\mathbf{k} \in \mathcal{K}$, assume a circular image patch $\mathbf{P}$ of size $W \times W$ (in this work we consider $9 \leq W \leq 48$) centered at $\mathbf{k}$. We use a fixed pattern with locations given by distribution function $\mathcal{D}(\mathbf{k})$ for sampling pixel pairs around the keypoint $\mathbf{k}$. We also smooth the patch with a Gaussian kernel with $\sigma = 2$ and a window with 9×9 pixels to decrease the sensitivity to noise and increase stability in the pixel comparisons.

Let the fixed set of sampled pairs from $\mathbf{P}$ be $S = \{(\mathbf{x}_i, \mathbf{y}_i), i = 1, \ldots, 256\}$. Before constructing the visual feature descriptor, the patch $\mathbf{P}$ is translated to the origin and then rotated and scaled by the transformation $\mathbf{T}_{\theta,s}$, which produces a set P, where

$$P = \{(\mathbf{T}_{\theta,s}(\mathbf{x}_i), \mathbf{T}_{\theta,s}(\mathbf{y}_i)) | (\mathbf{x}_i, \mathbf{y}_i) \in S\}. \tag{3.5}$$

This transformation normalizes the patch to allow comparisons between patches. Then, for each pair $(\mathbf{x}_i, \mathbf{y}_i) \in P$, we evaluate

$$\tau_a(\mathbf{x}_i, \mathbf{y}_i) = \begin{cases} 1 & \text{if } p_i(\mathbf{x}_i) < p_i(\mathbf{y}_i) \\ 0 & \text{otherwise,} \end{cases} \tag{3.6}$$

where the comparison term captures gradient changes in the keypoint neighborhood.

3.2 EDVD Descriptor

In this section, we present our descriptor called Enhanced Descriptor for Visual and Depth Data (EDVD). After extracting the gradient features according to our methodology, we group the results of eight tests and represent it as a floating point number. Therefore, we can use a vector $\mathbf{V}_a$ with 32 elements to store the results of all 256 comparisons performed by the function τ_a (Equation 3.6).

The EDVD's approach builds a rotation invariant representation based on the direction of the normals using an extended Gaussian image followed by the appli-

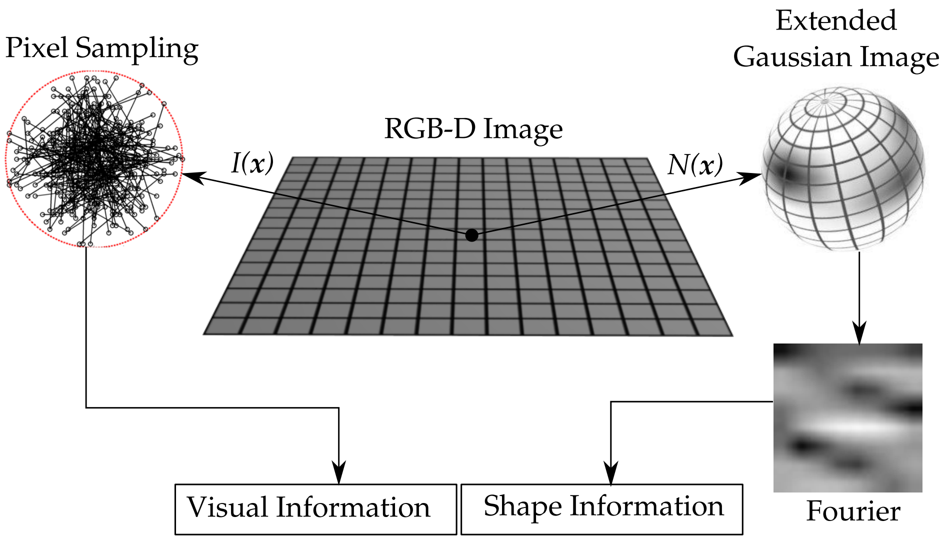


Figure 3.3. The proposed descriptor combines shape and visual information based on invariant measurement in both domains.

cation of the Fourier transform. This process is illustrated in Figure 3.3.

Geometrical Feature Analysis We use orientation histograms to capture the geometric characteristics of the patch $\mathbf{P}$ in the 3D domain. Since orientation histograms are approximations of Extended Gaussian Images (EGI) [Horn, 1984], they constitute a powerful representation invariant to translational shift transformations.

The first step in orientation histograms creation is to represent each normal $p_n(\mathbf{x})$ in spherical coordinates (ϕ, ω) (Figure 3.4). These angles are compute as [Hetzel et al., 2001]:

$$\phi = \arctan\left(\frac{n_z}{n_y}\right), \omega = \arctan\left(\frac{\sqrt{n_y^2 + n_z^2}}{n_x}\right). \tag{3.7}$$

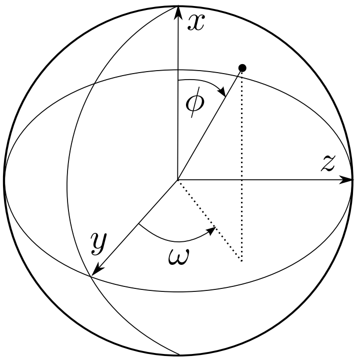


Figure 3.4. Representation of normals in ϕ and ω sphere coordinates.

Then, the coordinates ϕ and ω are discretized into 8 values each, and the number of normals falling inside each discretized orientation is accumulated. Figure 3.3 depicts the accumulation of normal directions in the sphere. Dark spots represent a large number of normals accumulated in that orientation.

Since rotations in the normal orientations become translations in the EGI domain, we apply the Fourier transform in the EGI domain to obtain a translation invariant Fourier spectrum. Finally, the Fourier spectrum is linearized and converted to a 64-dimension vector $\mathbf{V}_s$. In addition to the rotation invariance, the use of spectral information emphasizes differences among different descriptors.

Fusion Process Once the visual and geometrical features have been extracted, we concatenate the geometrical vector $\mathbf{V}_g$ and the appearance vector $\mathbf{V}_a$, creating a 96-dimension vector which captures both appearance and geometrical information.

Despite the high quality of matching and invariance to rotation transforms, EDVD algorithm has drawbacks in the processing time to create the EGI histograms and the vector size. Furthermore, the EDVD vectors are compared using correlation function, which is slower than other approaches such as Hamming distance.

3.3 BRAND Descriptor

In the following paragraphs we detail the design of our second descriptor, which we call BRAND from Binary Robust Appearance and Normal Descriptor. Throughout this section are shown the descriptor's characteristics and how they cover all the properties in Π set as well as how it overcomes the EDVD deficiencies.

Geometrical Feature Analysis and Fusion Information There are several choices available to compose a descriptor, and bit strings are among the best approaches, mainly due to the reduction in dimensionality and efficiency in computation achieved with their use. One of the greatest advantages of using a binary string as descriptors, besides its simplicity, is its low computational cost and memory consumption, whereas each descriptor comparison can be performed using a small number of instruction on modern processors. For instance, modern architectures have only one instruction (POPCNT) to count the number of bit sets in a bit vector [Intel, 2007].

Although our descriptor encodes point information as a binary string, similar to approaches described in [Calonder et al., 2010; Leutenegger et al., 2011; Rublee

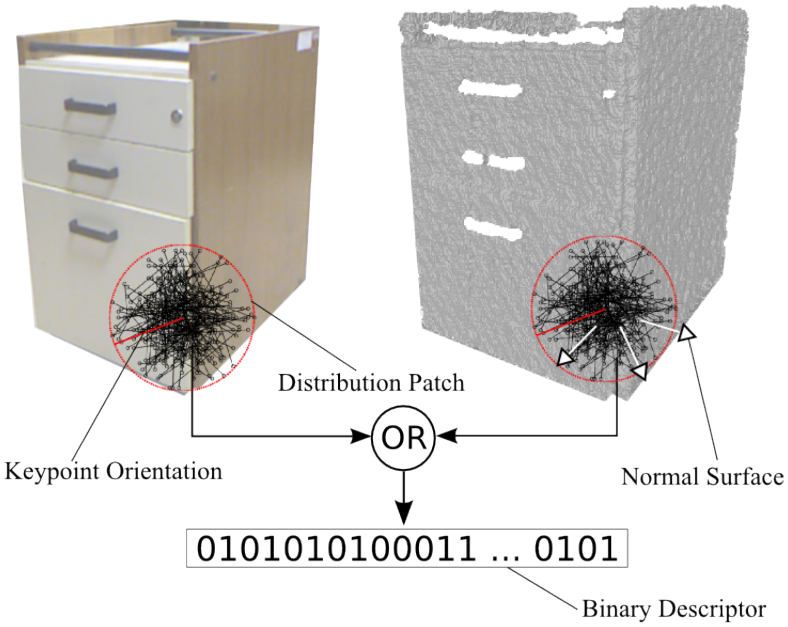


Figure 3.5. Binary descriptor diagram. The patch of size $S \times S$ is centered at the location of keypoint. For all positions in a set of $(\mathbf{x}, \mathbf{y})$-locations the intensity changes in image and the displacement of normals inside of projected patch in the point cloud is evaluated.

et al., 2011; Ambai and Yoshida, 2011], we embed geometric cues into our descriptor to increase robustness to changes in illumination and the lack of texture in scenes.

Following the steps in our methodology, unlike EDVD that builds an EGI histogram and uses concatenation operator to form the final vector, BRAND evaluates the function 3.8 for each pair $(\mathbf{x}_i, \mathbf{y}_i) \in P$:

$$f(\mathbf{x}_i, \mathbf{y}_i) = \begin{cases} 1 & \text{if } \tau_a(\mathbf{x}_i, \mathbf{y}_i) \vee \tau_g(\mathbf{x}_i, \mathbf{y}_i) \\ 0 & \text{otherwise,} \end{cases} \tag{3.8}$$

where the function $\tau_a(.)$ (Equation 3.6) captures the characteristic gradient changes in the keypoint neighborhood and $\tau_g(.)$ function evaluates the geometric pattern on its surface. Figure 3.5 illustrates the construction process of the bit string.

The analysis of the geometric pattern using $\tau_g(.)$ is based on two invariant geometric measurements: i) the normal displacement (Figure 3.6 illustrates two possible cases of normal displacement for a pair (x, y)) and ii) the surface's convexity. While the normal displacement test is performed to check if the dot product between the normals $p_n(\mathbf{x_i})$ and $p_n(\mathbf{y_i})$ is smaller than a displacement threshold ρ, the convexity test is accomplished by the local curvature signal, κ, estimated as:

$$\kappa(\mathbf{x}_i, \mathbf{y}_i) = \langle p_s(\mathbf{x}_i) - p_s(\mathbf{y}_i), p_n(\mathbf{x}_i) - p_n(\mathbf{y}_i) \rangle, \tag{3.9}$$

where $\langle . \rangle$ is the dot product and $p_s(\mathbf{x})$ is the $3D$ spatial point associated to the pixel $\mathbf{x}$ and the depth $D(\mathbf{x})$. Figure 3.7 illustrates an example where the dot product between surface normals is ambiguous, since $\theta_1 = \theta_2$, but different signed curvatures, $\kappa_1 < 0$ and $\kappa_2 > 0$, are used to unambiguously characterize these different shapes, besides capturing convexity as additional geometric features.

The final geometric test is given by:

$$\tau_g(\mathbf{x}_i, \mathbf{y}_i) = (\langle p_n(\mathbf{x}_i), p_n(\mathbf{y}_i) \rangle < \rho) \wedge (\kappa(\mathbf{x}_i, \mathbf{y}_i) < 0)\,. \tag{3.10}$$

Finally, the descriptor extracted from a patch $\mathbf{p}$ associated with a keypoint $\mathbf{k}$ is

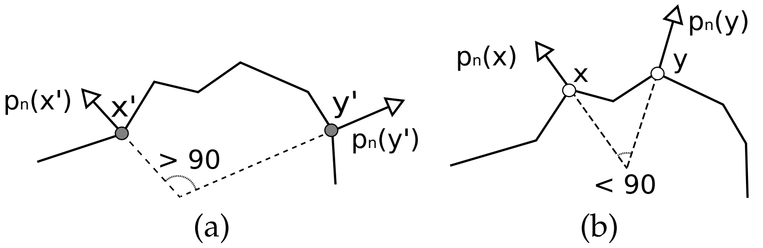


Figure 3.6. Image (a) shows a surface where the normal displacement of points $\mathbf{x}'$ and $\mathbf{y}'$ is greater than 90 degrees leading to bit value 1. In image (b) is shown the normals of points $\mathbf{x}$ and $\mathbf{y}$ that lead to bit 0 due to displacement less than 90 degrees.

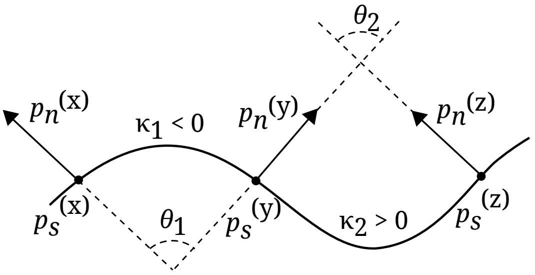


Figure 3.7. Example of ambiguity in the dot product. Despite the fact that the points $p_s(\mathbf{x})$ and $p_s(\mathbf{y})$ define a concave surface patch and $p_s(\mathbf{y})$ and $p_s(\mathbf{z})$ define a convex surface patch, the dot products $\langle p_n(\mathbf{x}), p_n(\mathbf{y}) \rangle = \langle p_n(\mathbf{y}), p_n(\mathbf{z}) \rangle$. In such cases, the curvature signals $\kappa 1 < 0$ and $\kappa 2 > 0$ are used to unambiguously characterize the patch shape.

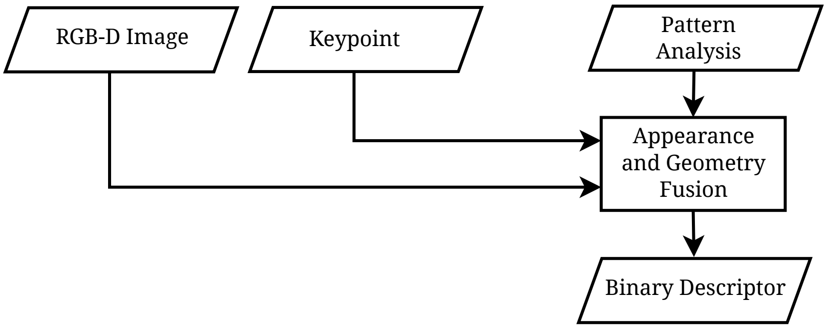


Figure 3.8. BASE diagram. The appearance and geometric information are fused based on the features selected with a pattern analysis.

encoded as a binary string computed by:

$$b(\mathbf{k}) = \sum_{1}^{256} 2^{i-1} f(\mathbf{x}_i, \mathbf{y}_i). \tag{3.11}$$

Once the descriptors $b(\mathbf{k}_1)$ and $b(\mathbf{k}_2)$ have been estimated for two keypoints $\mathbf{k}_1$ and $\mathbf{k}_2$, they are compared using the Hamming distance as

$$h(b(\mathbf{k}_1), b(\mathbf{k}_2)) = \sum_{1}^{256} 2^{-(i-1)}(b(\mathbf{k}_1) \oplus b(\mathbf{k}_2)) \wedge 1. \tag{3.12}$$

3.4 BASE descriptor

Not all applications require scale and rotation invariance. For these applications our BRAND descriptor can turn off the invariance properties removing the orientation and scale transformation estimation phases. The new simplified descriptor, called Binary Appearance and Shape Elements (BASE), uses a circular patch with a fix radius of size 24 to select pairs of pixels and normals in the point cloud. In constrast to BRAND and EDVD, BASE does not compute the canonical orientation. Figure 3.8 shows the BASE diagram. Similar to BRAND, the gradient information and geometrical features (based on the normal displacements) are combined using function 3.8.

One of the benefits of this version is that it requires modest computational costs, since the steps to compute the canonical orientation and the keypoint scale are not performed. In spite of the simplicity of our descriptor, our experiments have shown robustness against small rotation and scale changes.

3.5 Invariant Measurements of BRAND and BASE

An important characteristic of the approach that we adopted to use geometry from RGB-D images is the relation between normal's displacement and the transformations of rotation, scale and translation.

To prove the invariace properties of our approach we will present some important definitions of invariance measurements in geometry [Andrade and Lewiner, 2011].

Let S be a geometric object and A a transformation.

Definition 3.1 (Invariant Measurement). *A geometric measurement is invariant if* $\forall S, \forall A, m(A(S)) = m(S)$*, e.g surface curvature.*

Definition 3.2 (Covariant Measurement). *A geometric measurement is covariant if* $\forall S, \forall A, m(A(S)) = A(m(S))$*, e.g tangent vector.*

Definition 3.3 (Contravariant Measurement). *A geometric measurement is contravariant if* $\forall S, \forall A, m(A(S)) = A^{-1}(m(S))$*, e.g normal vector.*

Lemma 3.1. *Orthogonal transformations preserve the dot product.*

Proof. Let A be an orthogonal transformation and $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$:

$$\begin{aligned} \langle A\mathbf{x}, A\mathbf{y} \rangle &= (A\mathbf{x})^T (A\mathbf{y}) \\ &= (\mathbf{x}^T A^T)(A\mathbf{y}) \\ &= \mathbf{x}^T (A^T A)\mathbf{y} \\ &= \mathbf{x}^T I \mathbf{y} = \mathbf{x}^T \mathbf{y}. \end{aligned}$$

□

Lemma 3.2. *The length of vector is preserved under orthogonal transformations.*

Proof. Let A be an orthogonal transformation and $\mathbf{x} \in \mathbb{R}^n$:

$$\begin{aligned} \|A\mathbf{x}\|^2 &= (A\mathbf{x})(A\mathbf{x}) \\ &= \mathbf{x}^T A^T A \mathbf{x} \\ &= \mathbf{x}^T I \mathbf{x} \\ &= \mathbf{x}^T \mathbf{x} = \|\mathbf{x}\|^2. \end{aligned}$$

□

Lemma 3.3. *The angle between two vectors is preserved under orthogonal transformations.*

Proof. Let α be the angle between vectors $\mathbf{x}$ and $\mathbf{y}$ and let β be angle between the transformed vectors, $A\mathbf{x}$ and $A\mathbf{y}$. According to Lemma 3.1, $\langle A\mathbf{x}, A\mathbf{y}\rangle = \langle \mathbf{x}, \mathbf{y}\rangle$, thus:

$$\|A\mathbf{x}\|\|A\mathbf{y}\|\cos(\beta) = \|\mathbf{x}\|\|\mathbf{y}\|\cos(\alpha).$$

Also, according to Lemma 3.2, $\|A\mathbf{x}\| = \|\mathbf{x}\|$ and $\|A\mathbf{y}\| = \|\mathbf{y}\|$, consequently $\cos(\beta) = \cos(\alpha)$. Let V be a plane spanned by $\mathbf{x}$ and $\mathbf{y}$, and let ϕ be the angle of rotation for vector the $\mathbf{x}$ in V plane. For all ϕ, since $\cos(\beta) = \cos(\alpha)$, $\cos(\beta + \phi) = \cos(\alpha + \phi)$. Differenting with respect to ϕ, we obtain:

$$-\sin(\beta + \phi) = -\sin(\alpha + \phi),$$

for $\phi = 0$, $\sin(\beta) = \sin(\alpha)$, which implies $\beta = \alpha$. □

Theorem 3.1. *The BRAND and BASE measurement m_b is invariant under rigid transformations in the depth space.*

Proof. The group of transformations considered is composed of rotation, translation and uniform scaling. We will show that BRAND and BASE measurement is invariant to all these rigid transformations.

