Interest Point Detectors and Descriptors for IR Images

AN EVALUATION OF COMMON DETECTORS AND DESCRIPTORS ON IR IMAGES

JOHAN JOHANSSON
Interest Point Detectors and Descriptors for IR Images

An Evaluation of Common Detectors and Descriptors on IR Images

JOHAN JOHANSSON

Master’s Thesis at CSC
Supervisors: Martin Solli, FLIR Systems
Atsuto Maki
Examiner: Stefan Carlsson
Abstract

Interest point detectors and descriptors are the basis of many applications within computer vision. In the selection of which methods to use in an application, it is of great interest to know their performance against possible changes to the appearance of the content in an image. Many studies have been completed in the field on visual images while the performance on infrared images is not as charted.

This degree project, conducted at FLIR Systems, provides a performance evaluation of detectors and descriptors on infrared images. Three evaluations steps are performed. The first evaluates the performance of detectors; the second descriptors; and the third combinations of detectors and descriptors.

We find that best performance is obtained by Hessian-Affine with LIOP and the binary combination of ORB detector and BRISK descriptor to be a good alternative with comparable results but with increased computational efficiency by two orders of magnitude.
Referat

Detektorer och deskriptorer för extrempunkter i IR-bilder

Detektorer och deskriptorer är grundpelare till många applikationer inom datorseende. Vid valet av metod till en specifik tillämpning är det av stort intresse att veta hur de presterar mot möjliga förändringar i hur innehållet i en bild framträder. Grundlig forskning är utförd på visuella bilder medan det fortfarande saknas en lika grundläggande kartläggning av deras prestation på infraröda bilder.

Det här examensarbetet utvärderar, på uppdrag av FLIR Systems, hur detektorer och deskriptorer presterar i infraröda bilder. Arbetet är uppdelat i tre utvärderingar varav den första utvärderar detektorer, den andra deskriptorer och den tredje kombinationen av detektor och deskriptor.

Vi finner att bäst resultat uppnås av Hessian-Affine tillsammans med LIOP men att den binära kombinationen av ORB detektor och BRISK deskriptor är ett bra alternativ som har jämförbart resultat men en ökad effektivitet av två storlekordningar.
I would like to start by thanking my supervisor at FLIR Systems, Martin Solli, for his great support during this thesis.

Thank you Atsuto Maki, my supervisor at KTH, for all your help and dedication. Finally I would like to thank my family and friends for their love and support during my time at KTH.
Contents

1 Introduction ................................. 1
   1.1 Thesis Objective .......................... 1
   1.2 Demarcations .............................. 2
   1.3 Contribution .............................. 2
   1.4 Thesis Outline ............................. 2

2 Background ................................. 3
   2.1 Thermal Radiation .......................... 3
      2.1.1 Principles of Thermal Radiation ........... 4
      2.1.2 Emissivity ................................ 5
      2.1.3 Atmospheric Attenuation .................... 5
   2.2 Interest Point Detectors and Descriptors .......... 5
      2.2.1 Local Features ............................ 5
      2.2.2 Detectors ................................. 6
      2.2.3 Descriptors ............................... 7
   2.3 Related Work ................................ 9
   2.4 Discussion .................................. 10

3 Evaluated Detectors and Descriptors ................. 13
   3.1 Priority of Detectors and Descriptors ............... 13
   3.2 Introduction to Evaluated Detectors and Descriptors .... 14
      3.2.1 Corner Based Detectors .................... 14
      3.2.2 Blob Based Detectors ...................... 14
      3.2.3 Blob Based Detectors and Gradient Based Descriptors ... 15
      3.2.4 Intensity Order Based Descriptor ............ 16
      3.2.5 Binary Descriptors ....................... 16
      3.2.6 Corner Based Detectors and Binary Descriptors .... 16

4 Evaluation Setup ................................ 19
   4.1 Database ..................................... 19
      4.1.1 Deformation Specification .................. 19
      4.1.2 IR Cameras ................................ 20
   4.2 Evaluation Framework ......................... 20
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.2.1</td>
<td>Ground Truth</td>
<td>22</td>
</tr>
<tr>
<td>4.2.2</td>
<td>Matching</td>
<td>23</td>
</tr>
<tr>
<td>4.2.3</td>
<td>Region Normalization</td>
<td>23</td>
</tr>
<tr>
<td>4.2.4</td>
<td>Performance Measures</td>
<td>24</td>
</tr>
<tr>
<td>4.3</td>
<td>Implementation Details</td>
<td>25</td>
</tr>
<tr>
<td>4.3.1</td>
<td>Parameter Selection</td>
<td>26</td>
</tr>
<tr>
<td>5</td>
<td>Evaluation Results</td>
<td>27</td>
</tr>
<tr>
<td>5.1</td>
<td>Detectors</td>
<td>27</td>
</tr>
<tr>
<td>5.1.1</td>
<td>Summary</td>
<td>35</td>
</tr>
<tr>
<td>5.2</td>
<td>Descriptors</td>
<td>40</td>
</tr>
<tr>
<td>5.2.1</td>
<td>Precision-Recall Curve</td>
<td>40</td>
</tr>
<tr>
<td>5.2.2</td>
<td>Summary</td>
<td>46</td>
</tr>
<tr>
<td>5.3</td>
<td>Combinations</td>
<td>48</td>
</tr>
<tr>
<td>5.3.1</td>
<td>Summary</td>
<td>54</td>
</tr>
<tr>
<td>6</td>
<td>Application Analogies</td>
<td>61</td>
</tr>
<tr>
<td>7</td>
<td>Comparisons to Results in Earlier Work</td>
<td>63</td>
</tr>
<tr>
<td>8</td>
<td>Conclusions</td>
<td>65</td>
</tr>
<tr>
<td></td>
<td>Bibliography</td>
<td>67</td>
</tr>
<tr>
<td></td>
<td>Appendices</td>
<td>70</td>
</tr>
<tr>
<td>A</td>
<td>List of Considered Detectors and Descriptors</td>
<td>71</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

Thermography, also known as infrared (IR) imaging, is an imaging method for measuring temperatures in a non-contact approach. Since thermography measures infrared radiation it is a detection technique which makes it possible to visualize radiation not observable by human eye [1].

Infrared imaging is a fast growing field both in research and industry with a wide area of applications. At power stations it is used to monitor the high voltage systems. Construction workers use it to check for defective insulation in houses. Military uses it for surveillance in various contexts, e.g. mounted on vehicles or towers for border surveillance. Fire-fighters use it as a tool when searching for missing people in buildings on fire and electricians may use it to find overheated wiring or components.