- Rotation: Let m_b be the geometric measurement used in BRAND descriptor and A is a rotation matrix. We will show that $m_b(\mathbf{x}, \mathbf{y}) = m_b(A\mathbf{x}, A\mathbf{y})$, where $\mathbf{x}, \mathbf{y} \in \mathbb{R}^3$. $m_b(\mathbf{x}, \mathbf{y}) = \langle \mathbf{x}, \mathbf{y}\rangle$ and according to Lemma 3.1, orthogonal matrix preserves the dot product, as every rotation matrix is orthogonal, $\langle A\mathbf{x}, A\mathbf{y}\rangle = \langle \mathbf{x}, \mathbf{y}\rangle$.

- Translation: Let $\mathbf{x}$ be a normal vector of surface S. Let $p, q \in S$ two points that define the normal $\mathbf{x}$, $\mathbf{x} = p - q$. Applying a translation A to the surface S using a vector $\mathbf{t}$, p and q can be rewrite as:

$$\begin{aligned} p' &= p + \mathbf{t} \\ q' &= q + \mathbf{t}, \end{aligned}$$

the normal $\mathbf{x}$, after applying A is $\mathbf{x} = p' - q'$,

$$\begin{aligned}\mathbf{x}' &= p + \mathbf{t} - (q + \mathbf{t}) \\ &= p + \mathbf{t} - q - \mathbf{t} \\ &= p - q \\ &= \mathbf{x}.\end{aligned}$$

- Scale: Finally, to provide invariance in scale transformations, all normals used by BRAND are normalized. Indeed, if A is a uniform scale transform, $A(\mathbf{x}) = s\mathbf{x}$, therefore
$$\frac{s\mathbf{x}}{||s\mathbf{x}||} = \frac{s\mathbf{x}}{s * 1} = \mathbf{x}.$$
□

Theorem 3.1 shows that our approach provides a way to extract features from an object's geometry that do not suffer interference from rotation, scale and translation transformations.

3.6 Rating EDVD, BRAND and BASE based on the Π set

Table 3.1 shows the classification of EDVD, BRAND and BASE descriptor according to the properties from Π set. Note that all the properties are covered by BRAND and BASE presents a clear improvement on the BRIEF approach for textureless scenarios. In the independence detection property, the descriptors were rated according to results presented in Chapter 4.

	SURF	SIFT	Spin-Image	CSHOT	EDVD	BRAND	BASE
Robustness to noise	●	●	—	●	●	●	●
Scale invariance	●	●	—	●	●	●	—
Rotation invariance	●	●	●	●	●	●	—
Illumination invariance	○	○	●	●	●	●	●
Texture independence	—	—	●	●	●	●	●
Low time to compute	●	—	—	—	○	●	●
Low time to compare	○	○	—	—	—	●	●
Low memory consumption	—	—	—	—	—	●	●
Detection independence	○	○	—	—	○	●	●

Table 3.1. Properties of descriptors EDVD, BRAND and BASE.

Chapter 4

Experiments

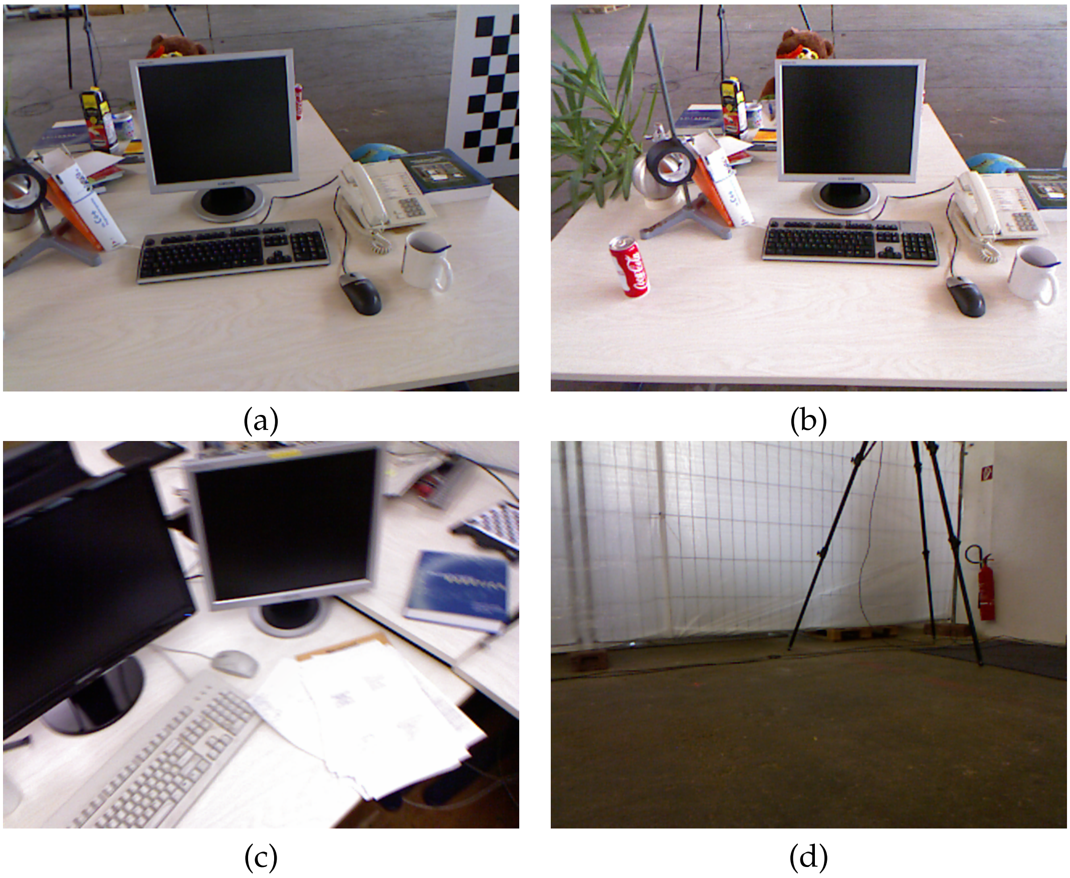
IN THIS CHAPTER WE DESCRIBE A SET OF EXPERIMENTS to analyze the behavior of our descriptors for matching tasks. Comparisons are performed with the standard approaches of the two-dimensional images descriptors, SIFT [Lowe., 2004] and SURF [Bay et al., 2008], with the geometric descriptor, spin-images [Johnson and Hebert, 1999], and with CSHOT [Tombari et al., 2011], the state-of-the-art approach in fusing both texture and shape information.

For the experiments we use the dataset presented in [Sturm et al., 2011]. This dataset is publicly available[1] and contains several real world sequences of RGB-D data captured with a Kinect$^{\text{TM}}$. Images were acquired at a frame rate of 30 Hz and a resolution of 640×480 pixels. Each sequence in the dataset provides the ground truth for the camera pose estimated by a motion capture system. We selected four sequences in the dataset to use in our experiments:

- *freiburg2_xyz*: Kinect sequentially moved along the x/y/z axes;
- *freiburg2_rpy*: Kinect sequentially rotated around the three axes (roll, pitch and yaw rotations);
- *freiburg2_desk*: A handheld SLAM sequence with Kinect;
- *freiburg2_pioneer_slam2*: A SLAM sequence with a Kinect mounted on the top of a Pioneer mobile robot.

Figure 4.1 shows a frame sample from each sequence.

[1] https://cvpr.in.tum.de/data/datasets/rgbd-dataset



(a) (b) (c) (d)

Figure 4.1. Frame samples from (a) *freiburg2_xyz*, (b) *freiburg2_rpy*, (c) *freiburg2_desk* and (d) *freiburg2_pioneer_slam2*.

To evaluate the performance of our descriptors and to compare with other approaches, we applied the same criterion used by Ke and Sukthankar [2004] and Mikolajczyk and Schmid [2005].

Using a brute force algorithm, we matched all pairs of keypoints from two different images. If the Euclidean (for SURF and SIFT), Correlation (for Spin-image and EDVD), Cosine (for CSHOT) or Hamming (for BRAND and BASE) distance computed between descriptors dropped below a threshold t, the pair was considered as a *valid match*. The number of *valid matches* which have two keypoints correspond to the same physical location (as determined by ground truth) defines the *true positives* matches. On the other hand, if the keypoints in a *valid match* come from different physical locations, then we increment the number of *false positives*. From these values, we compute the $recall$ and $1 - precision$.

The $recall$ values were determined by:

$$recall = \frac{\#truepositive}{\#total \quad of \quad positives},$$

where the total of positives is given by the dataset. The $1 - precision$ were computed as

$$1 - precision = \frac{\#falsepositive}{\#truepositive + \#falsepositive},$$

when the number of valid matches (true positives + false positives) is higher than zero, otherwise we assign zero to $1 - precision$. Using that information, we plotted the recall versus $1 - precision$ values, obtained by changing the values of t.

We also use Area Under Curve (AUC) of $recall$ vs $1 - precision$ curves in the parameter settings analysis where it is more clear to show our design decisions. For a fair comparison, the AUC values were computed for the curves with their intervals extrapolated using the point with the highest Recall value. Furthermore, the AUC measure was computed for only the well-behaved curves, *e.g.* the red and blue curves shown in Figure 4.4 (b). For curves like the green one in Figure 4.4 (b), that are clearly worse than the others and misbehave, we did not compute AUC values.

In the match experiments, for each sequence, given an RGB-D frame i, we computed a set of keypoints $\mathcal{K}_i$ using the STAR detector[2]. Using the groundtruth camera trajectory provided by the dataset, we transformed all keypoints $\mathbf{k} \in \mathcal{K}_i$ to frame $i + \Delta$ creating the second set $\mathcal{K}_{i+\Delta}$. We computed a descriptor for each keypoint in both sets and then perform the match.

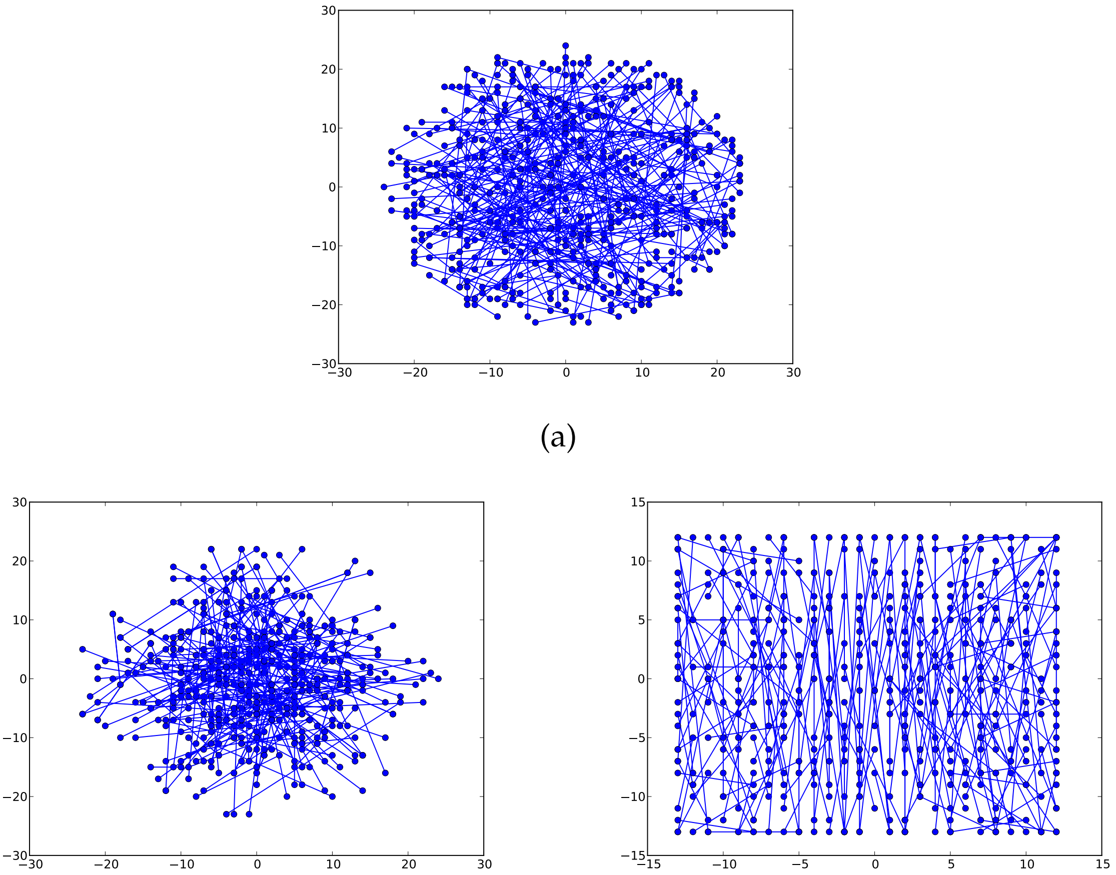
In the following sections, we evaluate and compare computation time, memory consumption and accuracy of EDVD, BRAND and BASE against other descriptors.

4.1 Parameter Settings

In this section we analyze what the best parameter values to be used by our three descriptors are. All of the following experiments were performed using the *freiburg2_xyz* sequence from the RGB-D SLAM dataset.

Pairs distribution inside the patch Our algorithms perform an analysis in the neighborhood around the keypoint (in image and depth domain). This analysis is based on a set of pixels selected by a distribution function $\mathcal{D}$. We tested three different distributions and the pattern of each is illustrated in Figure 4.2. Assuming

[2] The STAR detector is an implementation of the Center Surrounded Extrema [Agrawal et al., 2008] in OpenCV 2.3.



(a)

(b)

(c)

Figure 4.2. (a) Uniform distribution; (b) Isotropic Gaussian distribution and (c) Learned distribution.

the origin of the patch coordinate system located at the keypoint, we selected 256 pairs of pixels using the following distributions:

- A uniform distribution $U(-24, 24)$;
- An isotropic Gaussian distribution $\mathcal{N}(0, \frac{24^2}{25})$;
- A distribution created by Rublee et al. [2011].

The latter distribution was built using a learning method to reduce the correlation among pairs of pixels. Also, all pixels $(\mathbf{x_i}, \mathbf{y_i})$ outside the circle with radius $r = 24$ are removed in the uniform and Gaussian distributions to guarantee that all pixels within the circle are preserved independently of patch rotation.

	Accurate	**Fast**
Time in seconds	14.85	0.11

Table 4.1. Average processing time (over 300 point clouds) to compute normal surfaces from point cloud with 640×480 points.

Canonical Orientation We tested three algorithms to compute the canonical orientation and have provided a more detailed description of these in Chapter 3. The first of these algorithms, called Intensity Centroid (IC) [Rosin, 1999], computes the θ orientation using the orientation of a vector defined by the patch's center and its centroid. The second algorithm, which we call HAAR, is based on the fast estimator presented in [Bay et al., 2008]. The orientation assignment for each keypoint is achieved by computing the Haar wavelet responses in both the x and y directions. The third algorithm, which we call BIN, is a modified version of the SIFT algorithm. It creates a histogram of gradient directions, but unlike SIFT, it chooses only the maximum bin as the canonical orientation.

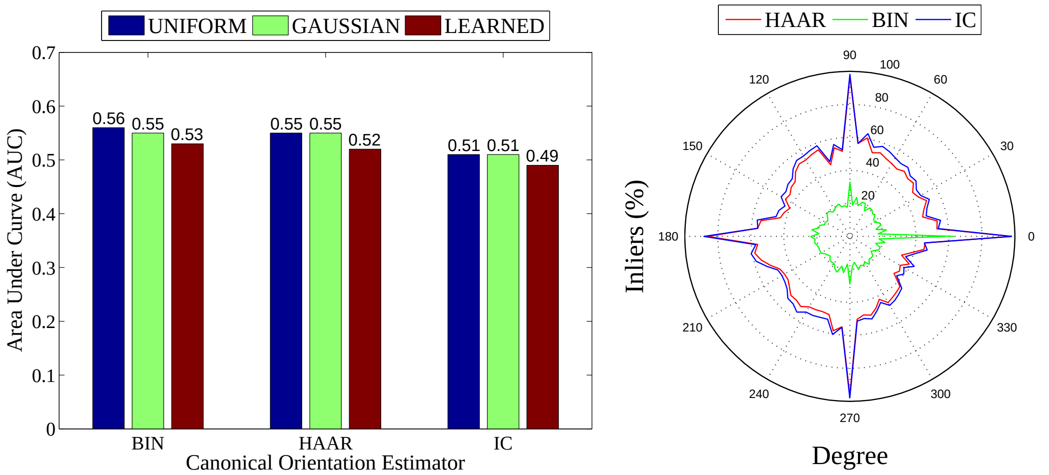
Normal Surface Estimation All of the geometric descriptors used for comparison in the experiments require that the point clouds have normals. There are several methods to estimate these normals from a point cloud [Klasing et al., 2009]. One accurate approach consists of estimating the surface normal by PCA from a covariance matrix created using the nearest neighbors of the keypoint [Berkmann and Caelli, 1994]. This was the method used to estimate normals in all of the match experiments.

A less accurate, but faster approach, is to use the pixel neighborhoods defined by the structure from RGB-D images [Holz et al., 2011]. Using two vectors, *e.g.* the left and right neighboring pixel and upper and lower neighboring pixel, the algorithm computes the cross product to estimate the normal surface. Table 4.1 shows the processing time to compute all normal surfaces from a typical point cloud with 640×480 points. The less accurate approach is more 100 times faster than accurate one.