In the field of image analysis, especially within computer vision, the major parts of the research have focused on regular visual images. However, research on other types of images, in this case IR images, are less explored. Within image analysis many tasks comprise the usage of an interest point (IP) or feature detector in combination with a feature descriptor. These are, for example, used in subsequent processing to achieve panorama stitching, content based indexing, tracking etc. Development of new IP detectors and corresponding evaluations have, for many years, been an active research area both in the academic world and for commercial purposes.

In this thesis, common feature detectors in combination with descriptors are evaluated on FLIR IR images. The goal of the thesis is to show what combination of detectors and descriptors should be used for applications in image stitching and image retrieval for IR images.

1.1 Thesis Objective

The objective of the thesis is to evaluate the performance of state-of-the-art detectors and descriptors for visual images on IR images. In the evaluation the performance of detectors and descriptors are tested against image condition changes
such as scale, rotation etc. The aim is to, from the evaluation results, conclude and recommend which methods should be used in applications as image stitching and image retrieval on IR images.

1.2 Demarcations

- Evaluated detectors and descriptors will be of floating point and binary types.
- Image condition changes to be tested against in the evaluation are viewpoint, rotation, scale, blur, noise and downsampling.

1.3 Contribution

This work contributes to the field with:

- An evaluation of the performance of detectors and descriptors on IR images.
- A brief analysis of the evaluation results with connection to the applications image stitching and image retrieval.

1.4 Thesis Outline

The outline for the remainder of the thesis is as follows. Chapter 2 introduces some background to thermal imaging and image analysis. Chapter 3 describes how detectors and descriptors were prioritized and gives a brief introduction to selected methods. Chapter 4 introduces the evaluation framework and creation of the database. Chapter 5 presents the results of the evaluation. Chapter 6 briefly analyses the results with connection to applications. Chapter 7 discusses the result compared to earlier results in both visual images and IR images. In Chapter 8 the conclusion of the evaluation is presented. Last, a list of considered detectors and descriptors to include in the evaluation can be found in Appendix A.
Chapter 2

Background

This chapter introduces some background to thermal imaging and image analysis followed by related work within the field.

2.1 Thermal Radiation

The radiation measured in thermography is called thermal radiation, or more commonly infrared light. Infrared light is described in physics as electromagnetic waves and has its spectrum between 780 nm to 1 mm \[1\] compared to visible light which covers wavelengths between 380 to 780 nm.

In IR imaging only a fraction of the IR spectrum is used, about 0.8 to 14 µm which is visualized in Figure 2.1. The infrared light in the imaging range is divided into near-infrared, mid-infrared and far-infrared light.

- Near-infrared light, also called short wave (SW) infrared, usually covers wavelengths between 0.9 to 1.7 µm, but can also be classified to 0.8 to 2.5 µm.
- Mid-infrared light, also called mid wave (MW) infrared, has wavelengths between 3 to 5 µm.
- Far-infrared light, long wave (LW), has wavelengths between about 7 to 14 µm \[1\].

The gaps that appear between SW and MW as well as between MW and LW are bands in which the atmosphere attenuates the infrared light, further discussed in Section 2.1.3. This atmospheric attenuation is also what limits the IR imaging range between 0.8-14 µm.

The different IR-bands are typically used in different applications. SW has applications like visual image enhancement \[2\], eye tracking and Kinect-like cameras. The MW band is typically used in military surveillance cameras \[3\] while the LW band is used in cameras for monitoring, construction work or by fire-fighters to locate missing people in houses on fire.
CHAPTER 2. BACKGROUND

Figure 2.1: Spectrum of the distribution of light. The gap between 5 to 7.5 μm illustrates the attenuation by the atmosphere, replicated from [1].

Used in this work are images from a FLIR T640 camera which detects infrared radiation in the LWIR band, 7.5 to 14 μm, a FLIR P660 and a FLIR SC660 both operating between 7.5 and 13 μm.

2.1.1 Principles of Thermal Radiation

All objects with a temperature above 0 K (−273.15°C) emit thermal radiation [1]. The quantity of thermal radiation is dependent on the emittance, reflectance, transmittance and absorbency of an object and its environment. When measuring the temperature of an object it is the emitted radiation from the object which is of interest.

According to Kirchhoff’s law of thermal radiation an object’s property of emission $\epsilon$ and absorption $\alpha$ may vary with the wavelength [3]. A demonstrative example of this is glass which for visual light and SW infrared light is transparent while for MW and LW infrared light is nearly opaque. Objects having this ability are called selective emitters. In the example of the glass, the transitivity in the visual spectrum has changed to decreased transitivity and increased absorptivity and/or reflectivity in the infrared spectrum due to the increase in wavelength. Therefore to measure the temperature through glass the SW band needs to be used as it becomes opaque for larger wavelengths.

In Kirchhoff’s law it is also stated that the total absorbed radiation is equal to the total emitted radiation for a specific wavelength and temperature, giving $\epsilon = \alpha$ [1]. For regular objects the relationship between absorbed, reflected and transmitted energy is

$$1 = \alpha + \rho + \tau$$  \hspace{1cm} (2.1)

where $\alpha$ is the absorptivity, $\rho$ the reflectivity and $\tau$ the transitivity [3]. The relationship makes it possible to estimate the emissivity through $\epsilon = 1 - \rho - \tau$. 

4
2.2. INTEREST POINT DETECTORS AND DESCRIPTORS

2.1.2 Emissivity

Emissivity is defined as the fraction between the object’s ability to emit radiation and the ability of a blackbody at the same temperature. The higher the emissivity of an object the more radiation will be emitted. An object’s ability to emit radiation is affected by the type of material, surface structure, viewing angle, wavelength and temperature. Worth to notice is that the reflectance for polished surfaces, like metal, increases for increases in wavelength when at the same time the emissivity decreases. For these kinds of materials very small values in emissivity are obtained in the IR band and temperature measurements become more difficult.

2.1.3 Atmospheric Attenuation

Another important aspect, not related to the measured object, is the influence of the atmosphere. Due to absorption of radiation by particles in the atmosphere different bands in the IR spectrum are attenuated. The large gap between 5 to 7.5 µm is due to absorption by water molecules and hence works as a bandstop filter for these wavelengths. Therefore when selecting operating wavelengths the attenuation caused by the atmosphere needs to be taken into account as well as the emitted radiation by the atmosphere.

2.2 Interest Point Detectors and Descriptors

2.2.1 Local Features

In the literature the word *interest point* is one of a few for referring to points with specific characteristics in its neighbourhood in an image, so called image features. When the neighbourhood of an interest point is used to describe the image feature the word *region* or *interest region* is typically used. Two other widely used terms are *local features* and *keypoints*, where the latter will show up in the name of some methods included in the evaluation. In this work IP or *feature* will be used to designate interest points. Common features in computer vision are edges, corners and blobs.

- **Edges** are features with the property of having a large gradient in one direction. In images edges are visualized as rapid changes in color or intensity, see Figure 2.2a.
- **Corners** are similar to edges but with the difference of having large gradients in all directions as in the case of two intersecting lines, see Figure 2.2b. Since corners have a stricter definition than edges they are usually resulting in fewer but more reliable features.
- **Blobs** are regions containing points with similar properties compared to the nearby environment of the blob. In Figure 2.2c a blob will be detected in the circular area of darker pixels and is restricted by the brighter neighbourhood.
Local features are the basis for many applications in image analysis and computer vision. They are used in image retrieval, image stitching, content based indexing and object tracking to mention a few application areas. In these applications it is crucial that the features are of good quality to obtain the desired result. A good feature is in [4] defined to have the properties:

- **Repeatability** - For two images of the same scene or object, the same features should be found in the areas which visualize the same content.
- **Distinctiveness** - The information content in the neighbourhood of a detected feature should be of high variation such that it can be distinguished from other features in a matching process.
- **Locality** - To decrease the probability of occlusion, features should be local. Locality is a competitive property to distinctiveness as an increase in locality decreases the size of the neighbourhood and hence the possibility of high variation.
- **Quantity** - The number of features extracted from the image should be large enough to include necessary information in the image. However, it is undesired to have too many points since it increases the computation time.
- **Accuracy** - It should be possible to decide the localization of the feature accurately, as well as its shape and size.
- **Efficiency** - The computational time needed to extract the features should be low enough for the intended application.

In evaluations discussed in Section 2.3 the most common properties tested as performance measures are repeatability, accuracy and efficiency.

### 2.2.2 Detectors

Detector is the most frequently used term for the tool which extracts features in an image.

Throughout the years numerous detectors have been invented and naturally some of them have become more popular than others due to their robustness and invariance to different changes in the image. The most common detectors and about
2.2. INTEREST POINT DETECTORS AND DESCRIPTORS

Figure 2.3: Advances of feature detection since 2005, replicated from [5].

when they got invented can be observed in Figure 2.3 [5]. In the figure they are divided in different segments depending on if they are edge, corner or blob based.

This work evaluates detectors which are divided into the two categories, corner based detectors and blob based detectors. Edge based detectors are not evaluated since they extract similar kind of features as corner detectors, with the difference of corners being more stable IPs than edges.

In the application areas of detectors it is mainly of interest to find matching features between different images. A big problem in extracting local features is that their occurrence changes when the environment changes, e.g. illumination or distance to an object, and hence aggravates the procedure of finding the same features. In some situations it is difficult, or sometimes even impossible, to take photographs of the same scene having the same view-point, rotation of the camera or the same zoom. What if the sun suddenly gets covered by the clouds? The photos now have different illumination than the ones taken just seconds ago. Due to these possible changes it is desired for the combination of detector and descriptor to be invariant or robust to these kind of changes.

When performing tests on detectors the most common image condition changes to test against are: view-point, rotation, illumination, blur, scale and JPEG compression [6,7]. In some performance tests the detectors are also tested against noise.

Depending on the application these variations have effects on the result in a way or another. For an IP detector to be suitable it should find the same features in an image independent of these changes.

2.2.3 Descriptors

When a local feature has been detected a descriptor is used to depict the information content within the neighbourhood of the feature. The reason why this is of interest
is to match different points and regions which are obtained in the detection step. Obtained matches can then be used in object tracking, object recognition or finding the transformation to stitch two images. Depending on the application, descriptors, just as detectors, need to be robust or invariant to image condition changes. If a detector has extracted the same points, under different image conditions, the descriptor has to be able to store the information in a consistent manner such that a match will occur in a matching step. Usually hundreds or even thousands of interest points are detected which aggravate the procedure of e.g. image stitching if the points can not be distinguished.

Descriptors are mainly divided into two groups: 1) floating point descriptors and 2) binary descriptors. Example of floating point descriptors are SIFT, SURF and LIOP while ORB and BRISK are examples of binary descriptors, which will be further discussed in Section 3.2.

Combination of Detector and Descriptor

All applications make use of the combination of a detector and a descriptor. When deciding what detector and descriptor to use it must be kept in mind that the combination of the two has an impact on the performance and not only the performance of them separately. The choice of combination also depends on the trade-off between accuracy and efficiency needed for the specific implementation. As discussed previously the robustness and invariances of the algorithms also have a great impact.

Floating Point Detectors and Descriptors

The majority of the research on detectors and descriptors has been completed within floating point detectors and descriptors, also known as real valued methods. All methods explained in [4] belong to this category and the two most popular are SIFT [8] and SURF [9].

Floating point descriptors included in this work are either based on gradients or intensity orders.

Binary Detectors and Descriptors

In early days, binary point algorithms were an important part of the research when the computational power in computers was limited. Lately the focus has been brought back to research on detectors and descriptors using binary point algorithms to be used in real time applications or on devices with less computational power e.g. mobile phones.

This family of descriptors stores the information as binary vectors. The benefits of binary methods are high efficiency and low memory requirements at a low cost on the performance. In [10] it is concluded that the presented method ORB is an order of magnitude faster than SURF, and two orders of magnitude faster than SIFT. This is strengthened in [7] where it can be studied that the detector and
2.3. RELATED WORK

descriptor of the binary methods ORB, BRISK and BRIEF are outperforming the floating point methods in speed and memory size.

**Learning Based Methods**

With the growing fields of machine learning and artificial intelligence, research on learning based descriptors has recently started. In [11] the descriptor is based on convolutional neutral networks and uses large databases of images to create large amount of patches used in a training sequence. The received result in the paper shows that learning based methods in descriptors are promising, hence it is likely that much more research will be performed in the field.

Other descriptors which exploit the use of learning based methods are ORB and FREAK which use learned pairs of sampling points when generating the descriptor.

2.3 Related Work

Throughout the years several detectors and descriptors have been invented and tested in corresponding evaluations.

In 2005 Mikolajczyk and Schmid [12] carried out a performance evaluation of local descriptors. The local descriptors were tested on both circular and affine shaped regions with the result of GLOH [12] and SIFT to have the highest performance. For the evaluation they created a database consisting of images of different scene types under different geometric and photometric transformations. The database has later become a benchmark for visual images to use in evaluations and tests of novel detectors and descriptors.