4.1.1 EDVD descriptors

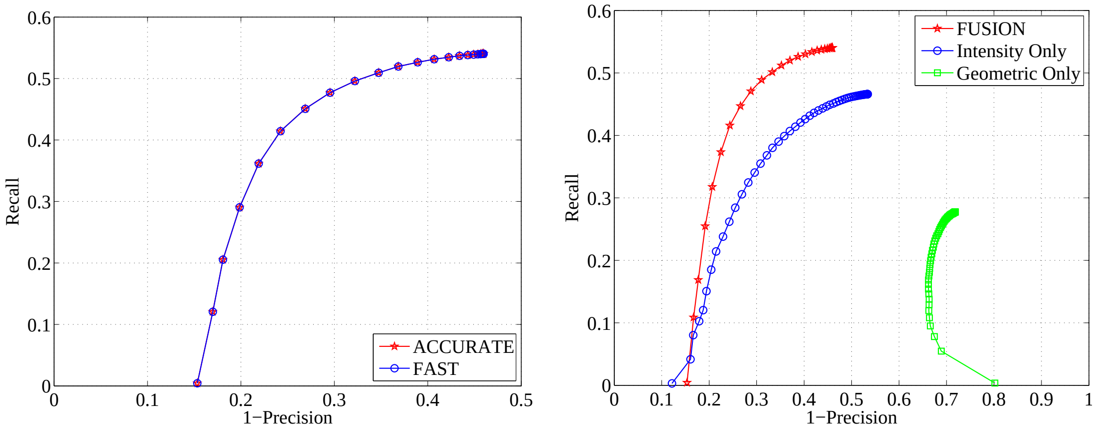
Experimentally, we found that the combination of a Gaussian distribution and the HAAR algorithm is the best configuration for EDVD. We can readily see in Figure 4.3 that the HAAR algorithm provides a more stable invariance to rotation and the highest AUC values when combined with the Gaussian distribution. Therefore, we chose to use the HAAR algorithm in the canonical orientation and a Gaussian distribution to select the pairs of pixels.

Additionally, we carried out several experiments to verify the influence of the normal surface algorithm in the accuracy of EDVD. Figure 4.4 (a) shows that EDVD provides the same accuracy independent of the algoritm used. In Figure 4.4 (b) we can also see that, after fusing texture and geometrical information, the accuracy increases.



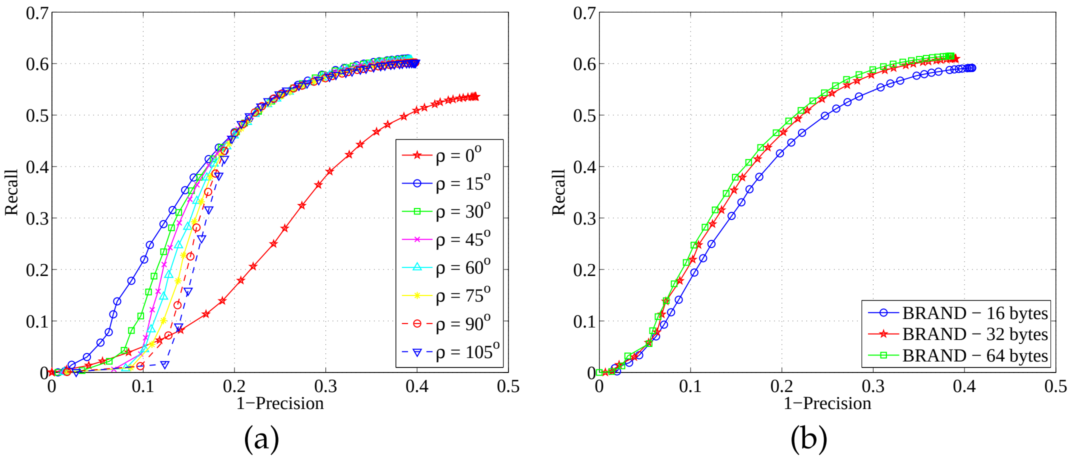
(a) (b)

Figure 4.3. Parameter analysis of EDVD descriptor. (a) The match performance using 9 combinations of 3 distributions and 3 algorithms to estimate the canonical orientation; (b) Invariance to orientation with the 3 algorithms used to estimate the canonical orientation (using Gaussian distribution).



(a) (b)

Figure 4.4. (a) Accurate *versus* Fast normal estimation; (b) Combining texture and geometrical information to increase the accuracy.



(a) (b)

Figure 4.5. (a) Angular threshold for the dot product test. On average, the best choice is to use 15 degrees. (b) Different sizes for the BRAND descriptor.

4.1.2 BRAND and BASE descriptors

For BRAND and BASE, we tested different configurations of the angular displacement threshold ρ as well as the size and the best binary operator to be used in the information fusion step. Since BASE is a special case of BRAND, we perfomed all test in this section using BRAND only.

Experimentally, we found that a threshold ρ, corresponding to 15 degrees for the maximum angular displacement of normals results in a larger number of inliers (Figure 4.5 (a)). The plot shown in Figure 4.5 (b) depicts the accuracy *versus* the number of bytes used for the BRAND descriptor. Moreover, the results show that the accuracy for 32 bytes is similar to the accuracy for 64 bytes. Therefore, in order to obtain a more compact representation, we have chosen to use 32 bytes in the experiments.

Figure 4.6 shows the matching accuracy and the time spent by BRAND using both normal estimation techniques. We can see that, even with a less precise normal estimation, BRAND presents high accuracy in the correspondences. This shows that BRAND can be optimized if necessary for a given application without significantly penalizing its accuracy.

Binary Operator We chose to use a bit operator to combine appearance with geometry in order to maintain the simplicity and computational efficiency of the descriptor. To fuse the required information, we evaluated different operators such as XOR, AND, and OR, and the best result was obtained using the OR operator (Figure 4.7).

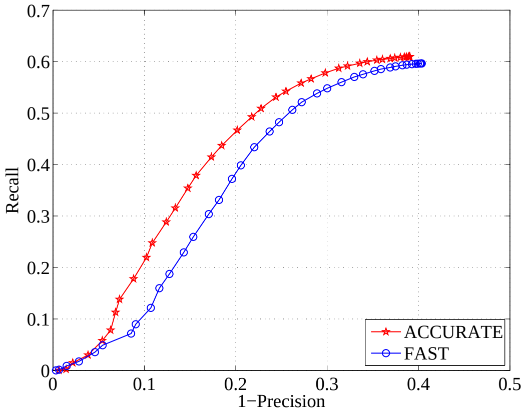


Figure 4.6. Accurate *versus* Fast normal estimation. Even with the less precise normal estimation, BRAND still had high accuracy in keypoint correspondences.

We also performed experiments with larger signatures to separately handle intensity and normal by concatenating them in order to avoid ambiguity. It can see clearly in Figure 4.7 (a), that fusing both texture and geometrical information provides a signature with better discriminative power than concatenating these features. The use of information from two different domains has the disadvantage of being exposed to two different sources of noise. However, using a binary operator rather than concatenation, our descriptors are able to balance noise in one domain using other kinds of information.

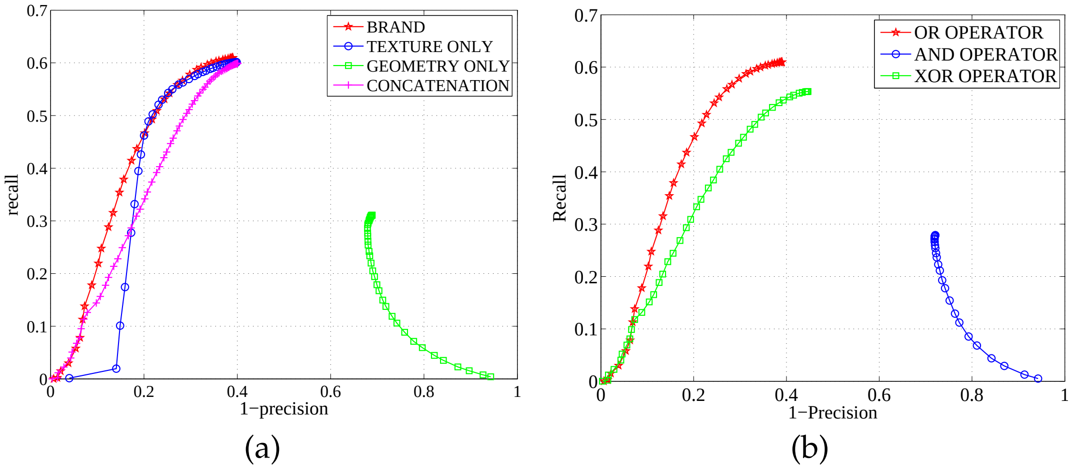


Figure 4.7. (a) Accuracy with OR operator, only intensity, only geometrical information and concatenating intensity and geometrical features; (b) The best binary operator to be used for fusing appearance and geometric was the OR operator.

		1	2	3	4	5	6	7	8	9
D_1	*Normal*	0	**0**	0	**1**	1	1	1	1	1
	Intensity	1	**1**	1	**0**	0	0	1	1	1
D_2	*Normal*	0	**1**	1	**0**	1	1	0	1	1
	Intensity	1	**0**	1	**1**	0	1	1	0	1

Table 4.2. This table shows all 9 cases that can produce $D_1 = 1$ and $D_2 = 1$. For all theses cases only 2 can be ambiguous (columns 2 and 4 in bold). Changes in normal or intensity are represented with bit equal to 1.

Binary Operator versus Concatenation One of the problems with using binary operators to define bits of the descriptors is its ambiguity. We do not know if a bit was set to 1 due to a variation in the normal or intensity.

Let D_1 and D_2 be two descriptors each of one bit size and the operator OR. For these two descriptors there are four possible cases, outlined as follows:

- $D_1 = 0$, $D_2 = 0$: There is no normal variation in the surface or intensity variation in the image determines a bit equals to zero;
- $D_1 = 0$, $D_2 = 1$: There is no normal or intensity variation in surface reported by descriptor D_1, but some variation was detected by D_2 (normal or intensity);
- $D_1 = 1$, $D_2 = 0$: Similar case as the previous except with variation detected by D_1;
- $D_1 = 1$, $D_2 = 1$: Both descriptors reported some variation.

In the latter case, the source of variation can be different and thus the descriptors should be different. This is the case that may have ambiguity. Table 4.2 shows all 9 cases that can produce $D_1 = 1$ and $D_2 = 1$ and 2 of them set the bit to 1 when they should be set to 0. These cases are shown in bold in the Table 4.2. This occurs when there are changes in normal direction but none in intensity of the surface that generated descriptor D_1 and the surface that produce D_2 does not have variation in the normal directions but has changes in intensity. Thus, the probability of comparing ambiguous bits is $\frac{1}{4} \times \frac{2}{9} = 5.6\%$. In practice, the ambiguity is smaller. We computed for 420 keypoints in 300 pairs of images the number of ambiguities. We have found the rate to be equal to 0.7%.

The probability of ambiguity with the XOR operator is higher than with OR. When using the XOR operator, D_1 and D_2 will be set to 1 in 4 cases and two of them will produce ambiguity: i) There is a variation only in intensity for D_1, and for

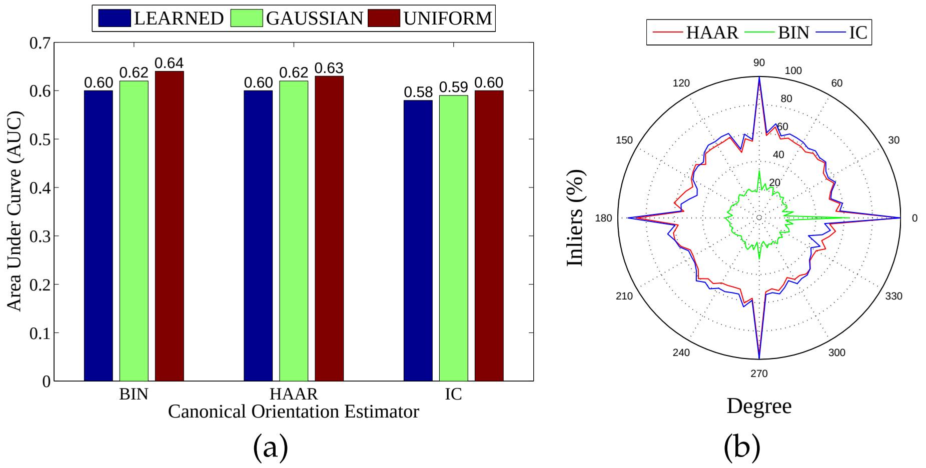


Figure 4.8. (a) The match performance using 9 combinations of 3 distributions and 3 algorithms to estimate the canonical orientation; (b) Invariance to orientation with the 3 algorithms used to estimate the canonical orientation (using the uniform distribution).

D_2 there is a variation only in normal displacement, or ii) D_1 has variation only in normal and D_2 has variation only in intensity. Thus, the probability of ambiguity is $\frac{1}{4} \times \frac{2}{4} = 12.5\%$. Although the probability of the AND operator generating ambiguity is null, the use of the AND operator is too restrictive since it requires a detection of variation in normal and intensity to set a bit. Noise in either image or in depth map can produce different descriptors for the same surface.

Finally, Figure 4.8 shows that the combination which provides the largest AUC result and a more stable invariance to rotation is that using the HAAR algorithm with a uniform distribution.

Analysis of Correlation and Variance In this section we analyze the discriminative power of BRAND and BASE. We also evaluate the variance of each bit in the descriptor vector and examine their correlation.

To evaluate the discriminative power, we computed the bit variance and tested the correlation between each pair of points in the distribution patch. The reason for these experiments is the fact that when bits have high variance, there are different responses to inputs, which leads to more discriminative descriptors. Additionally, a set of uncorrelated pairs is also desirable, since each pair being tested will contribute to the final result.

Figure 4.9 shows the distribution of the averages for a descriptor with 256 bits over 50k keypoints as computed by BRIEF, BASE and BRAND. Note that each bit feature of the BRIEF descriptor has a large variance and a mean close to 0.5. This is

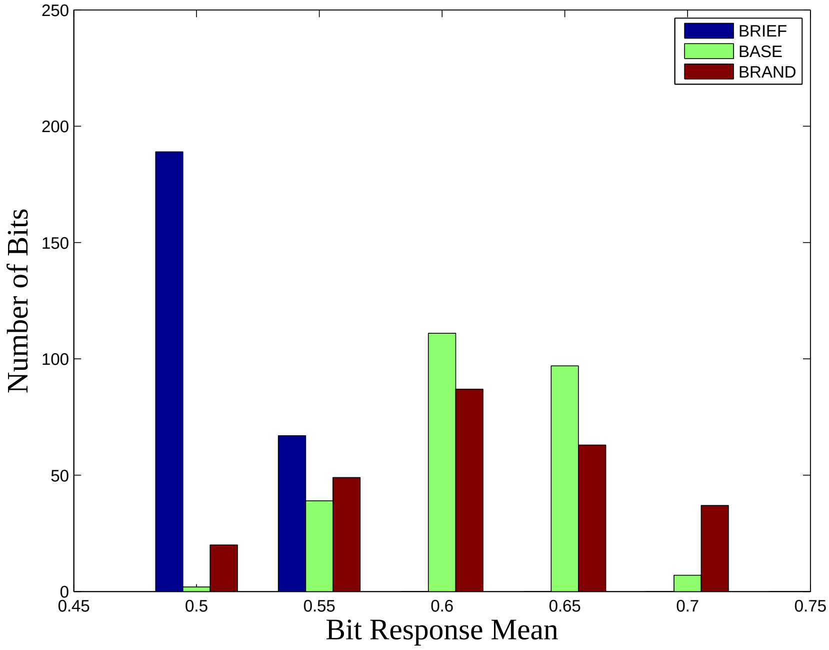


Figure 4.9. Histogram of the descriptor bit mean values for BRIEF, BASE and BRAND over 50k keypoints.

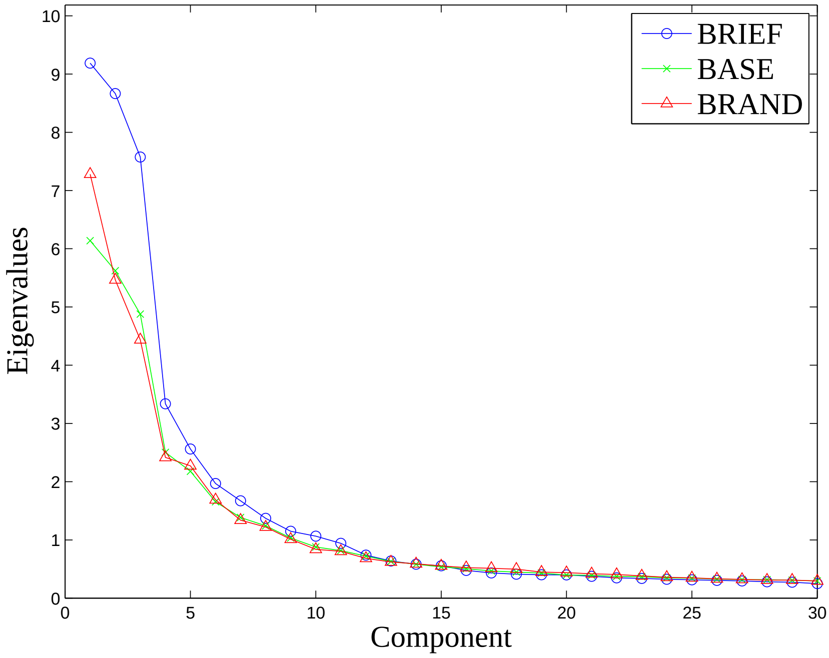
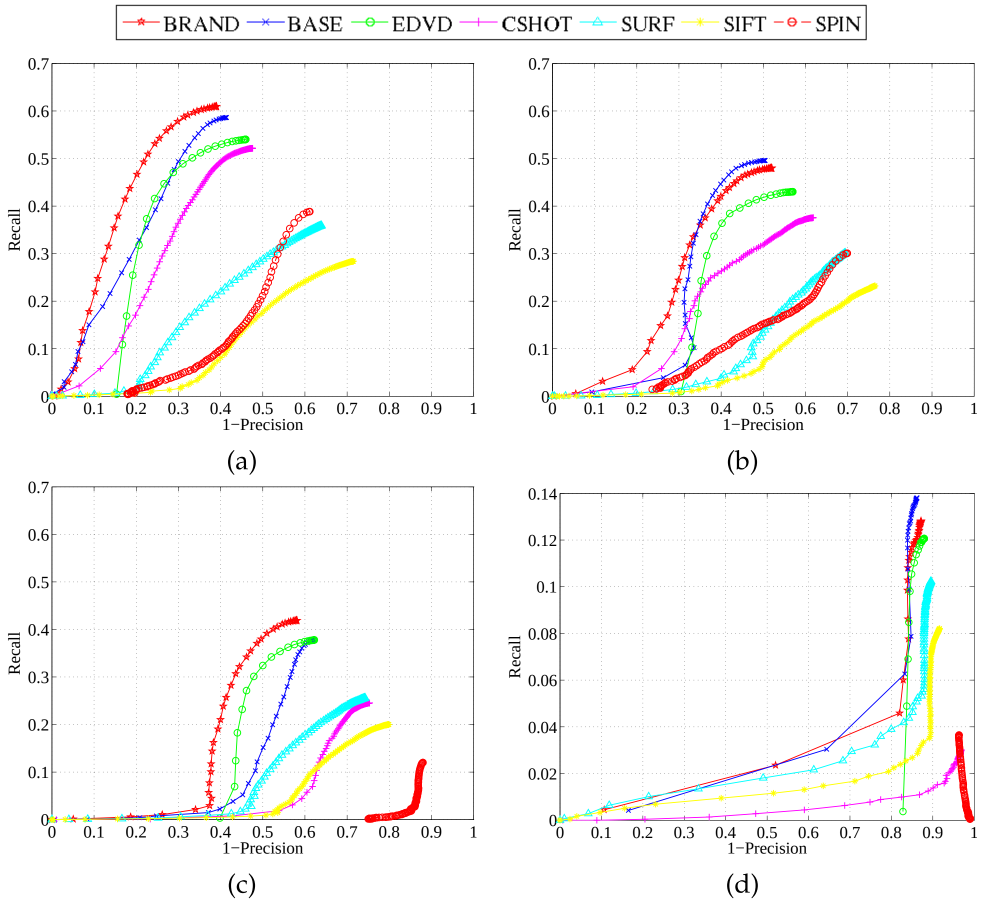


Figure 4.10. PCA decomposition over 50k keypoints of BRIEF, BASE and BRAND.

the best case. Although BASE and BRAND do not show the same spread of means, they do not present a uniform distribution pattern, which is the worst pattern in regard to variance measure.