In [6] a thorough evaluation of affine region detectors was performed by Mikolajczyk et al. in 2005. Here the focus was to evaluate the performance of affine region detectors under the image condition changes: scale, view-point, blur, rotation, illumination and JPEG compression. The detectors included in the evaluation were Harris-Affine [13-15], Hessian-Affine [13,14], MSER [16], EBR [17], IBR [18] and Salient regions [19]. Best performance in many cases was obtained by MSER followed by Hessian-Affine and Harris-Affine. To match found interest points between images the SIFT descriptor was used since it showed good performance in [12].

A third evaluation was performed in 2012 by Miksik and Mikolajczyk [7] containing both detectors and descriptors with focus on fast feature matching. The evaluation contained the binary methods ORB [10] and BRISK [20], which are developments of both detectors and descriptors. It also included the binary descriptor BRIEF [21] and the real valued descriptors SIFT, SURF, MROGH, MRRID [22] and LIOP [23]. To perform comparable tests of the descriptors they were all used on IPs obtained by the SURF detector. The result of the comparison of descriptors shows that the novel real valued descriptors LIOP, MRRID and MROGH outperform state-of-the-art descriptors as SIFT and SURF at the expense of decreased efficiency. A brief comparison of the run time of detectors as FAST [24], SURF, SIFT, STAR etc. was also completed. The obtained result confirmed the result
in [10] that FAST and detectors based on the same detector are one to two orders of magnitude faster than detectors as SURF and SIFT.

Similar tests in performance, as in above evaluations, are commonly performed in papers introducing novel detectors or descriptors, e.g. [20,25]. Common for performance tests are that these novel detectors and descriptors are compared to the well known, patented, SIFT and SURF detectors and descriptors due to their high performance and popularity.

Studies made on IR images are mainly on face recognition [26–28] and object detection/tracking [29,30]. Other fields that have been studied are image enhancement of visual images using near-infrared images [2], IP detection on faces using the Harris interest point detector [31], a feature point descriptor for both far-infrared and visible spectrum images [32] and last a scale invariant IP detector of blobs tested against common detectors on IR images [33]. The latter paper does not use the benchmark evaluation setup and can therefore not be compared to other results in a proper way.

The most analogous study to this thesis, within IR image analysis, were made by Ricuarte et al. in [34]. It is an evaluation of classical feature point descriptors in both IR images and visual images under image condition changes as rotation, blur, noise and scale. Included descriptors were SIFT, SURF, ORB and BRISK combined with their corresponding detectors as well as BRIEF and FREAK descriptors used on regions detected by the SURF detector. A stereo rig is used to capture both visual images and IR images of the same scenes which later are used to compare the performances of the descriptors in the two spectral bands. A conclusion in the paper was that no descriptor outperformed the other, but SIFT performed among the best with good performance in most of their tests. They also conclude that it seems like their results could have been improved by preprocessing the images. Compared to evaluations made on visual images, this evaluation does neither test the methods under view-point changes nor the performance of detectors themselves or different combinations of detector and descriptor. In their evaluation they do not use the established evaluation framework from [6,12]. LWIR images were captured by a Gobi-640-GigE camera from Xenics, compared to the FLIR T640, FLIR P660 and FLIR SC660 used in this work.

Despite these studies, especially the latter, a thorough evaluation of the performance of combinations of detectors and descriptors on IR images is missing.

2.4 Discussion

From the examples of changing properties of glass and polished surfaces under Section 2.1.1 and Section 2.1.2 it can be interpreted that the appearance of objects may differ in visual images and infrared images. Typically visual images also contain more high frequency information, such as edges and corners. This is also illustrated in Figure 2.4 where it can be observed that the information content in the two images vastly differs. In such a situation a combination of detector and descriptor suitable for the visual image may not be the optimal choice for the IR images.
Figure 2.4: Visible image and IR image of the same sign illustrating possible differences in object appearance.
Chapter 3

Evaluated Detectors and Descriptors

This chapter describes how the selection of detectors and descriptors are prioritized and also gives a brief introduction to selected methods.

3.1 Priority of Detectors and Descriptors

The choice of descriptors and detectors to include in the evaluation depends on performances in earlier evaluations on visual images as well as how commonly they are used. Influencing the selection is to include methods from the categories corner detectors, blob detectors, gradient based descriptors and binary descriptors. This is to get a good variation of methods with different properties in the evaluation. Of interest is also to find the differences in performance between floating point combinations and binary point combinations.

The final choice of detectors and descriptors to be evaluated can be found in Table 3.1. These detectors and descriptors are combined in different set-ups to evaluate which detector, descriptor and combination has the highest performance on IR images. For an overview of considered detectors and descriptors see Appendix A.

<table>
<thead>
<tr>
<th>Detector</th>
<th>Type</th>
<th>IP</th>
<th>Detector</th>
<th>Type</th>
<th>Utilize</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSER</td>
<td>Floating point</td>
<td>Blob</td>
<td>SIFT</td>
<td>Floating point</td>
<td>Gradients</td>
</tr>
<tr>
<td>Harris-Affine</td>
<td>Corner</td>
<td></td>
<td>SURF</td>
<td>Corner</td>
<td></td>
</tr>
<tr>
<td>Hessian-Affine</td>
<td>Blob</td>
<td></td>
<td>LIOP</td>
<td>Intensities</td>
<td></td>
</tr>
<tr>
<td>SIFT</td>
<td>Corner</td>
<td></td>
<td>ORB</td>
<td>Binary point</td>
<td></td>
</tr>
<tr>
<td>SURF</td>
<td>Corner</td>
<td></td>
<td>BRISK</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FAST</td>
<td>Corner</td>
<td></td>
<td>BRIEF</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ORB</td>
<td>Corner</td>
<td></td>
<td>FREAK</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BRISK</td>
<td>Corner</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 3.1: Detectors and descriptors to be evaluated.
3.2 Introduction to Evaluated Detectors and Descriptors

3.2.1 Corner Based Detectors

Harris-Affine

Harris-Affine [14] is a corner detector developed to be invariant to scale, rotation and affine transformations. It is based on Harris-Laplace, a scale invariant extension of the reliable Harris corner detector. Harris-Laplace is extended to Harris-Affine by an iterative affine shape estimation of the feature.