To estimate the correlation among test pairs, we used PCA on the data and selected the highest 30 eigenvalues. In Figure 4.10 we can see these values. In spite of the fact that BRIEF exhibits larger variance, it also has large initial eigenvalues, which indicates correlation among the pairs. In this test, our descriptors present less correlation among the bits, and BRAND is more discriminative than BASE, given that BASE has smaller eigenvalues.

4.2 Matching Performance Evaluation



(a)

(b)

(c)

(d)

Figure 4.11. Precision-Recall curves for (a) freiburg2_xyz, (b) freiburg2_rpy, (c) freiburg2_desk and (d) freiburg2_pioneer_slam2. The keypoints were detected using the STAR detector. Our descriptors outperform all other approaches, including CSHOT, which like EDVD, BRAND and BASE, combines texture and geometric information. Among our descriptors, BRAND stands out as the best.

Figure 4.11 shows the results of the threshold-based similarity matching tests. As illustrated in the precision-recall curves, the BRAND, BASE and EDVD descrip-

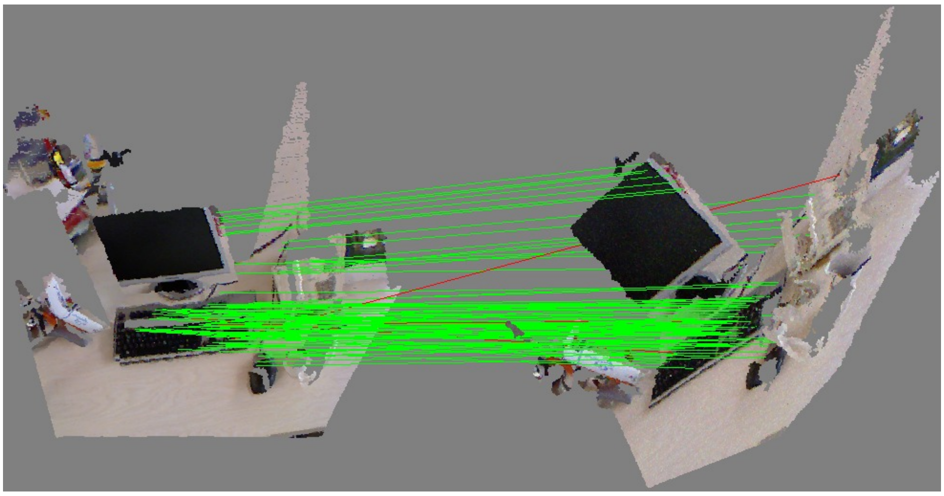


Figure 4.12. Three-dimensional matching example for two scenes using BRAND descriptor. Mismatches are shown with red lines and correct matches with green lines.

tors demonstrated a significantly better performance than other approaches for each sequence. Even for the two more challenging sequences, *freiburg2_desk* and *freiburg2_pioneer_slam2*, which contain high speed camera motion, and in the case of the *freiburg2_pioneer_slam2* sequence, data that was acquired with a robot manually controlled by joystick along a long textureless hall.

Among the descriptors developed in this thesis, on the average, BRAND presented the best results of accuracy followed by the BASE descriptor.

4.3 Rotation Invariance and Robustness to Noise Experiments

Each of the descriptor's invariance to rotation was also evaluated as well as their robustness to noise. For these tests we used synthetic in-plane rotation and added Gaussian noise with several standard deviation values (Figure 4.13). After applying rotation and adding noise, we computed keypoint descriptors using BRAND, EDVD and SURF, followed by brute-force matching to find correspondences.

Figure 4.13 (a) shows the results for the synthetic test when using standard deviation of 15. We can see that both BRAND and EDVD outperform SURF descriptor in rotation invariance. The results are given by the percentage of inliers as a function of the rotation angle.

In Figure 4.13 (b), the results for the synthetic test for noise with standard deviation of $15, 30, 45, 60$ and 75 are shown. Notice that BRAND and EDVD are more

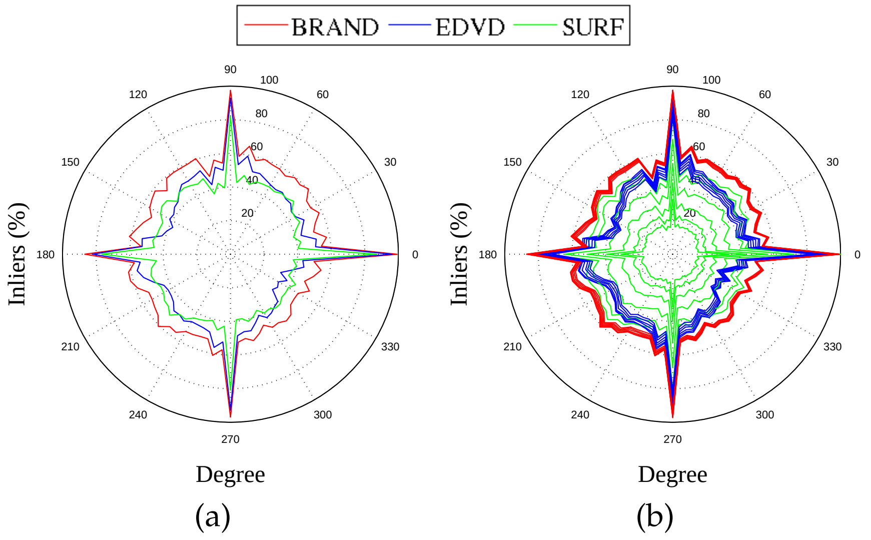


Figure 4.13. Percentage of inliers as a function of rotation angle for BRAND, EDVD and SURF algorithms. (a) Matching performance for synthetic rotations with Gaussian noise with a standard deviation of 15; (b) Matching sensitivity for an additive noise with standard deviations of 0, 15, 30, 45, 60 and 75. The noise was applied in the image and depth domain for BRAND and EDVD experiments.

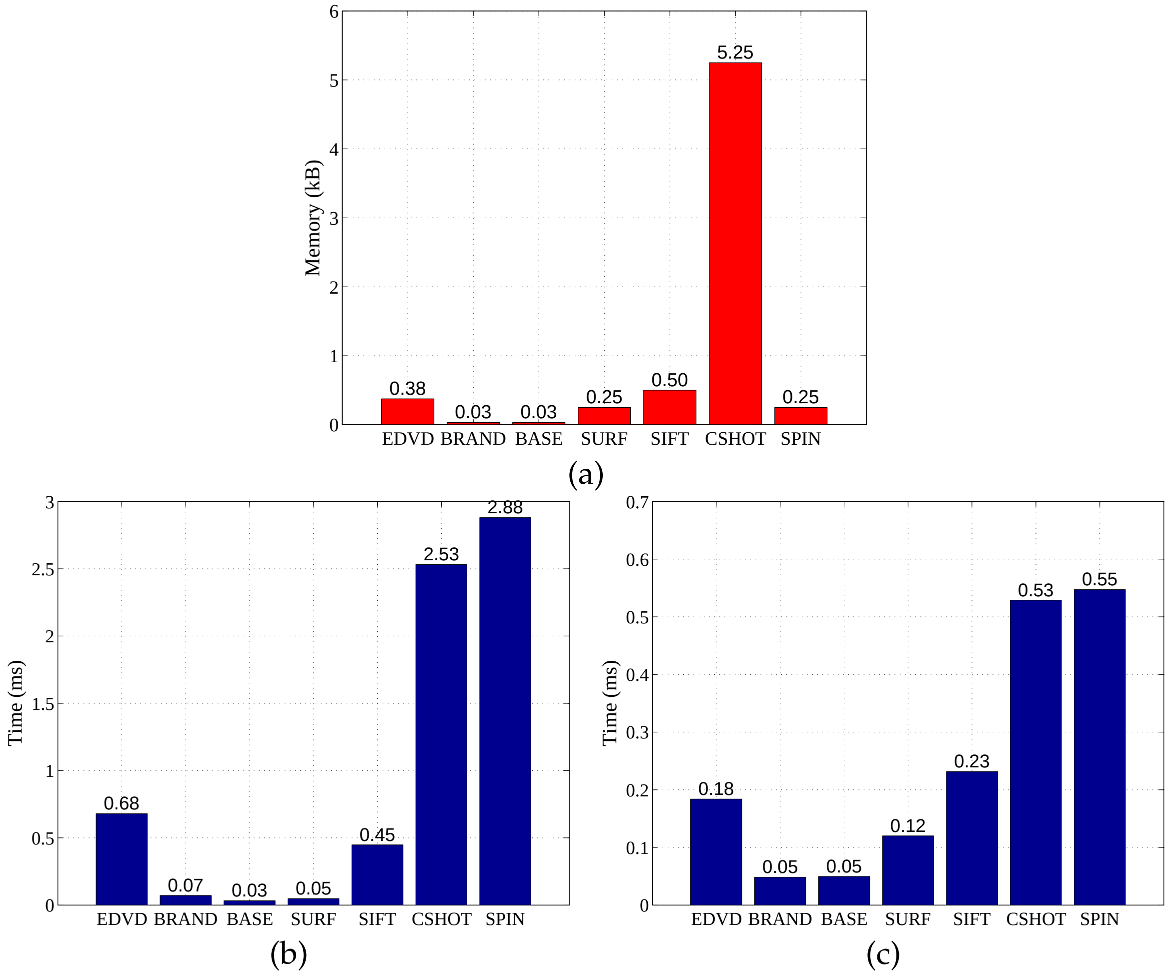
stable and outperform SURF in all scenarios and BRAND and EDVD are largely unaffected by noise. Figure 4.12 shows an example of a three-dimensional matching for two scenes with a rotation transform.

4.4 Processing time and Memory Consumption

Another important property for a descriptor is the processing time to create a signature and to compare two vectors. We performed several experiments to measure these times for our descriptors. Descriptor creation and matching times have been measured and the experiments executed on an Intel Core i5 2.53GHz (using only one core) processor running Ubuntu 11.04 (64 bits). Time measurements were averaged over 300 runs and all keypoints (about 420) were detected by the STAR detector.

Figure 4.14 (b) clearly shows that BRAND is faster than the other descriptors in the matching step, and that it spends slightly more time than SURF for the creation step (Figure 4.14 (a)). This is due to the scale and canonical orientation estimation, a necessary step to rotate and scale the distribution pattern.

Additionally, Figure 4.14 (c) shows that BRAND and BASE present the lowest memory consumption with 32 bytes for the keypoint descriptors, while CSHOT, which also combines appearance and geometry, has descriptors of 5.25 kBytes in



(a)

(b)

(c)

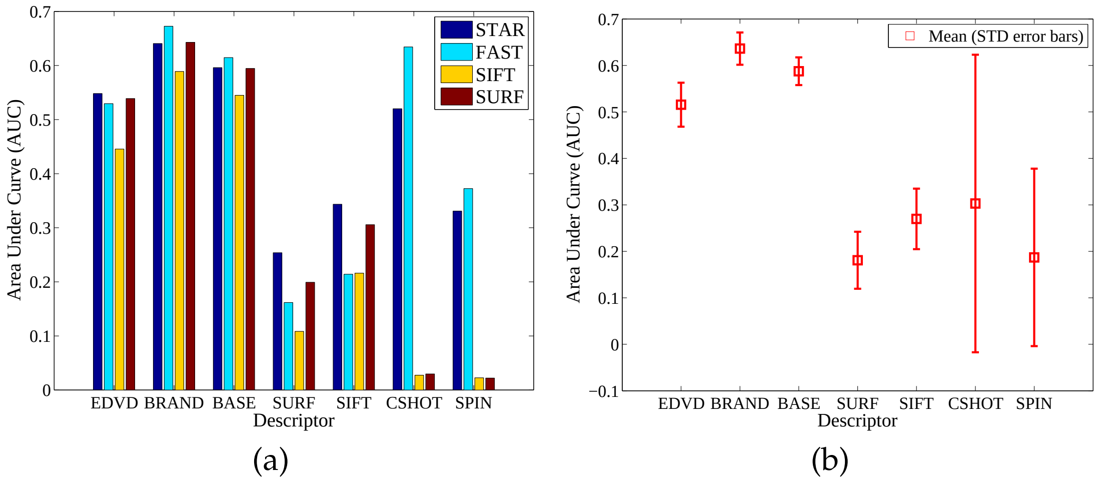
Figure 4.14. Comparison among descriptors using: (a) memory consumption in Kbytes; (b) processing time to compute a single keypoint descriptor and (c) to perform the matching between a pair of points.

size.

4.5 Keypoint Detector versus Accuracy

In this section we evaluate the influence of keypoint detector algorithms in the matching quality. For all descriptors, we match keypoints detected with four differents methologies: STAR [Agrawal et al., 2008], FAST [Rosten et al., 2010], SIFT [Lowe., 2004] and SURF [Bay et al., 2008]. All experiments were executed using the *freiburg2_xyz* sequence.

The independence of a descriptor from keypoint dectector algorithms is highly desirable. With this descriptor independence it is possible to take advantage of the vast number of methodologies that are proposed every year.



(a) (b)

Figure 4.15. Comparison between descriptors using four different keypoint detectors: (a) Respective AUC of the recall vs 1-precision curves for each combination descriptor and detector; (b) The standard variation for each descriptor.

Figure 4.15 (a) shows AUC values for all experiments. We can see, in Figure 4.15 (b), that among all methologies, EDVD, BRAND and BASE stand out as descriptors with the smallest standard variation and highest average in the accuracy. The plots also show that our main competitor, CSHOT, is the least stable method having the highest standard variation equals to 0.32 for an average accuracy of 0.30. BRAND has a standard variation of 0.03 and an average accuracy of 0.64.

4.6 Remarks

This chapter presented several experiments that we performed to show the behaviour of our three descriptors. A comparative analysis in terms of robustness to affine transformations, processing time and memory consumption was conducted against the standard descriptors in the literature for appearance and geometric information. In these experiments, EDVD, BRAND and BASE outperformed the other approaches, including the state-of-the-art CSHOT, which also fuses appearance and geometry information.

Thanks to the strategy of combining different cues, our descriptors were more stable in matching experiments as well as in the invariance to rotation tests. As shown in the experiments, the combination of appearance and geometry information indeed enables better performance than using either information alone. Moreover, our binary descriptors, BRAND and BASE, had superior performance in time and memory consumption and presented high accuracy in matching, achiving the properties of being fast and lightweight. Finally, the three descriptors presented in this thesis showed a small dependence on the keypoint detector.

Chapter 5

Applications

IN THIS CHAPTER WE APPLY OUR DESCRIPTORS to two important tasks in Computer Vision and Robotics: Semantic Mapping and Tridimensional Alignment. In order for robots to achieve higher levels of abstraction, they must be able to build structured representations of their environment by categorizing spatial information. The building of accurate 3D models of a scene, however, is a fundamental problem in Computer Vision.

After demonstrating good performance of our descriptors in the experiments shown in the previous chapter, the following sections will evaluated the behaviour of our descriptors in less controlled data acquisition.

5.1 Semantic Mapping and Object Recognition

The use of categorization in mapping tasks can be used to generate semantic information which would enable robots to distinguish objects, to identify events and to execute high-level tasks. The importance of including semantic information in understanding the environment has been advocated in several works, some examples of which include [Chatila and Laumond, 1985] and [Kuipers and Byun, 1991].

Visual classification tasks are typically tackled with the extraction of image features which are then used to represent individual characteristics of objects and classes. The high dimensionality of data is greatly reduced by using image features, which enables increased performance in the matching process and a reduction in memory usage during the training and the recognition steps. Therefore, feature point descriptors are a part of the underlying structure of a large number of state-of-the-art classification approaches.

As discussed in the previous chapters, the features of these other approaches are estimated from images alone and they rarely use other information such as geometry. Consequently, variation in scene illumination and textureless objects, common issues with real scenes, may dramatically decrease performance of classifiers based solely on the image.

The combination of visual and shape cues is a very promising approach for object recognition but is still in its prelude. As far as efficacy is concerned, however, Lai et al. [2011b] have already shown that the combined use of intensity and depth information outperforms view-based distance learning using only one of the two. The reason that many descriptors have not used shape information can be partially explained by the fact that until recently object geometry was not easy to obtain, nor quick, so as to be combined with image feature data in a timely manner.

5.1.1 Object Recognition

In this section we show the performance of our three descriptor algorithms in an object recognition task. Our experiments were performed using the RGB-D Object Dataset presented by Lai et al. [2011a]. This dataset is availabe from the Computer Science Department from Washington University [1] and contains 51 categories for a total of 300 objects. The images were acquired with a prototype RGB-D camera from Prime-Sense and a firewire camera from Point Grey Research. The images have 640×480 resolution, the color and depth informations were simultaneously recorded. The data was recorded at three differents viewing heights at approximately 30, 45 and 60 degrees above the horizon. Figure 5.1 shows some samples of the objects used in our experiments.

Recognition System To test the discriminative power of our descriptors, we built a recognition system using the Bag of Features (BoF) approach [Csurka et al., 2004] combined with Partial Least Squares (PLS) technique [Rosipal and Krämer, 2006]. The main reason for using the BoF approach was that it is not possible to extract keypoints from the same location in differents samples, the choice of PLS, however, was due to good results in several recognition tasks such as human detection [Schwartz et al., 2009] and face recognition [Schwartz et al., 2010].

Like other recognition systems based on the BoF approach, our system is composed of four main steps:

1. **Feature extraction**: In this step, we split RGB-D images into a grid with 1000 cells and for each cell we compute a descriptor vector;

[1] http://www.cs.washington.edu/rgbd-dataset

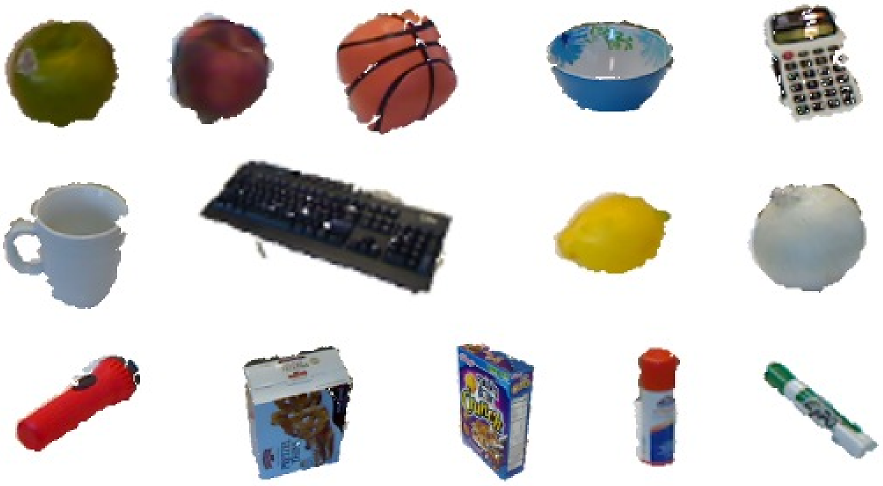


Figure 5.1. Some samples of objects used for recognition experiments. From top left to bottom right: two kinds of apples, a ball, a bowl, a calculator, a coffe mug, a keyboard, a lemon, onion, a flashlight, two kinds of cereal box, a glue stick and a Marker.

2. **Codebook creation**: After running a *k-means* algorithm [Duda et al., 2001] for all descriptors in the dataset, we built a set of K clusters, each represented by a descriptor vector. Finally, we stack all of the K clusters creating a matrix of K rows. The number of columns is defined by the size of the descriptors, e.g. 32 columns of bytes using BRAND and BASE or 512 columns of bytes using SIFT. The number of clusters used in our experiments to build the codebook was $K = 512$ for all descriptors;

3. **Bag of Feature vectors extraction**: For every image in the dataset, we computed a histogram of the number of descriptors assigned to each cluster. These histograms are called bag of features vectors and are used to represent each image in the codebook domain;

4. **Learning**: In this last step, we run the PLS algorithm with a set of bag of features vectors to build the classification model. Since our recognition system uses the one-against-all scheme, we build a model for each class using the remaining samples of other classes as negative samples.

In our experiments we used 20 classes with 30 samples from each class. For every class we randomly selected 5 objects. Our recognition system was trained using four of the five objects and then tested with the fifth, never-before-seen, object. This procedure is repeated three times to obtain the confusion matrices shown in

Figure 5.2. Recognition using spin-image descriptors were not tested since the k-means algorithm was unabled to find the clusters due to the lack of discrimance of descriptors.

Object Recognition Results Figure 5.2 shows the confusion matrices for our three descriptors, SIFT, SURF and CSHOT. As can be seen from the results, our descriptors presented an accuracy similar to others, even though less memory was used and had a faster processing time.

5.1.2 Object Recognition Using the BASE descriptor

In this section, we used the BASE descriptor as presented in Chapter 3 in a simple adaptive boost classification framework to provide semantic information in a mapping task. This framework was used to detect and recognize objects under different illumination conditions. We chose to use the BASE descriptor because it presented the highest accuracy in experiments from the previous section.

Although the recognition system using the BoF approach and the PLS algorithm worked well, a semantic mapping framework needs to be fast and have low memory consumption since the classification algorithm is used in almost every frame grabbed during robot navigation. In this section we present a simpler and faster classification strategy. We then used this second classification strategy to test our descriptor against the first strategy.

Experimental results presented later show that in spite of the simplicity of the recognition approach, high accuracy classification was obtained with processing time on the order of few milliseconds running on current generation processors.

Learning Algorithm Objects are modeled as weighted sets $\mathcal{O}$ of descriptors $\mathbf{f_o}$ computed at keypoints. A careful choice of these keypoints allows not only for good object detection from multiple views, but also decreases the search space making it adequate for online applications.

The weight of each set is computed by a learning process using the *Adaboost* algorithm [Freund and Schapire, 1995]. In order to classify a new RGB-D image, we find the nearest neighbor matching for all sets of object models and a voting mechanism is used to extract a model from the weighted sets.

One of the simplest methods to classify a test set of descriptors $\mathcal{T}$ as belonging to an object $\mathcal{O}$ is to find the nearest neighbors of each descriptor $\mathbf{f_t} \in \mathcal{T}$ that

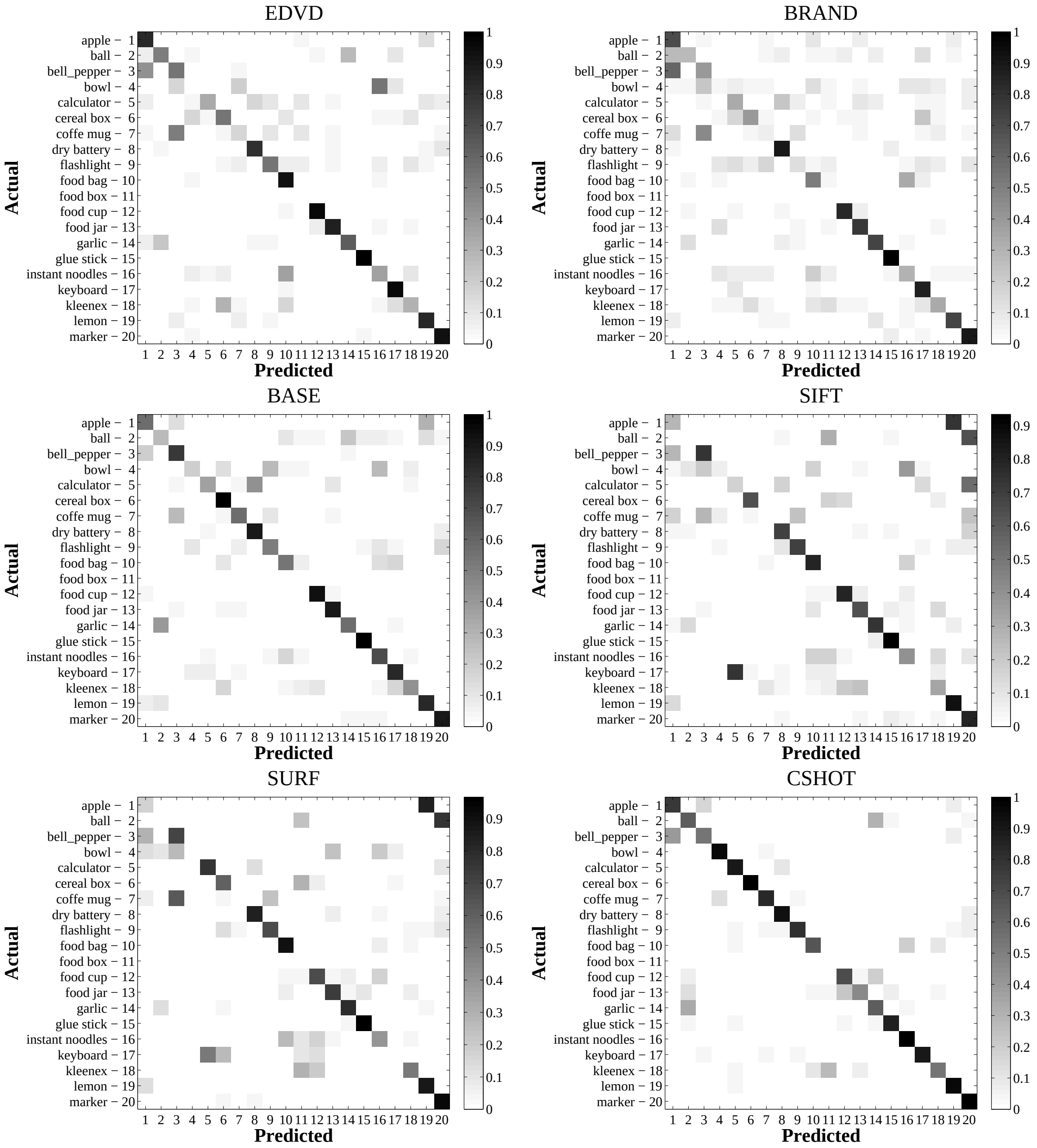


Figure 5.2. Confusion matrices (rows-normalized) among 20 classes from the RGB-D Object Dataset [Lai et al., 2011a].

minimizes the distance function D as given below:

$$D(\mathbf{f_t}, \mathcal{O}) = \min_{\mathbf{f_o} \in \mathcal{O}} \mathrm{D}(\mathbf{f_t}, \mathbf{f_o}). \tag{5.1}$$

Since BASE descriptors are strings of bits, the *Hamming* distance D is used as a distance metric. One of the greatest advantages of this approach, besides its simplicity, is its low computational cost. On the downside of this naïve approach, however, is that it tends to produce several false positives in the final classification. Therefore, to improve classification, we use a multiclass discriminative algorithm which returns the probability of a datum belonging to a given class.

Our classifier is composed of binary weak classifiers h_i, $i \in \{1, \ldots, n\}$ integrated by the *Adaboost* algorithm. Each weak classifier contains a set of descriptors $\mathcal{O}$ which represents an object. The probability that a test set $\mathcal{T}$ corresponds to the object is given by:

$$h(\mathcal{T}) = \frac{1}{|\mathcal{T}|} \sum_{f_t \in \chi} \chi(D(\mathbf{f_t}, \mathcal{O}) \leq \tau), \tag{5.2}$$

where χ is a indicator function that returns 1 if the condition in the argument is true and 0 otherwise. The term $\mathbf{f_t}$ is the descriptor vector of test object $\mathcal{T}$. Threshold τ restricts the minimum distance for a valid match.

The multi-class classifier then selects a classifier H with maximum membership probability. Hence, the class of a test RGB-D image with a set of descriptors $\mathcal{T}$ is given by:

$$c^* = \arg\max_{H \in \mathcal{H}} \sum_{i=1}^{|H|} w_i h_i(\mathcal{T}), \tag{5.3}$$

where $\mathcal{H}$ is the set of trained classifiers, w_i is the weight of the weak classifier h_i and c^* is the class represented by classifier H.

5.1.3 Mapping System

A particle filter was used for robot localization by selecting its most probable position. The classifier returns a class label c^* for each frame acquired from the RGB-D sensor during robot navigation. If the label is different than "none" then it is indexed to the current location. Algorithm 1 describes the mapping process.

5.1.4 Recognition Results

To evaluate the performance of the proposed classification approach, we initially collected several images with a Kinect system mounted on a Pioneer P3-AT mobile ground robot, as shown in Fig. 5.4. The final dataset was composed of 17 samples of 9 objects with different shapes and textures and 30 samples of random images from our lab to represent the negative dataset (Figure 5.3). Two images of the objects in distinct views were used to extract the sets of keypoints to build weak classifiers in

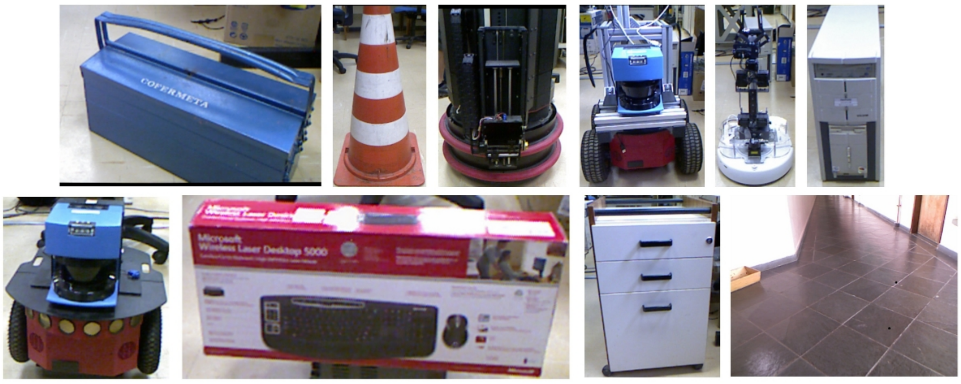


Figure 5.3. Objects used for classification and detection experiments. From top left to bottom right: toolbox, cone, Nomad robot, Pionner Robot Model 2, iCreate Robot, PC, Pioneer Robot Model 1, keyboard box and cabinet. The last image is an example of negative sample used in the training and test steps.

the training step.

We then performed two tests: i) First we trained the classifier and verified the quality of the classification using all of the images in dataset that were not used in the learning stage; ii) second, the objects in the dataset were randomly positioned in the hallways of the laboratory building and a map with the location of each detected object was created.

Finally we compared the performance of our descriptor to that of SURF, a standard 2D descriptor in the literature, and BRIEF, a binary and fast descriptor.

Algorithm 1 Semantic Map($\mathcal{H}$)

```
while true do
    $p \leftarrow \mathrm{ParticlefilterPosition}()$
    $f \leftarrow \mathrm{getRGBDimage}()$
    $\mathcal{K} \leftarrow \mathrm{FAST}(f)$
    $\mathcal{T} \leftarrow \{b(\mathbf{p}) | \mathbf{p} \in \mathcal{K}\}$
    Find label class $c^*$ solving:
        $c^* = \arg\max_{H \in \mathcal{H}} \sum_{i=1}^{|H|} w_i h_i(\mathcal{T})$
    if $c^* \neq$ "none" then
        $\mathrm{map}[p] \leftarrow c^*$
    end if
end while
```

Receiver operating characteristic To evaluate the correct matching rate using our descriptor, we selected one image from the dataset in which the object was directly facing the sensor and computed the set of descriptors. These descriptors were matched against the descriptors of all 16 others images of the objects as well as with the 30 negative images.

Figure 5.5 summarizes the results of true positive and false positive rates as Relative Operating Characteristic (ROC) curves for all objects in the dataset. Better matchings are closer to the upper-left corner. Six out of nine objects had their curves very close to the upper-left corner. Even though the curves of three objects (PC, Cone and Toolbox) were not as close as those of other objects, the true positive rate for them were larger than 80% with a false positive rate lower than 20%.

Learning and Classification Time Keypoint descriptors are at the heart of a large number of vision based machine learning algorithms. In spite of the ever growing performance of computer systems, the unsurmountable volume on visual data now available tends to be processed and used on mobile devices with limited resources. Therefore faster and more efficient keypoint descriptors need to be developed.

In order to estimate the performance of our descriptor, we ran the learning and classification algorithms on our dataset five times and measured CPU time. We

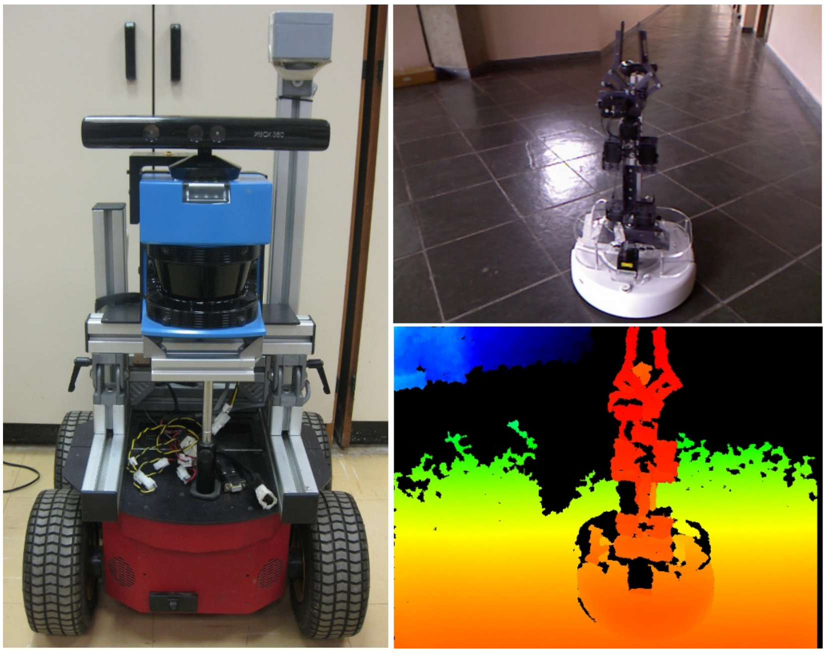


Figure 5.4. Experimental Setup. Left: The mounted Kinect RGB-D camera in a Pionner P3-AT. Right: RGB camera and the depth camera views.

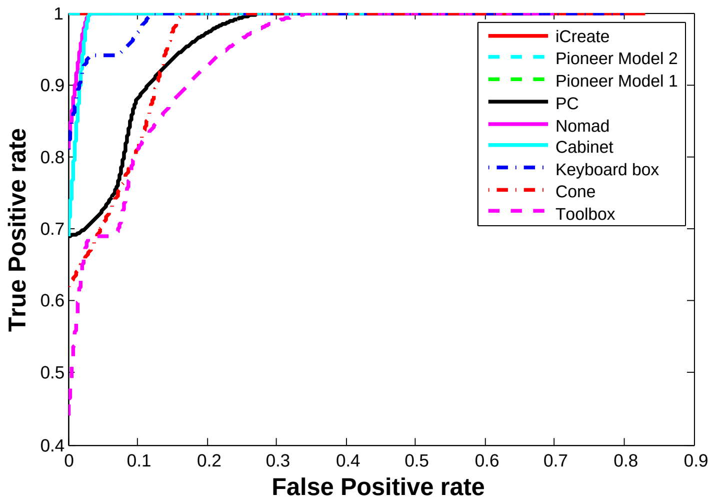


Figure 5.5. ROC curve of matching using BASE descriptor. The best matching is closer to the upper-left corner. We note high true positive rate with low false positive rate for all objects in the dataset.

compared the performance with two intensity only descriptors: BRIEF and SURF on the same dataset. Figure 5.6 shows that in both steps, learning and classification, our descriptor was faster than the others. The learning time of our descriptor was 60% faster than SURF and 15% faster than BRIEF. For the classification step, our descriptor ran 2 times faster than BRIEF and almost 4 times faster than SURF.

One reason why our descriptor runs faster than BRIEF is due to the fact that our descriptor includes more meaningful information to build the classifier. In our experiments we observed that BRIEF uses more than one weak classifier for its binary classifier. This leads to a matching with more than one set of descriptors. This also demonstrates the discrimination superiority of our descriptor over BRIEF.

A far as memory usage is concerned, our descriptor and BRIEF have similar performances since both use binary strings which result in low memory utilization.