To extract interest points the detector uses the second order matrix in Equation 3.1

\[
M = \mu(x, \sigma_I, \sigma_D) = \begin{bmatrix} \mu_{11} & \mu_{12} \\ \mu_{21} & \mu_{22} \end{bmatrix} = \sigma_D^2 g(\sigma_I) \begin{bmatrix} I_x^2(x, \sigma_D) & I_x I_y(x, \sigma_D) \\ I_x I_y(x, \sigma_D) & I_y^2(x, \sigma_D) \end{bmatrix}
\]  

(3.1)

where \(x\) is the location of the interest point, \(\sigma_I\) the integration scale for a Gaussian kernel in a smoothing step and \(\sigma_D\) the differentiation scale of the Gaussian kernel used to calculate the derivatives of image \(I\). Features are only considered as corners for the local maxima of the cornerness measure in Equation 3.2 where \(\alpha\) is a constant.

\[
cornerness = \det(\mu(x, \sigma_I, \sigma_D)) - \alpha \text{trace}^2(\mu(x, \sigma_I, \sigma_D))
\]  

(3.2)

From the features extracted by fulfilling the cornerness measure, the final features are obtained from the affine estimation step by an iterative usage of the second order matrix.

Features from Accelerated Segment Test - FAST

Inspired by the SUSAN algorithm, FAST [24] is a corner detector based on intensity comparisons. Corners are detected by comparing the intensity of the center pixel to pixels in the border of the examined region. If enough of consecutive pixels are all either higher or lower than the center value a corner is detected.

A common FAST method is FAST 9-16, which imply that 9 out of 16 consecutive pixels must be either higher or lower than the center pixel intensity. One of the reasons FAST is so efficient is in the first steps in the intensity comparison. In the case of FAST 9-16 the first pixels checked are pixels 1,5,9,13. If pixel \(p\) is a corner at least three of these pixels must be brighter or darker than the intensity of \(p\).

3.2.2 Blob Based Detectors

Hessian-Affine

The procedure for extracting blobs with Hessian-Affine [14] is much related to Harris-Affine. Equivalent to Harris-Affine, Hessian-Affine is an extension of Hessian-Laplace. The difference is that Hessian-Affine makes use of the Hessian matrix,

\(\text{In } [13] \alpha \text{ is set to } 0.06.\)
3.2. INTRODUCTION TO EVALUATED DETECTORS AND DESCRIPTORS

Equation 3.3 instead of the second order matrix. To select scale and location of
local features, the determinant and trace of the Hessian matrix is employed.

\[
H = H(x, \sigma_D) = \begin{bmatrix}
  h_{11} & h_{12} \\
  h_{21} & h_{22}
\end{bmatrix} = \begin{bmatrix}
  I_{xx}(x, \sigma_D) & I_{xy}(x, \sigma_D) \\
  I_{xy}(x, \sigma_D) & I_{yy}(x, \sigma_D)
\end{bmatrix}
\] (3.3)

For the affine shape estimation the second order matrix is applied to the same
procedure as for the Harris-Affine detector.

Maximally Stable Extremal Regions - MSER

MSER \[16\] extracts blobs of arbitrary size and shape. It is based on water-shedding
the image by iteratively moving a threshold from the lowest value to the highest, or
the opposite, and assigning all values below the threshold to zero while the values
above are assigned the top value. Blobs are then extracted by choosing those regions
with area variations less than an acceptable limit for a chosen number of iterations.
To be able to make comparisons with other detectors and also apply descriptors on
MSER regions, these are typically approximated by ellipses with the same second
order moments as the arbitrary regions.

3.2.3 Blob Based Detectors and Gradient Based Descriptors

Scale Invariant Feature Transform - SIFT

In \[8\] Lowe presented SIFT which since has become the most popular descriptor due
to its invariance to scale and rotation. Due to its high performance it has become the
target for benchmarking in evaluations as \[7,12,35\] and in articles presenting novel
algorithms. It has therefore also been the preferred descriptor to use in evaluations
of detectors as \[6\].

For feature extraction SIFT makes use of Difference of Gaussians(DoG), an
approximation of the second order derivatives in Laplacian of Gaussian(LoG). DoG
is implemented on a scale space \[36\], created to make SIFT scale invariant.

The descriptor in SIFT divides the region into 16 bins. For each bin the magni-
tudes of gradients in 8 directions are computed. Finally, the resulting descriptor is
a vector of 128 elements storing the magnitudes of each bin in an order depending
on the global orientation of the feature.

Speeded Up Robust Features - SURF

In the development of SURF \[9\], the aim was to keep the performance as for SIFT
but with an increase in efficiency. The detector in SURF is based on the Hessian
matrix as in Hessian-Affine with the difference of only using the determinant to
select location and scale. Instead of calculating the second order derivatives in the
Hessian matrix, approximations are made using box filters and integral images.

Descriptors are created by dividing the regions into 16 bins as in SIFT. The
descriptor components are created from a few regularly spaced points in each bin. In
CHAPTER 3. EVALUATED DETECTORS AND DESCRIPTORS

each point the Haar wavelet response is calculated and eventually a four-dimensional vector is obtained for each bin resulting in a descriptor of 64 elements in total.

3.2.4 Intensity Order Based Descriptor

Local Intensity Order Pooling - LIOP

LIOP is a novel descriptor used with detected regions from Harris-Affine and Hessian-Affine in [23]. The region around the local feature is divided into subregions, which are created by sorting all pixels in an ascending order based on intensity. Next, the list of pixels is divided into a predetermined number of bins of equal size. For each pixel the intensity values of neighbouring pixels are used and mapped to an index. These indexes create the histogram for each subregion which then are concatenated to one long vector of 144 elements.

3.2.5 Binary Descriptors

Binary Robust Independent Elementary Features - BRIEF

Common for binary descriptors is to use sampling points for intensity comparisons. The comparisons are performed in a boolean approach checking which of two pixels intensities is higher. The output is a binary vector storing the result of all comparisons.

In BRIEF [21] the sampling procedure consists of 128 predetermined comparisons, which were randomly selected when developed.

Fast Retina Keypoint - FREAK

The novelty in FREAK [25] is the developed sampling pattern which is inspired by the human retina and has a higher density of sampling points near the center, observable in Figure 3.1. Analogous to ORB, FREAK employs learning based sampling points which maximize the variance of the pairs. To make FREAK rotation invariant a local feature’s orientation is calculated through estimated local gradients from a symmetric pattern similar to BRISK.

3.2.6 Corner Based Detectors and Binary Descriptors

Oriented FAST and Rotated BRIEF - ORB

As the name announces ORB [10] is based on an extension of FAST, oFAST, which efficiently assigns an orientation to the IP using the intensity centroid. To make ORB robust to scale changes a multi-scale detector is created using a scale space pyramid. A scale space pyramid contains images of different sizes, octaves, and of varying strength in blur, scales, which makes it possible to detect features of different sizes.
3.2. INTRODUCTION TO EVALUATED DETECTORS AND DESCRIPTORS

Figure 3.1: FREAK pattern, replicated from [25].

Figure 3.2: BRISK pattern, replicated from [20].

The descriptor used in ORB is an extension of BRIEF, rBRIEF, which makes it aware to rotation. In rBRIEF rotation invariance is obtained by steering the descriptor according to the orientation of the IP. Unlike BRIEF, ORB uses a learning based method to choose a good subset of binary tests.

Binary Robust Invariant Scalable Keypoints - BRISK

To detect features in BRISK [20] another extension of FAST is employed. The detector is AGAST, which is similar to the approach in FAST with the difference of applying an optimal decision tree for pixel comparisons. Similar to ORB the local maximum of detected points is searched for in a scale space pyramid to achieve scale invariance. Different from other usages of the scale space pyramid, where one chooses the discrete value of the scale, this implementation interpolates between the layers to find the continuous maxima.

To describe the information content of an IP BRISK uses the predefined sampling pattern observable in Figure 3.2 consisting of 60 sampling points. A set of long distance comparisons are used to estimate the orientation of the local feature while an independent set of short distance comparisons are used to create the descriptor.

Discussion

Worth pointing out from the brief introduction of the various methods is that the extracted regions for different detectors differ in size and shape. Since the methods differ in their implementations the amount of detected features are also naturally different which will be further noticed from the results in Chapter 5.
Chapter 4

Evaluation Setup

This chapter presents the evaluation framework as well as how the database of IR images is created.

4.1 Database

When creating the database an important aspect is to include images with various properties. The images contained in the database can be divided into the categories textured and structured scenes. A textured scene has repetitive patterns of different shapes while a structured scene has homogeneous regions with distinctive edges. Examples of structured scenes are presented in the left column of image pairs in Figure 4.1 and of textured scenes in the right column of image pairs.

Out of the standard images, taken by the cameras, the database is created by synthetic modification to include the desired image condition changes. An exception is for view-point changes where all images were captured with the FLIR T640 camera without modification. In Figure 4.1 are examples of image pairs for all deformations to be evaluated. Each image pair displayed consists of a reference image, left, and a test image, right.

4.1.1 Deformation Specification

View-point Images are taken from different view-points starting at 90° angle to the object. The maximum view-point angle is about 50-60° relative to the starting angle.

Scale Zoom is imitated by scaling the height and width of the image. The region of the zoomed image is then extracted by cropping the initial image around the center point with the new scaled height and width. To achieve images of the same dimensions as the standard images, the extracted region is resized using bilinear interpolation. The zoom of the image is in the range 1.25x-2.5x zoom with 0.25x increments.
CHAPTER 4. EVALUATION SETUP

### Table 4.1: Spectral range and resolution of cameras used to create the database.

<table>
<thead>
<tr>
<th>Camera</th>
<th>FLIR T640</th>
<th>FLIR P660</th>
<th>FLIR SC660</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spectral range (µm)</td>
<td>7.5-14</td>
<td>7.5-13</td>
<td>7.5-13</td>
</tr>
<tr>
<td>Resolution (pixels)</td>
<td>640x480</td>
<td>640x480</td>
<td>640x480</td>
</tr>
</tbody>
</table>

**Rotation** Rotated images are created by rotating the standard images with 10° increments. The size of the reference image and rotated test images are constrained to have a diagonal of the same length as the height of the initial captured image. This lead to a maximum size of the rotated images without getting any padded information.

**Blur** Images are blurred using a gaussian kernel of size 51x51 pixels and standard deviation between 1-10 pixels.

**Noise** Noise is added to images by inducing white gaussian noise of increasing variance from 0.0001 to 0.005 digital number values with 0.0005 increments if the image is normalized between 0 and 1.

**Downsampling** Images are downsampled to three reduced sizes; by a factor of 2, 4 and 8.

#### 4.1.2 IR Cameras

The IR cameras used to create the database are a FLIR T640, a FLIR P660 and a FLIR SC660. In Table 4.1 their spectral range and resolution are displayed. As can be interpreted all cameras operate in the LWIR band and have a 640x480 pixel resolution.

#### 4.2 Evaluation Framework

The evaluation frameworks used in [6][12] have become well established evaluation methods for measuring the performance of detectors and descriptors. When novel detectors and/or descriptors are presented these frameworks are therefore commonly used. Consequently these frameworks are chosen to ensure reliability and comparability of the results in this work.

Performance measures of detectors and descriptors are much related to what describes a good feature. The most common measure of detectors is repeatability while for descriptors and combinations the accuracy measures, recall, 1-precision and matching score. These measures are further discussed under Section 4.2.4.
4.2. EVALUATION FRAMEWORK

Figure 4.1: Dataset of images, under various deformations, that are included in the database. Each image pair consists of a reference image, left, and a test image, right. (a)-(b) Viewpoint, (c)-(d) Rotation, (e)-(f) Scale, (g)-(h) Blur, (i)-(j) Noise, (k)-(l) Downsampling. In the left column of image pairs are examples of structured scenes while the left column give examples of textured scenes.
4.2.1 Ground Truth

In order to evaluate the repeatability and accuracy of the algorithms the ground truths of the correct matches between test images and reference images are needed. A ground truth in this work is created by mapping the detected regions in a test image to the reference image of a scene using a 2D-linear transformation, homography. Usage of homographies is valid if images are of planar scenes or for camera rotations around its center. Since most evaluations are performed on the benchmark dataset created in [12] it is also most common to use homographies to create the ground truth.