Classification Rate As described in Section 5.1.2, we have adopted an ensemble approach using *Adaboost* algorithm. The descriptors used in the experiments were BRIEF, SURF and BASE. We train the classifiers with 9 positive samples of objects in different views and 9 negatives samples. By comparing the confusion matrices among nine classes, in Figure 5.7 we observe that classification using our descriptor obtains results significantly better than the others. Although the values in the confusion matrix of our descriptor, shown in Figure 5.7 (a), are slightly more spread than for BRIEF or SURF, this matrix clearly shows better accuracy by the diagonal

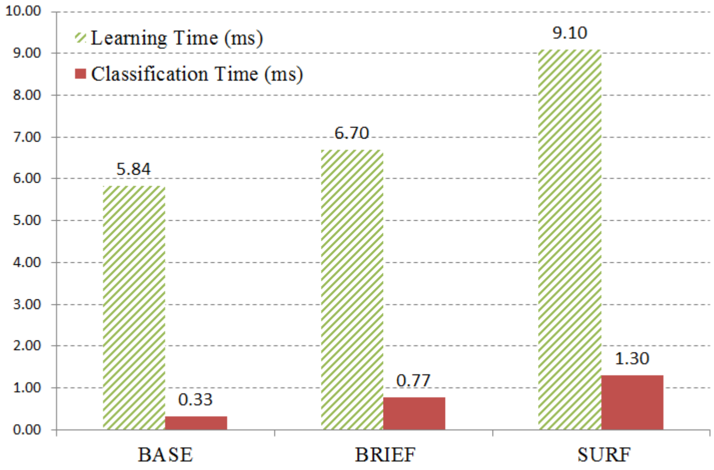


Figure 5.6. CPU time for Learning and Classification steps. In both experiments BASE was faster than BRIEF and SURF. While the use of BASE in learning step is approximately 2 times of SURF and it is closer to BRIEF, in the classification performance our descriptor was almost 2 times faster than BRIEF and 4 times faster than SURF.

that is not found in the confusion matrices of the other descriptors. Also, an analysis of the BRIEF and SURF confusion matrices show that the classifiers built with those descriptors present a strong bias toward a given class (e.g. Toolbox).

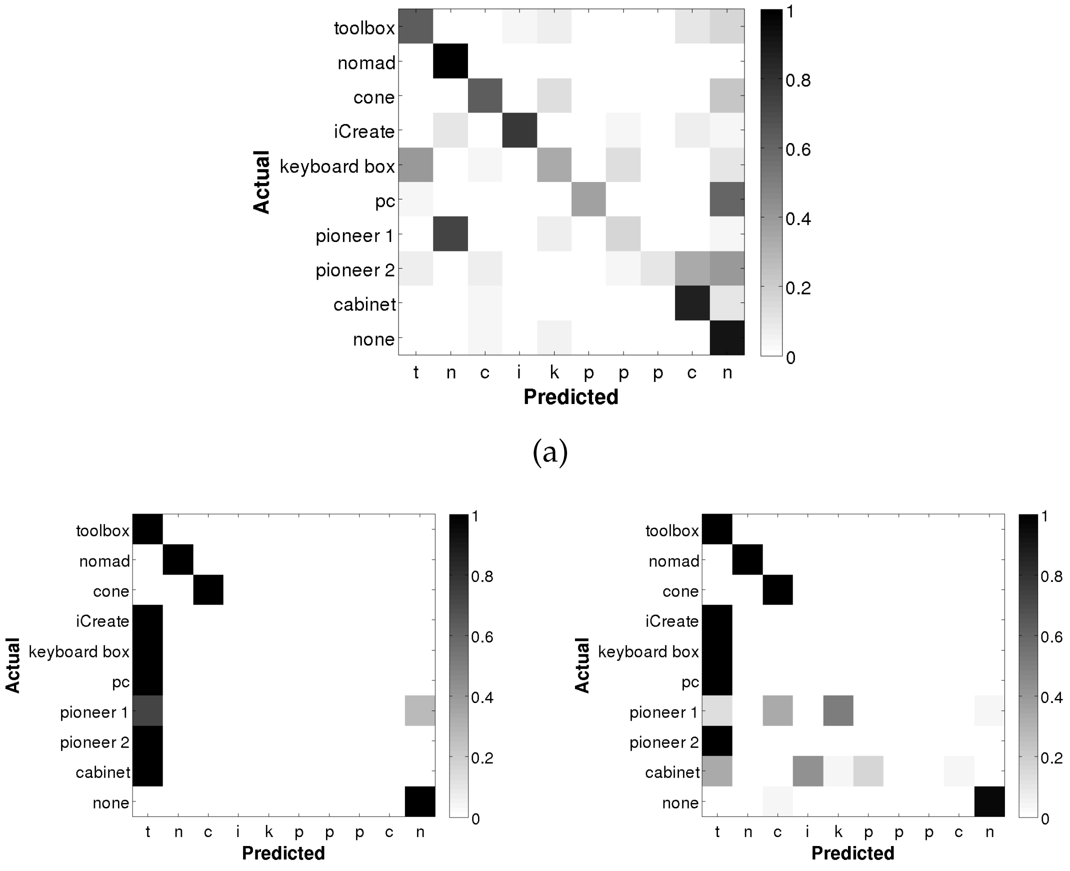
5.1.5 Mapping Results

We tested semantic mapping by spreading objects through the laboratory building (ICEx/UFMG). Navigation was based on the Vector Field Histogram [Borenstein et al., 1991] and for localization a particle filter [Thrun, 2002] was used. These algorithms were implemented on Player 3.0.2 and tested on a computer running Linux on a Intel Core i5 with 6 Gb of RAM.

Figure 5.8 (a) shows the results obtained with our approach. The green circles indicate a correct detection and classification. Red stars represent false positive detection and cyan squares mean correct detection, but wrong classification.

We note a superior number of true positive recognitions and higher rate of classified objects, with a small amount of false positive detections. By comparing these results respectively with BRIEF and SURF, Figure 5.8 makes clear the superiority of our descriptor both in recognition rate and robustness to false positives.

In spite of a large variation in illumination between the time when data was collected for training and the testing itself, our method proved to be the least affected by lighting conditions as a result of taking advantage of geometrical information.



(a)

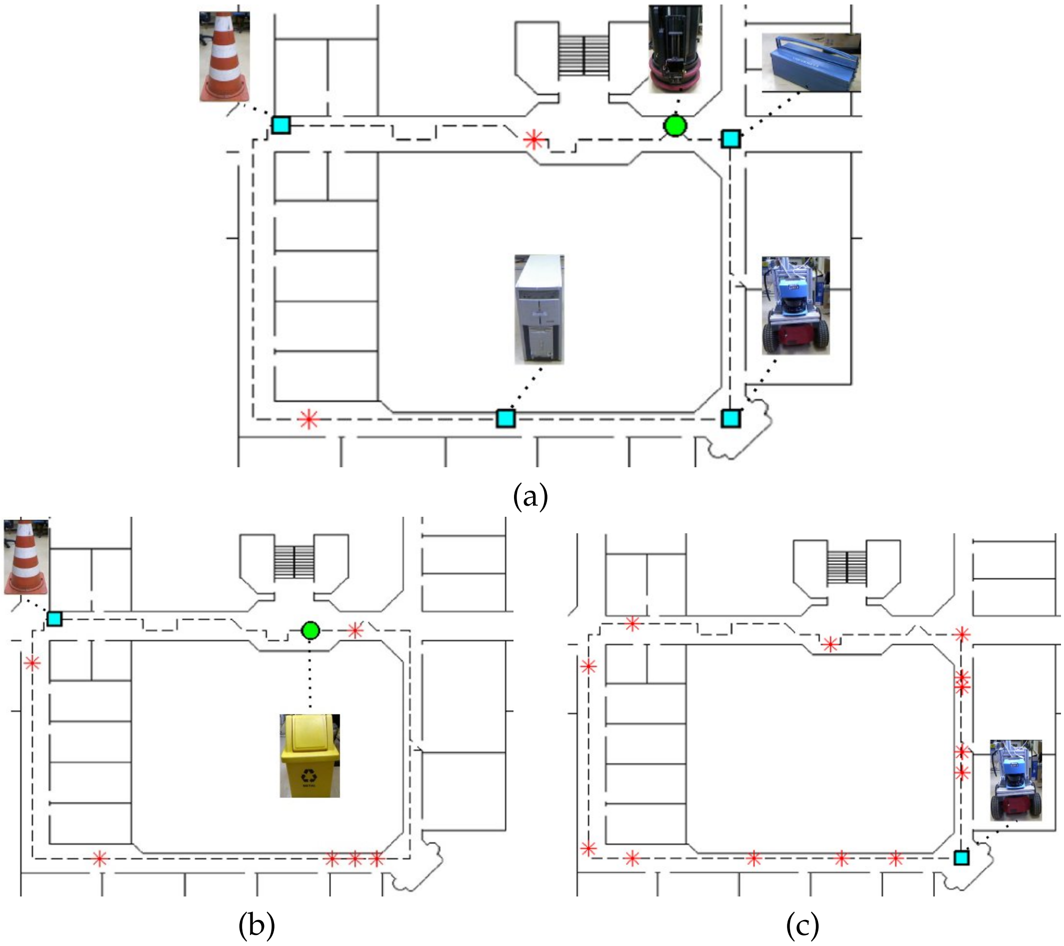
(b)

(c)

Figure 5.7. Confusion matrices (rows-normalized) among nine classes for (a) BASE descriptor, (b) BRIEF and (c) SURF. We observe a much better classification for BASE descriptor justified by the clear diagonal on its confusion matrix even using less time of CPU processing. We also note that the classifiers built with BRIEF and SURF descriptors present a strong bias toward Toolbox class.

5.2 Three-dimensional Alignment

A great challenge in registering multiple depth maps is related to the process of recovering the rigid affine transformation T to describe two depth maps in a single coordinate system. To address this issue, descriptors have been applied to find corresponding points from two depth maps in order to constrain the search space for the transformation T. The work proposed by Vieira et al. [2007] uses a descriptor to propose an iterative framework to address pair-wise alignment of a sequence of depth maps while ensuring global coherence of the registration for implicit reconstruction purposes. A global alignment algorithm that does not use local feature descriptors was presented by Makadia et al. [2006] using Extended Gaussian Images


(a)

(b)

(c)

Figure 5.8. Semantic Map result using (a) BASE, (b) BRIEF and (b) SURF descriptor. The use of BRIEF or SURF result in a large number of false positive detections (red stars). While SURF detected only one object (cyan square) and BRIEF two our descriptor found five without generate too many false positive detections. Green circles indicate correct detection and classification.

(EGI).

Independently of strategies used to pre-align depth maps, a common requirement is that the data have sufficient overlap in order to establish correspondences and a graph defining which pairs, among all depth maps, have such an overlap. Most commercial packages such as DAVID Laserscanner [Winkelbach et al., 2006], require that users manually select the pairs to be aligned. Furthermore, this prealignment is generally refined by local minimization algorithms, such as the classical Iterative Closest Point (ICP) algorithm [Besl and McKay, 1992], in order to achieve the best alignment, given an initial guess of pre-alignment.

Non-rigid and scale invariant registration such as proposed in [Cheng et al., 2010] and [Sehgal et al., 2010] are more often used for matching purposes than reconstruction. A survey on range image registration has been presented in [Salvi et al., 2007], where different methods for pre-alignment and fine registration are compared in terms of robustness and efficiency.

In this section we applied our novel feature descriptors and also described the method employed to perform the registration of multiple indoor textured depth maps.

5.2.1 RGB-D Point Cloud Alignment Approach

The main goal of the registration process is to find an affine transformation T between two point clouds taken from different points fo view.

The approach used to register point clouds in this work is divided into two steps: coarse and fine alignment. In coarse alignment, we compute an initial estimation T of the rigid motion between two clouds of 3D points using correspondences provided by a feature descriptor. Then, in fine alignment, we employ the ICP algorithm to find a local optimum solution based on the previours coarse alignment. The ICP algorithm uses an initial estimate of the alignment and then refine the transformation matrix T^* by minimizing the distances between the closest points. The use of ICP was considered due to its simplicity and low computational time.

The registration process is summarized in Algorithm 2. It has four main steps:

1. **Keypoint Descriptors**: The function ExtractDescriptor receives the source and target point clouds, denoted by $\mathcal{P}_s$ and $\mathcal{P}_t$ respectively, and returns corresponding sets of keypoints with their descriptors, denoted by $\mathcal{K}_s$ and $\mathcal{K}_t$. The first step to compute the set of descriptors for an image or, in our case, an RGB-D point cloud, is to select the subset of keypoints. This selection of keypoints

Algorithm 2 Point Cloud Alignment($\mathcal{P}_s$, $\mathcal{P}_t$)

```
$(\mathcal{K}_s, \mathcal{K}_t) \leftarrow \mathrm{ExtractDescriptor}(\mathcal{P}_\mathrm{s}, \mathcal{P}_\mathrm{t})$
$\mathcal{M} \leftarrow \mathrm{matchDescriptor}(\mathcal{K}_s, \mathcal{K}_t)$
$\mathrm{R} \leftarrow \mathrm{coarseAlignmentSAC}(\mathcal{M})$
repeat
    $\mathcal{A} \leftarrow \mathrm{closestPoints}(\mathcal{P}_s, \mathrm{R}(\mathcal{P}_t))$
    Find T solving:
    $\mathrm{T} \leftarrow \arg\min_{T^*} \frac{1}{|\mathcal{A}|} \sum_{(p_s,p_t)\in\mathcal{A}} |p_s - T^*(p_t)|^2$
    $\mathrm{R} \leftarrow \mathrm{T} \times \mathrm{R}$
until MaxIter Reached or ErrorChange(T) $\leq \theta$
```

with properties such as repeatability provides good detection from multiple views and allows a constrained search space of features making the registration suitable to online applications.

2. **Matching Features**: The function $\mathrm{matchDescriptor}$ matches two sets of descriptors, $\mathcal{K}_s$ and $\mathcal{K}_t$, using a force brute algorithm and return a set $\mathcal{M}$ of correspondence pairs among source and target point clouds. The distance metric used varies with the type of feature descriptor used. The BASE and BRAND descriptors consider the *Hamming* distance metric, EDVD and Spin-Images use correlation function, SIFT and SURF the Euclidian distance and CSHOT the cosine distance.

3. **Coarse Alignment with SAC**: The function $\mathrm{coarseAlignmentSAC}$ is used to provide an initial transformation T using the matching set $\mathcal{M}$. We used a Sampled Consensus-Initial Alignment (SAC) approach [Fischler and Bolles, 1981] to reduce the outliers in correspondences (false correspondences). Our SAC approach works as follow: there is a transformation T from point cloud $\mathcal{K}_s$ to point cloud $\mathcal{K}_t$, this transformation is our model. The algorithm's goal is to estimate the model, i.e. find the matrix T. To achieves this, the algorithm select a random pairs of matchings in $\mathcal{M}$, then it uses this pairs to estimate all the free parameters in the model. All other data are tested against the fitted model and, classified as inliers (if it fits well to the estimated model) or outliers otherwise. If estimated model has sufficiently many points it is classified as consensus set and its parameter are reestimated using only the inliers pairs. At last, the algorithm computes the error of the inliers relative to the model. These steps are performed a fixed number of times and the model with smaller error is selected. The initial transformation T is usually not accurate but constrains to a local search for the optimal transformation using a fine alignment algorithm. We noted, as expected, that less descriptive features provide smaller sets of inliers than more descriptive features.

4. **Fine Alignment**: Finally, the function $\mathrm{closestPoints}$ receives the pre-aligned sets $\mathcal{P}_s$ and $\mathcal{P}_t$ and, constructs the set of pairs $\mathcal{A}$. The set of pre-aligned pairs $\mathcal{A}$ is then used to find a refined transformation in an iterative process. We use a kd-tree for finding the closest point and, in contrast to the work by Henry et al. [2010] which minimizes a non-linear error, we choose an ICP variant that minimizes the error function point-to-point $\sum |p_s - \mathrm{T}(p_t)|^2$. This error function can be solved using the Horn closed-form [Horn, 1987].

5.2.2 Alignment Results

We examined the performance of our descriptor to the registration task for several images of a research laboratory collected with a Kinect sensor (see Figures 5.9 and 5.11). From this we created five challenging sets with different views:

1. **Lab180**: point cloud with holes (regions not seen by the sensor);
2. **Boxes**: scene with three object (boxes) with similar geometry;
3. **Robots**: scene with three robots with the same geometry and texture;
4. **Wall**: scene rich with textureless regions;
5. **DarkLab**: a set of point clouds acquired from a partially illuminated scene.

The experiments were performed on a computer running Linux on an Intel Core i5 with 4 Gb of RAM. For each final alignment we evaluated the alignment error returned by ICP, the number of inliers retained in the coarse alignment and the time spent for fine and coarse alignment. In all experiments, the convergence criteria was a maximum of 100 iterations of ICP or an error less than 0.001. We also used same parameter in SAC algorithm for all descriptors. Tables 5.1, 5.2 and 5.3 show the registration results. We note that the alignment with the EDVD, BASE, BRAND descriptor provides the smallest error despite of its low computational cost. Figure 5.9 shows visual results of the alignment achieved using our descriptors for the five sequences used in the experiments. The screenshots shown in Figure 5.10 present several three-dimensional alignments of our laboratory. These results were provided running our alignment algorithm with BASE in a set of RGB-D images acquired from our laboratory moving a kinect sensor in 360 degrees around its base.

Since the our descriptors consider shape information and the RGB-D camera has its own illumination, we were able to register point clouds even with sparsely illuminated environments. To test the proposed approach, an experiment was performed in a poorly illuminated room. We collected 229 frames of the scene with images ranging from well illuminated to complete lack of light. The final alignment, shown in the Figure 5.11, makes clear that even with some regions without illumination it was possible to align the clouds.

Also we examined the performance of our descriptor with the proposed registration approach in the Freiburg's dataset. We selected three sequences in the Freiburg's dataset: *freiburg2_xyz*, *freiburg2_desk* and *freiburg2_pioneer_slam2*. To evaluate a set of estimated poses we measure the Relative Pose Error (RPE), which is

Table 5.1. This table shows mean values of the ICP error.

	ICP Score				
Descriptor	**Robots**	**Boxes**	**Lab180**	**Wall**	**DarkLab**
EDVD	0.0025	0.0002	0.0047	0.0001	0.0038
BRAND	0.0025	0.0002	0.0041	0.0001	0.0059
BASE	0.0025	0.0002	0.0041	0.0001	0.0043
SURF	0.0035	0.0002	0.0070	0.0004	–
SIFT	0.0058	0.0042	0.0281	0.0021	–
SPIN	0.0046	0.0017	0.0356	0.0205	–
CSHOT	0.0043	0.0002	0.0095	0.0013	0.0033

Table 5.2. This table shows mean values of time spent to register two clouds.