As most of the image deformations are synthetically generated so can most of the homographies. Synthetically generated homographies are created for scaled, rotated and downsampled images. In the case of noise and blur deformations no homography is needed as only the quality of the image is modified. To create homographies for view-point changes four point pairs between the test image and the reference image are selected by hand. These four point pairs are related as in Equation 4.1, with short representation in Equation 4.2. The sought homography \( h \) is obtained through the least squares solution of the equation.

\[
\begin{bmatrix}
  x_1 & y_1 & 1 & 0 & 0 & 0 & -x_1 x'_1 & -x_1 y'_1 \\
  0 & 0 & x_1 & y_1 & 1 & -x_1 y'_1 & -y_1 y'_1 \\
  x_2 & y_2 & 1 & 0 & 0 & 0 & -x_2 x'_2 & -y_2 y'_2 \\
  0 & 0 & x_2 & y_2 & 1 & -x_2 y'_2 & -y_2 y'_2 \\
  x_3 & y_3 & 1 & 0 & 0 & 0 & -x_3 x'_3 & -y_3 y'_3 \\
  0 & 0 & x_3 & y_3 & 1 & -x_3 y'_3 & -y_3 y'_3 \\
  x_4 & y_4 & 1 & 0 & 0 & 0 & -x_4 x'_4 & -y_4 y'_4 \\
  0 & 0 & x_4 & y_4 & 1 & -x_4 y'_4 & -y_4 y'_4 \\
\end{bmatrix}
\begin{bmatrix}
  h_{11} \\
  h_{12} \\
  h_{13} \\
  h_{14} \\
  h_{21} \\
  h_{22} \\
  h_{23} \\
  h_{24} \\
  h_{31} \\
  h_{32} \\
\end{bmatrix}
= \begin{bmatrix}
  x'_1 \\
  y'_1 \\
  x'_2 \\
  y'_2 \\
  x'_3 \\
  y'_3 \\
  x'_4 \\
  y'_4 \\
\end{bmatrix}
\]

\( (4.1) \)

\[
A h = b
\]

\( (4.2) \)

Correspondences

The correspondences between a test image and a reference image are the ground truth of the correct corresponding IPs in the images. When projecting the detected IPs from the test image to the reference image overlapping regions are obtained as in Figure 4.2. For two regions to be considered as a correspondence the overlap error of the regions must be less than a predefined threshold, \( \epsilon \), in percentage. The overlap error is calculated as in Equation 4.3

\[
\text{overlap error} = 1 - \frac{A \cap B}{A \cup B}
\]

\( (4.3) \)

\[
\text{overlap error} < \epsilon \rightarrow \text{correspondence}
\]

\( (4.4) \)

where \( A \) and \( B \) are the two overlapping regions in Figure 4.2. In Figure 4.3 an example of a possible correspondence and denied correspondence are given. To remember
4.2. EVALUATION FRAMEWORK

is that the threshold $\epsilon$ affects if these regions are considered as correspondences or not.

4.2.2 Matching

To realize performance measures for descriptors matching between different descriptors is needed. For floating point descriptors matches are obtained by calculating the Euclidean distance between all descriptors followed by selecting the nearest neighbours (NN) as matches. In the case of binary descriptors the Hamming distance is used as a similarity measure followed by the same matching step as for floating point descriptors. The Hamming distance is applied as its result is equal to the square of the Euclidean distance for binary vectors, but with a decrease in complexity and hence an increase in efficiency for binary feature matching.

To become a NN match, in this work, the two candidates have to be the closest descriptors in descriptor space for both descriptors. An example is if descriptor $A$ has descriptor $B$ as its closest neighbour, $B$ has descriptor $C$ and $C$ has $B$ as its closest neighbour. Then $B$ and $C$ would be selected as NN while $A$ would be discarded or matched with another descriptor. A descriptor is only allowed to be matched once, such that a descriptor can not be associated with numerous descriptors.

Out of the acquired matches, correct matches are obtained by comparing the result to the ground truth correspondences.

4.2.3 Region Normalization

When evaluating the performance of descriptors a measurement region different from the detected region is used. As in [12] the measurement region is chosen to be a scaled version of the detected region by a factor of three. The motivation is that e.g. blob detectors as Hessian-Affine and MSER extract regions with large signal variations in the borders. To increase the distinctiveness of extracted regions the measurement region is therefore increased to include larger signal variations. This scale factor is applied to all detectors. A side-effect from scaling the measurement region is its possibility of reaching outside the image border, e.g. the dark region
CHAPTER 4. EVALUATION SETUP

(a) Example of normalized patch with measurement region outside of image border.

(b) Example of extended region normalization used in this work.

Figure 4.4

to the left in Figure 4.4a. In this work an extension of the region normalization used in [6], source code available at [37], is implemented to expand the image by assigning values to the unknown area by bilinear interpolation on account of the border values. The idea is that the unknown area is more likely to be similar to the nearby border values than all dark. A result of the procedure applied to the image in Figure 4.4a is presented in Figure 4.4b.

As detected regions are of circular or elliptical shape all regions are normalized to circular shape of constant radius to become scale and affine invariant. In [12] a diameter of 41 pixels is chosen arbitrary while in this work all regions are normalized to have a diameter of 49 pixels. The choice of a larger diameter is based on to use the standard settings in the OpenCV library for the BRIEF descriptor.

4.2.4 Performance Measures

Repeatability

Repeatability is used as a performance measure for detectors. As described in Section 2.2.1 it is a measure of how well a detector performs in extracting the same features in images under different image condition changes. The repeatability is defined in Equation 4.5 as the ratio of the number of correspondences and the number of detected features, where only detected features in the parts represented in both images are taken into account. In the ideal situation 100% is obtained.

\[
\text{Repeatability} = \frac{\text{Number of correspondences}}{\text{Smallest number of detected regions in the two images}} \quad (4.5)
\]

Recall

Recall is a measure of the ratio of correct matches and correspondences. Since the measure is about matching features, it is a performance measure for how well descriptors are able to match extracted features. In the ideal situation 100% is
4.3. IMPLEMENTATION DETAILS

obtained.

\[
Recall = \frac{\text{Number of correct matches}}{\text{Number of correspondences}} \quad (4.6)
\]

1-Precision

The 1-Precision measure portraits the ratio between the number of false matches and total number of matches. As the name describes it is the opposite of a precision measure. In the ideal situation 0% is achieved.

\[
1\text{-Precision} = \frac{\text{Number of false matches}}{\text{Total number of matches}} \quad (4.7)
\]

Matching Score

The matching score for a descriptor is similar to recall with the difference of calculating the ratio between the number of correct matches and total matches. Here the number of total matches is restrained by the least number of detected features in the two images.

\[
\text{Matching score} = \frac{\text{Number of correct matches}}{\text{Number of total matches}} \quad (4.8)
\]

4.3 Implementation Details

The evaluation is performed on Windows 7 installed on a virtual machine through Parallels Desktop 10, with the settings of 4 CPU’s and 4 GB RAM. Parallels Desktop is running on a Macbook Pro mid 2014 with OS X Yosemite version 10.10.3, a 2.6 GHz Intel Core i5 processor and 8 GB 1600 MHz DDR3 RAM.