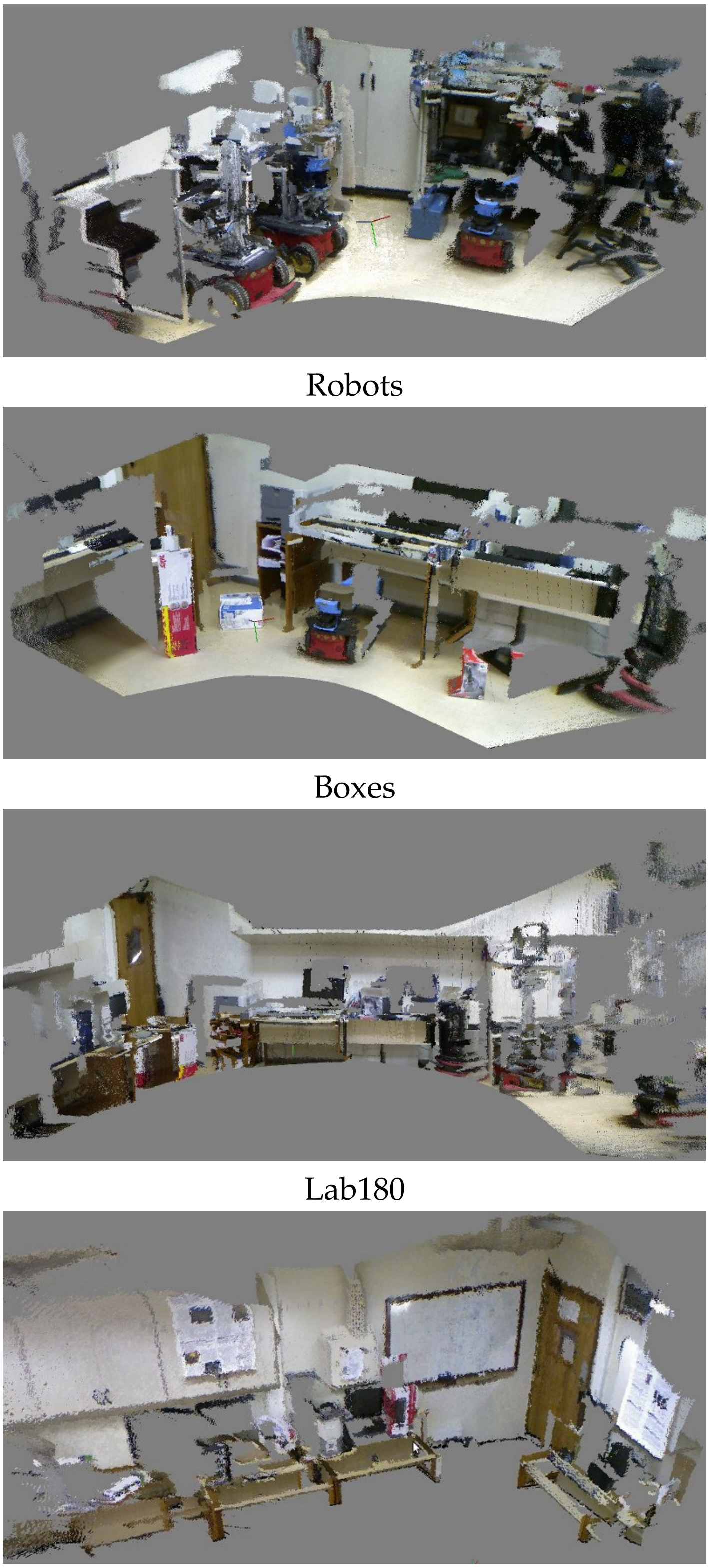
	Alignment Time (seconds)				
Descriptor	**Robots**	**Boxes**	**Lab180**	**Wall**	**DarLab**
EDVD	0.83	0.57	1.01	1.07	2.45
BRAND	0.34	0.27	0.59	0.72	1.09
BASE	0.30	0.27	0.68	0.71	0.81
SURF	0.69	0.31	2.40	0.97	–
SIFT	1.28	1.24	6.29	2.09	–
SPIN	2.56	1.70	8.13	9.18	–
CSHOT	2.29	1.30	2.60	2.40	2.20

Table 5.3. This table shows average number of inliers retained by SAC in the coarse.

	Number of Inliers				
Descriptor	**Robots**	**Boxes**	**Lab180**	**Wall**	**DarkLab**
EDVD	111.85	95.46	52.22	64.87	117.89
BRAND	131.95	105.18	51.75	70.64	64.87
BASE	116.95	108.96	53.00	70.96	63.99
SURF	96.59	58.39	82.09	46.47	–
SIFT	152.10	99.52	129.23	69.66	–
SPIN	155.05	71.30	176.82	181.60	–
CSHOT	143.49	53.54	113.52	66.29	50.28

well-suited for measuring the drift of a visual odometry system. The resuls are shown in Figure 5.12 and 5.13. One may readily see that our descriptors show less error for all sequences, both in translation and rotation. In the most challenge sequence, our descriptors provide an alignment with a translation error less than 1 meters and CSHOT presented an error of about 6 meters.

Additionally, Figures 5.14 and 5.15 show that the most stable descriptor algorithms for differents distances between two frames were BRAND and BASE.



Robots

Boxes

Lab180

Wall

Figure 5.9. Dataset used in the alignment tests. The images show clouds aligned using the BASE descriptor.

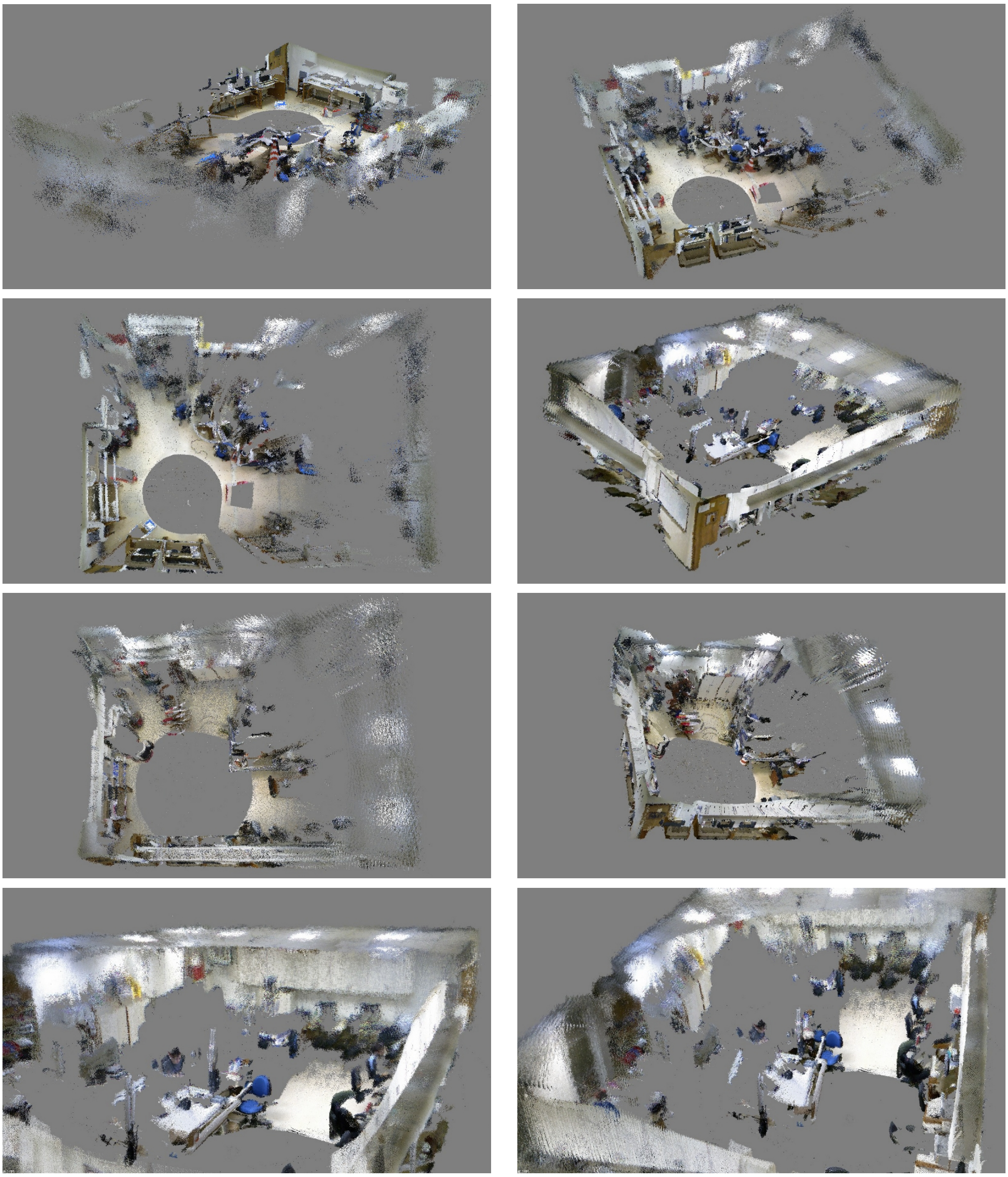


Figure 5.10. Three-dimensional point clouds alignment of VeRLab laboratory.

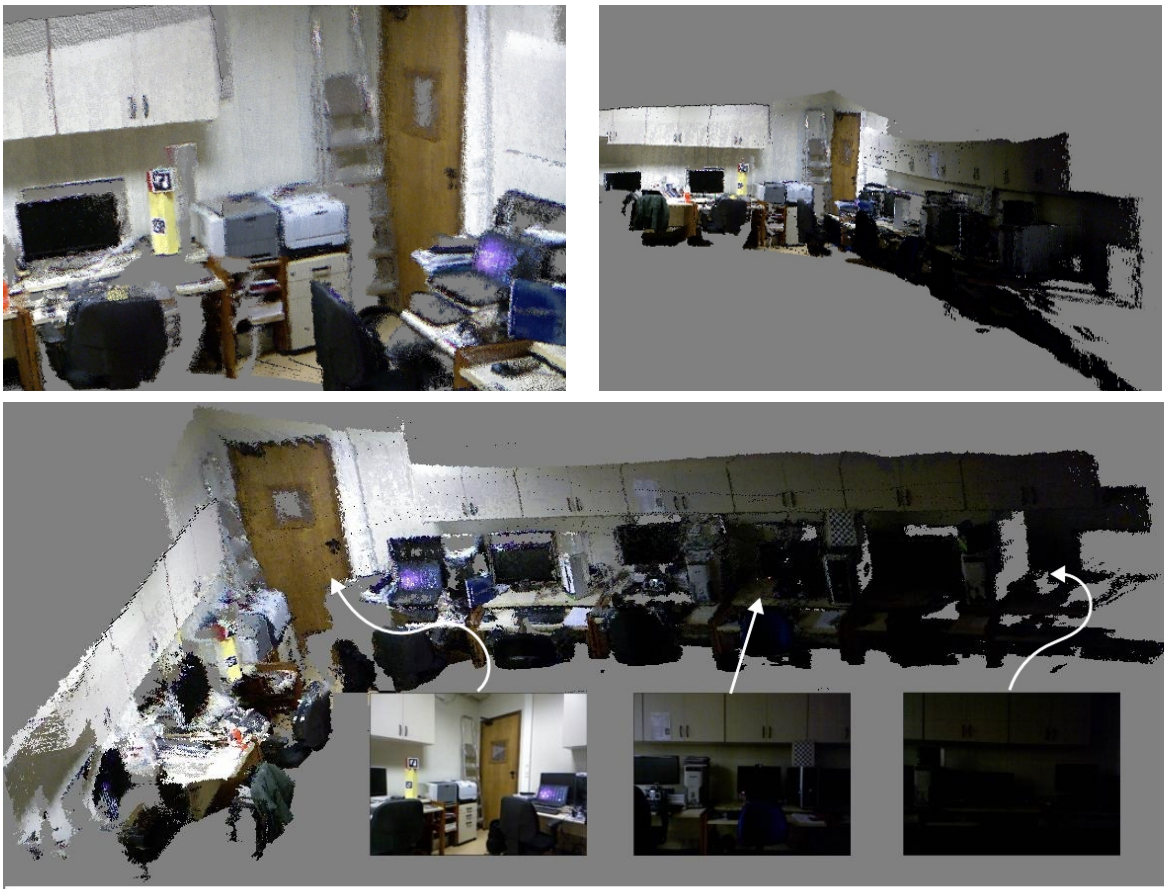


Figure 5.11. Registration of a partially illuminated lab. The frames were used with images from a scene ranging from well illuminated to complete darkness. As BRAND contains geometric information, it is possible to perform the match of the keypoints even if the scene is under inadequate illumination.

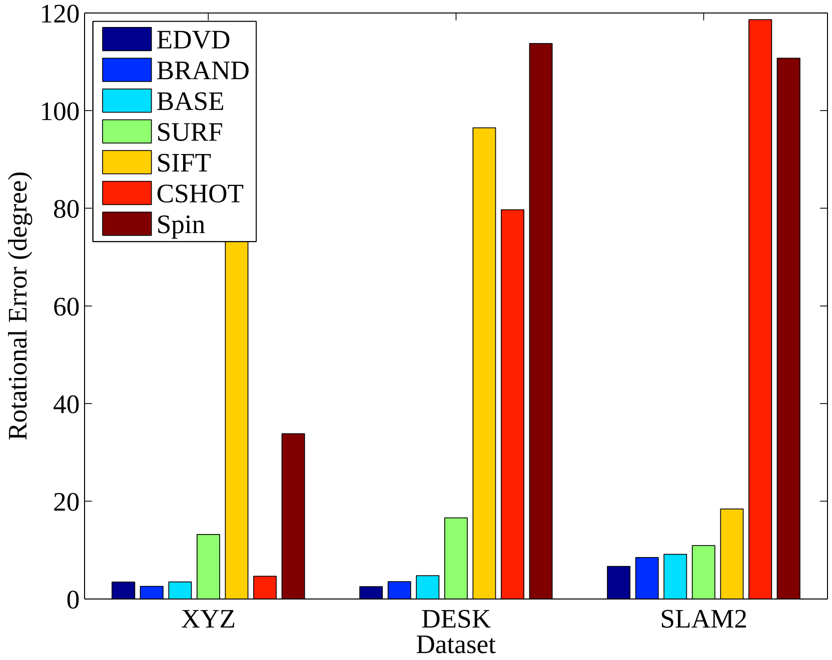


Figure 5.12. Relative Pose Error (RPE) for rotational error in degrees.

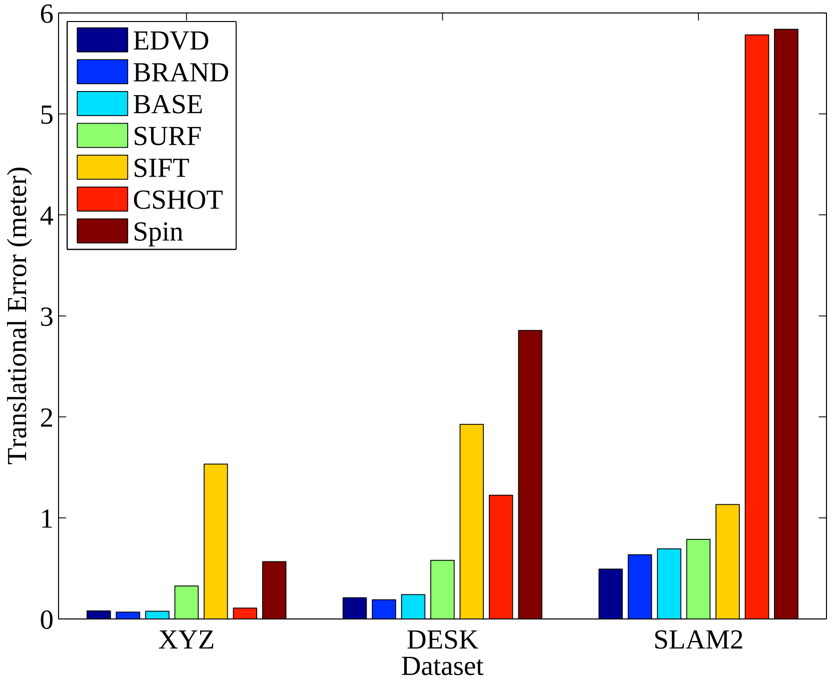


Figure 5.13. Relative Pose Error (RPE) for translational error in meters.

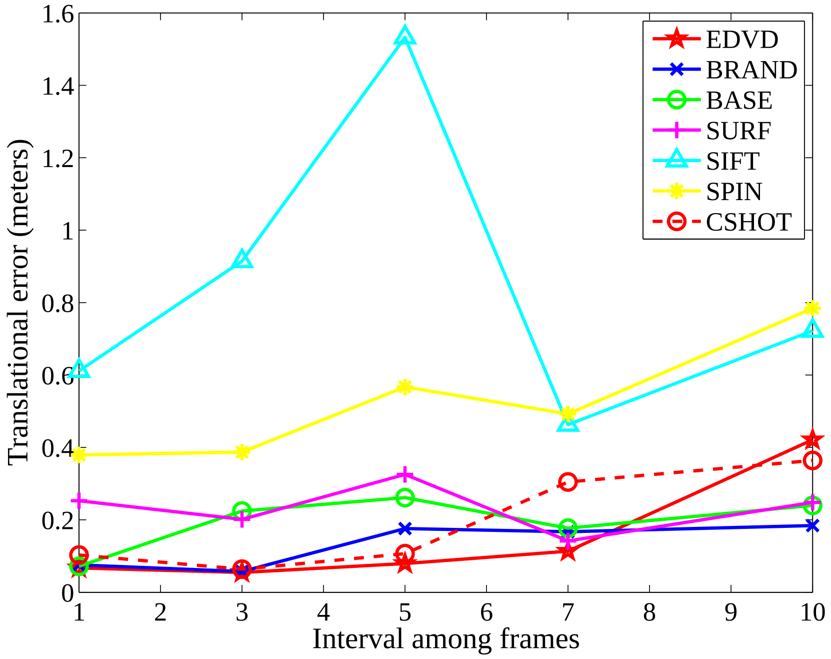


Figure 5.14. Translational error (RPE) for several differents distance between two frames. The error is in meters.

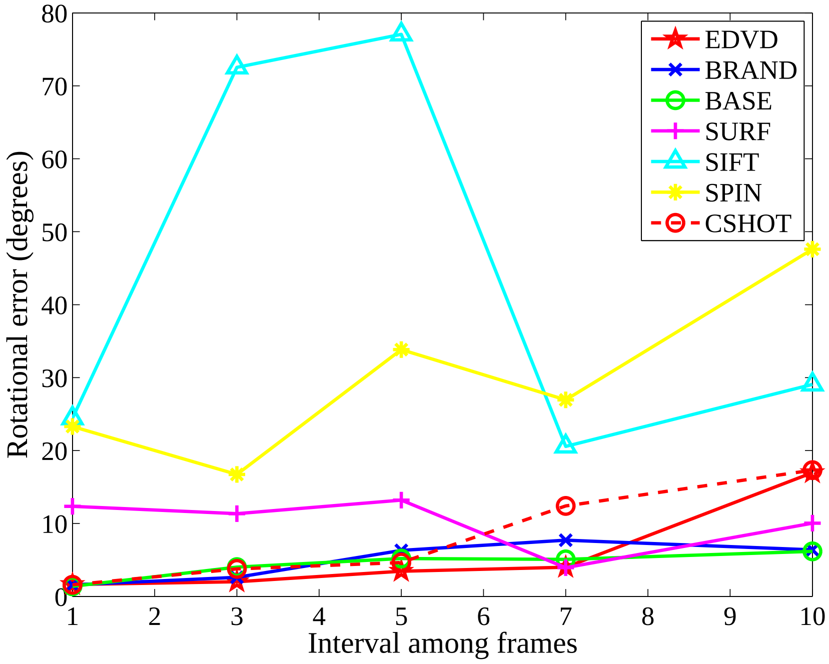


Figure 5.15. Rotational error (RPE) for several differents distance between two frames. The error is in degrees.

Chapter 6

Conclusions and Future Work

In this chapter we present a summary of the accomplished work, emphasizing the main contributions of this thesis. Afterwards, we conclude the chapter by presenting future directions of this work.

6.1 Summary

In this thesis, the problem of how to design an approximation of an ideal descriptor has been addressed. We have proposed a general methodology of constructing robust, scale and rotation invariant descriptors. We designed three novels descriptors using our methodology presented in the Chapter 3 and we believe that this methodology is adequate to be used as a design guide in the creation of new robust and invariant descriptors. Thereby, this work offers three main contributions to the state-of-the-art:

- The robust descriptor EDVD, which, besides providing invariance to orientation, scale and different illumination conditions, presents an algorithm with a low dependence on the normal estimation. Since, even using a coarse normal estimation approach EDVD has the same accuracy when using a precision approach;

- The fast, robust and lightweight descriptor BRAND, which like EDVD has invariance to orientation, scale and illumination conditions, is fast to compute and compare and has low memory usage;

- And the BASE descriptor, a fast and ligthweight descriptor which in spite of the lack of robustness to scale and orientation transforms, presents high accuracy in matching tasks. As with EDVD and BRAND, the BASE descriptor efficiently combines intensity and shape information to improve the discriminative power enhancing the matching process.

From a theorical standpoint, our work exploits these techniques to build robust, quick and low memory consumption descriptors suitable for online applications such as 3D mapping and object recognition applications. These techniques are able to work in modest hardware configurations with limited memory and processor use. For instance, even being invariant to scale and rotation transform, the BRAND descriptor can be stored using just 256 bits of memory.

A comparative analysis was conducted against three standard descriptors in the literature and the state-of-the-art, and we showed that our three descriptors outperform all of these other approaches in terms of robustness to affine transformations estimation, processing time, memory consumption and matching accuracy. Moreover, our descriptors shown a smaller dependence on the keypoint detector.

Additionally, we applied our descriptors in two challenging applications: Semantic mapping and registering multiple indoor textured depth maps. Experiments on registration tasks demonstrated that our technique provides small alignment errors similar to other less efficient descriptors and in some cases proved even better. The experiments also show that our descriptors are robust under poor lighting and sparsely textured scenes as expected.

To evaluate the use of our descriptor for object detection and recognition, we proposed an efficient and simple framework based on *Adaboost* and a more accurate complex framework using the combination of bag of features and partial least squares algorithm. We tested these approaches using two different datasets and our descriptors demonstrated high accuracy in the confusion matrix and faster execution times for both the learning and classification steps.

We also demonstrated the application of the proposed methodology in a classification task for Semantic Mapping. We compared the performance of our descriptors with the results obtained with BRIEF and SURF for the same task, and showed that our approach outperforms the other two descriptors both in detection and in recognition rates.

The results presented here extend the conclusion of Lai et al. [2011b]; Tombari et al. [2011] and Henry et al. [2010] where the combined use of intensity and shape information is advantageous not only in perception tasks, but also in improving the quality of other tasks such as the correspondence and registration process. Combined shape and intensity information indeed renders performances figures that are higher than those attained using either information set alone.

The main constraint of our methodology are the bumpy surfaces. Since the geometrical features are extracted using a threshold for the displacement between normals, the small irregularities of these surfaces can be confused with noise. Another important drawback in our methodology is due to RGB-D camera limitations. While laser scanners have Field of View (FOV) of about 180 degrees, RGB-D sensors have FOV of 60 degrees. And the maximum distance typically less than 5m for RGB-D. Morevover, the currently RGB-D sensors are confined to indoor scenes.

6.2 Future Work

There are several possibilities of research in order to continue the work developed in this thesis. First of all, strong results shown in the experiments and applications chapters have demonstrated the importance of using an appropriate strategy to combine texture and geometrical information. We believe that it is important and necessary to proceed with a theorical investigation about the limits and best ways to perform such combinations.

Another important direction in the near future is to apply the information fusion approach in keypoint detection algorithms. We would like to try a similar strategy to fuse intensity and geometrical features for keypoint detectors. With this, it would be possible to extract keypoints from texturless data as well as from images acquired in scenes with lack of illumination and homogeneous surfaces.

Although our descriptor EDVD presents higher matching accuracy than SIFT, SURF and CSHOT and has invariance to rotation transforms, it suffers with differences in the cell size. We can see in the Figure 3.3 of Chapter 3 that the cells closest to the equator have the largest surface areas, and the bins closest to the north and south poles are the smallest. As future work, we intend to investigate how to overcome this issue, for example, using spherical harmonics directly on the EGI histogram instead of computing the fourier transform in the 2D histogram.

In this thesis, we have worked fusing information in the low level layer, i.e, creating signatures to identify keypoints. A very interesting possibility for continued research involves working with texture and geometrical features on a higher level. For example, we could use the geometrical and image features of our descriptors separately as input data for a learning algorithm such as Lai et al. [2011b].

Finally, as far as mapping is concerned, we will use our methodology to enhance loop closure and modelling of three-dimensional environments. We aspire to developed a dense, real-time SLAM algorithm with our descriptors, where this low consumption technique could be proposed for embedded systems. This work is related to a method of loop closure using RGB-D images using online learning, which aims to amend our registration, realigning point clouds based on the estimation error when a loop occurs.

In summary, as future work, we intend to continue the investigation about the benefits of working with apperance and geometrical information to improve the detection and description of keypoints as well as for use in object recognition and tridimensional alignment.

Bibliography

Agrawal, M., Konolige, K., and Blas, M. R. (2008). CenSurE: Center Surround Extremas for Realtime Feature Detection and Matching. In *Proc. of the Europ. Conf. on Comp. Vision (ECCV)*, pages 102--115.

Ambai, M. and Yoshida, Y. (2011). CARD: Compact And Real-time Descriptors. In *IEEE Int. Conf. on Comp. Vision (ICCV)*.

Andrade, M. and Lewiner, T. (2011). *Cálculo e Estimação de Invariantes Geométricos: Uma Introdução às Geometrias Euclidiana e Afim*. IMPA.

Bay, H., Ess, A., Tuytelaars, T., and Van Gool, L. (2008). Speeded-up robust features (surf). *Computer Vision and Image Understanding*, 110:346--359.

Berkmann, J. and Caelli, T. (1994). Computation of surface geometry and segmentation using covariance techniques. *IEEE Trans. Pattern Anal. Mach. Intell. (PAMI)*, 16(11):1114–1116.

Besl, P. J. and McKay, N. D. (1992). A method for registration of 3-d shapes. *IEEE Trans. Pattern Anal. Mach. Intell. (PAMI)*, 14:239--256.

Borenstein, J., Koren, Y., and Member, S. (1991). The vector field histogram - fast obstacle avoidance for mobile robots. *IEEE Journal of Robotics and Automation*, 7:278--288.

Calonder, M., Lepetit, V., Strecha, C., and Fua, P. (2010). BRIEF: Binary Robust Independent Elementary Features. In *Proc. of the Europ. Conf. on Comp. Vision (ECCV)*.

Chatila, R. and Laumond, J. (1985). Position referencing and consistent world modeling for mobile robots. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 138--145.

Cheng, Z.-Q., Jiang, W., Dang, G., Martin, R. R., Li, J., Li, H., Chen, Y., Wang, Y., Li, B., Xu, K., and Jin, S. (2010). Non-rigid registration in 3d implicit vector space.

In *Proceedings of the 2010 Shape Modeling International Conference*, SMI '10, pages 37--46.

Choi, J., Schwartz, W. R., Guo, H., and Davis, L. S. (2012). A Complementary Local Feature Descriptor for Face Identification. In *IEEE Workshop on Applications of Computer Vision*.

Csurka, G., Dance, C. R., Fan, L., Willamowski, J., and Bray, C. (2004). Visual categorization with bags of keypoints. In *In Workshop on Statistical Learning in Computer Vision, ECCV*, pages 1--22.

Duda, R., Hart, P., and Stork, D. (2001). *Pattern classification*. Pattern Classification and Scene Analysis: Pattern Classification.

Fischler, M. A. and Bolles, R. C. (1981). Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Communications of the ACM*, 24(6):381–395.

Freund, Y. and Schapire, R. E. (1995). A decision-theoretic generalization of online learning and an application to boosting. In *Proc. of the 2nd European Conf. on Computational Learning Theory*, pages 23--37.

Harris, C. and Stephens, M. (1988). A Combined Corner and Edge Detection. In *Proceedings of The Fourth Alvey Vision Conference*, pages 147--151.

Henry, P., Krainin, M., Herbst, E., Ren, X., and Fox, D. (2010). Rgb-d mapping: Using depth cameras for dense 3d modeling of indoor environments. In *International Symposium on Experimental Robotics (ISER)*.

Hetzel, G., Leibe, B., Levi, P., and Schiele, B. (2001). 3D Object Recognition from Range Images using Local Feature Histograms. In *IEEE Conf. on Comp. Vision and Pattern Recog. (CVPR)*, page 394–399.

Holz, D., Holzer, S., Rusu, R. B., and Behnke, S. (2011). Real-Time Plane Segmentation using RGB-D Cameras. In *RoboCup Symposium*.

Horn, B. K. P. (1984). Extended gaussian images. *Proceedings of the IEEE*, 72(2):1671--1686.

Horn, B. K. P. (1987). Closed-form solution of absolute orientation using unit quaternions. *Journal of the Optical Society of America A*, 4(4):629--642.

Hua, G., Brown, M., and Winder, S. (2007). Discriminant Embedding for Local Image Descriptors. In *IEEE Int. Conf. on Comp. Vision (ICCV)*, volume 0, pages 1–8.

Intel (2007). SS4 Programming Reference. http://software.intel.com/file/18187.

Johnson, A. E. and Hebert, M. (1999). Using Spin Images for Efficient Object Recognition in Cluttered 3D Scenes. *IEEE Trans. Pattern Anal. Mach. Intell. (PAMI)*, 21(5):433--449.

Kanezaki, A., Marton, Z.-C., Pangercic, D., Harada, T., Kuniyoshi, Y., and Beetz, M. (2011). Voxelized Shape and Color Histograms for RGB-D. In *IROS Workshop on Active Semantic Perception*.

Ke, Y. and Sukthankar, R. (2004). PCA-SIFT: A More distinctive Representation for Local Image Descriptors. In *IEEE Conf. on Comp. Vision and Pattern Recog. (CVPR)*.

Kembhavi, A., Harwood, D., and Davis, L. (2011). Vehicle Detection Using Partial Least Squares. *IEEE Trans. Pattern Anal. Mach. Intell. (PAMI)*, (6):1250 –1265.

Klasing, K., Althoff, D., Wollherr, D., and Buss, M. (2009). Comparison of surface normal estimation methods for range sensing applications. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 3206 –3211.

Kuipers, B. and Byun, Y. (1991). A robot exploration and mapping strategy based on semantic hierarchy of spatial representation. *Journal of Robotics and Autonomous Systems*, 1(8):47--63.

Lai, K., Bo, L., Ren, X., and Fox, D. (2011a). A large-scale hierarchical multi-view rgb-d object dataset. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*.

Lai, K., Bo, L., Ren, X., and Fox, D. (2011b). Sparse distance learning for object recognition combining rgb and depth information. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*.

Leutenegger, S., Chli, M., and Siegwart, R. (2011). BRISK: Binary Robust Invariant Scalable Keypoints. In *IEEE Int. Conf. on Comp. Vision (ICCV)*.

Lindeberg, T. (1994). *Scale-Space Theory in Computer Vision*. Kluwer Academic Publishers.

Lowe., D. G. (2004). Distinctive image features from scale-invariant keypoints. *Internationl Journal of Computer Vision*, pages 91--110.

Makadia, A., Iv, E. P., and Daniilidis, K. (2006). Fully automatic registration of 3d point clouds. In *IEEE Conf. on Comp. Vision and Pattern Recog. (CVPR)*, pages 1297--1304.

Microsoft (2011). Microsoft kinect.

Mikolajczyk, K. and Schmid, C. (2005). A performance evaluation of local descriptors. *IEEE Trans. Pattern Anal. Mach. Intell. (PAMI)*, 27(10):1615--1630.

Ojala, T., Pietikäinen, M., and Harwood, D. (1996). A comparative study of texture measures with classification based on featured distributions. *Pattern Recognition*, 29(1):51 – 59.

Pang, Y., Li, X., and Yuan, Y. (2010). Robust Tensor Analysis With L1-Norm. *Circuits and Systems for Video Technology, IEEE Transactions on*, 20(2):172 –178.

Pang, Y. and Yuan, Y. (2010). Outlier-resisting graph embedding. *Neurocomputing*, 73(4âĂŞ6):968 – 974.

Rosin, P. (1999). Measuring Corner Properties. *Computer Vision and Image Understanding*, 73(2):291--307.

Rosipal, R. and Krämer, N. (2006). Overview and recent advances in partial least squares. In *Subspace, Latent Structure and Feature Selection Techniques, Lecture Notes in Computer Science*, pages 34--51. Springer.

Rosten, E., Porter, R., and Drummond, T. (2010). FASTER and better: A machine learning approach to corner detection. *IEEE Trans. Pattern Anal. Mach. Intell. (PAMI)*, 32:105--119.

Rublee, E., Rabaud, V., Konolige, K., and Bradski, G. (2011). ORB: an efficient alternative to SIFT or SURF. In *IEEE Int. Conf. on Comp. Vision (ICCV)*.

Rusu, R., Blodow, N., and Beetz, M. (2009). Fast Point Feature Histograms (FPFH) for 3D registration. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*.

Rusu, R., Marton, Z., Blodow, N., and Beetz, M. (2008a). Aligning Point Cloud Views using Persistent Feature Histograms. In *IEEE Intl. Proc. on Intelligent Robots and Systems (IROS)*, pages 22–26.

Rusu, R., Marton, Z., Blodow, N., and Beetz, M. (2008b). Learning Informative Point Classes for the Acquisition of Object Model Maps. In *Proceedings of the 10th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, pages 17–20.

Rusu, R., Marton, Z., Blodow, N., Dolha, M., and Beetz, M. (2008c). Towards 3D Point Cloud Based Object Maps for Household Environments. *Robotics and Autonomous Systems Journal (Special Issue on Semantic Knowledge)*.

Rusu, R. B., Marton, Z. C., Blodow, N., and Beetz, M. (2008d). Persistent Point Feature Histograms for 3D Point Clouds. In *Proceedings of the 10th International Conference on Intelligent Autonomous Systems (IAS-10)*.

Salvi, J., Matabosch, C., Fofi, D., and Forest, J. (2007). A review of recent range image registration methods with accuracy evaluation. *Image and Vision Computing*, 25(5):578 – 596.

Schmid, C. and Mohr, R. (1997). Local grayvalue invariants for image retrieval. *IEEE Trans. Pattern Anal. Mach. Intell. (PAMI)*, 19:530--535.

Schwartz, W. R., Guo, H., and Davis, L. S. (2010). A Robust and Scalable Approach to Face Identification. In *Proc. of the Europ. Conf. on Comp. Vision (ECCV)*, volume 6316 of *Lecture Notes in Computer Science*, pages 476–489.

Schwartz, W. R., Kembhavi, A., Harwood, D., and Davis, L. S. (2009). Human Detection Using Partial Least Squares Analysis. In *IEEE Int. Conf. on Comp. Vision (ICCV)*, pages 24–31.

Sehgal, A., Cernea, D., and Makaveeva, M. (2010). Real-time scale invariant 3d range point cloud registration. In *International Conference on Image Analysis and Recognition (ICIAR)*, pages I: 220–229.

Steder, B., Grisetti, G., and Burgard, W. (2010). Robust Place Recognition for 3D Range Data based on Point Features. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*.

Steder, B., Rusu, R. B., Konolige, K., and Burgard, W. (2011). Point feature extraction on 3d range scans taking into account object boundaries. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*.

Sturm, J., Magnenat, S., Engelhard, N., Pomerleau, F., Colas, F., Burgard, W., Cremers, D., and Siegwart, R. (2011). Towards a benchmark for rgb-d slam evaluation. In *Proc. of the RGB-D Workshop on Advanced Reasoning with Depth Cameras at Robotics: Science and Systems Conf. (RSS)*.

Thrun, S. (2002). Particle filters in robotics. In *Proceedings of the 17th Annual Conference on Uncertainty in AI (UAI)*.

Tombari, F., Salti, S., and Di Stefano, L. (2010). Unique signatures of histograms for local surface description. In *Proc. of the Europ. Conf. on Comp. Vision (ECCV)*, ECCV'10, pages 356--369.

Tombari, F., Salti, S., and Stefano, L. D. (2011). A combined texture-shape descriptor for enhanced 3D feature matching. In *IEEE Intl. Conf. on Image Processing (ICIP)*.

Tuytelaars, T. and Mikolajczyk, K. (2008). Local invariant feature detectors: a survey. *Found. Trends. Comput. Graph. Vis.*, 3(3):177--280.

Vieira, T., Peixoto, A., Velho, L., and Lewiner, T. (2007). An iterative framework for registration with reconstruction. In *Vision, Modeling, and Visualization 2007*, pages 101--108.

Winkelbach, S., Molkenstruck, S., and Wahl, F. M. (2006). Low-Cost Laser Range Scanner and Fast Surface Registration Approach. In *DAGM Symposium for Pattern Recognition*, pages 718--728.

Zaharescu, A., Boyer, E., Varanasi, K., and Horaud, R. P. (2009). Surface Feature Detection and Description with Applications to Mesh Matching. In *IEEE Conf. on Comp. Vision and Pattern Recog. (CVPR)*.

Zhang, Z., Deriche, R., Faugeras, O. D., and Luong, Q. T. (1995). A Robust Technique for Matching two Uncalibrated Images Through the Recovery of the Unknown Epipolar Geometry. *Artificial Intelligence*, 78(1-2):87--119.