Local features are extracted using OpenCV [38] version 2.4.10 and VLFeat [39] version 0.9.20 libraries. OpenCV implementations are used for SIFT, SURF, MSER, FAST, ORB, BRISK, BRIEF and FREAK while Harris-Affine, Hessian-Affine and LIOP are VLFeat implementations. Unless explicitly stated the parameters are the ones suggested by the authors. MATLAB R2014b is used as a central platform employing OpenCV and VLFeat through MEX files, MATLAB Executables, which provide an interface between MATLAB and subroutines written in C, C++ or Fortran.

IR images are loaded into MATLAB using FLIR’s Atlas SDK, supported only for MATLAB on the Windows operating system and hence the major factor to carry out the evaluation on the Windows operating system. When loaded in MATLAB the IR images contain 16 bit data which are quantized into 8 bit data to work with above mentioned libraries. Before quantization the images, which contain floating point numbers, are preprocessed by histogram equalization and normalized to span between 0 and 255.

To calculate the overlap error, repeatability, recall and 1-precision this work utilizes code from [37], the established evaluation framework, with the extension of using Hamming distance in the matching step for binary descriptors.
4.3.1 Parameter Selection

In VLFeat implementations of Harris-Laplace and Hessian-Laplace exist with the possibility of performing affine shape estimation. When to invoke the detectors functions there are parameters available to control a peak threshold and edge threshold.

The peak threshold affects the minimum acceptable cornerness measure, defined in Equation 3.2, for a feature to be considered as a corner in Harris-Affine and equivalently a blob feature by the determinant of the Hessian matrix in Hessian-Affine. According to the authors in [13], a predecessor to [14], the used value for the threshold on cornerness was 1000. As no similar value is found to the Hessian-Affine threshold it is selected to 150. With the selected threshold the number of extracted features is in the order of magnitude as other detectors in the evaluation.

The edge threshold is an edge rejection threshold and eliminates points with too small curvature. As this threshold is not related to any parameter by the authors it is selected to the predetermined value of 10 in the library.
Chapter 5

Evaluation Results

In this chapter the evaluation results are presented and discussed. It is divided to first introduce the results of detectors in Section 5.1, followed by the results of descriptors in Section 5.2 and last combinations of detectors and descriptors in Section 5.3.

We use the same dataset in the evaluations of the three categories, the 12 scenes displayed in Figure 4.1, which consists of 12 reference images and 96 test images.

5.1 Detectors

In the evaluation of detectors we allow an overlap error of 40% as in the evaluation of affine detectors on visual images in [6]. Detectors are evaluated by repeatability, the number of correspondences, the number of correct matches and the matching score. To check the detectors performances in a matching procedure combined with the same descriptor, they are all tested together with LIOP to calculate the matching score and the number of correct matches. We choose the LIOP descriptor as it overall has outperformed the other descriptors in [7]. In the ideal situation the performance curves are horizontal lines as the deformation strength increases, indicating that the detector is invariant to the deformation.

View-point

Results for view-point changes in the structured scene in Figure 4.1a are displayed in Figure 5.1. On the x-axis are the ascending image numbers illustrating evenly increased perspective angles between about 5° to 50-60°. As can be interpreted all detectors are subject to a decrease in performance as the perspective angle increases. Noteworthy is that in image number six, containing the largest perspective angle, only MSER, Harris-Affine and Hessian Affine are able to maintain any repeatability, correspondences, matching score and correct matches.

For the textured scene with a wall of bricks in Figure 4.1b, the result is presented in Figure 5.2. In this scene all detectors show slightly less dependency to perspective
Figure 5.1: View-point in a structured scene. (a) Repeatability relative the perspective angle, (b) the number of correspondences between test image and reference image, (c) matching score for detectors with LIOP descriptor and (d) the number of correct matches.

changes by their more horizontal behaviour and higher matching score than in the structured scene. Hessian-Affine indicates to be more sensitive to the textured scene than to the structured scene by its decrease until the peak in image 5, approximately 45°. This increased sensitivity can be considered as related to the much lower number of detected features in this scene. At image number 4, approximately 30°, the detectors SIFT and FAST start to decrease rapidly in repeatability while the remaining detectors among the top performers decrease after image 5.

Overall best performances for the two scenes are obtained by Hessian-Affine, SURF, ORB and Harris-Affine. Hessian-Affine has the best performance followed by SURF and ORB considered to be equally good. The two detectors show similar tendencies in both scenes with their drawback of large decreases in the structured scene. Harris-Affine is considered to come in the third with good performance in both scenes with less dependency than SURF and ORB but has a lower repeatability and matching score.
5.1. DETECTORS

Figure 5.2: View-point in a textured scene. (a) Repeatability relative the perspective angle, (b) the number of correspondences between test image and reference image, (c) matching score for detectors with LIOP descriptor and (d) the number of correct matches.

Rotation

The effect of rotation as image transformation is presented in Figure 5.3, for the structured scene in Figure 4.1c, and in Figure 5.4, for the textured scene in Figure 4.1d.

In case of repeatability FAST performs best followed by Hessian-Affine and Harris-Affine with ORB as a close competitor. The detector to be most dependent to rotation angle is SURF, also concluded in evaluations on visual images, although other detectors show small effect by rotation as well with small peaks at 90° rotation.

Studying Figure 5.3c and Figure 5.4c shows that MSER, SIFT, FAST and ORB have a decrease in performance of matching score for increase in rotation angle. In the textured scene the matching score loss in ORB for increase in rotation is not as large as for the other three detectors. For FAST the decrease in matching score might depend on the small size of the features extracted which leads to low distinctiveness.
CHAPTER 5. EVALUATION RESULTS

Figure 5.3: Rotation in a structured scene. (a) Repeatability relative the rotation angle, (b) the number of correspondences between test image and reference image, (c) matching score for detectors with LIOP descriptor and (d) the number of correct matches.

By studying the number of correspondences it is clear that the number of detected features may significantly differ between two scenes. For this scene pair the number of correspondence in the textured scene is much larger than in the structured scene. For example, to MSER the difference is greater than an order of 10.

Scale

When it comes to scale all evaluated detectors are developed to be invariant or robust to the image transformation except FAST. This becomes evident when examining the result in Figure 5.5, structured scene in Figure 4.1e, and Figure 5.6, textured scene in Figure 4.1f, as all measures reach zero at a scale of 1.5.

In the structured scene best performance is achieved by MSER followed by SIFT and Hessian-Affine when it comes to repeatability. It is worth mentioning that SURF, ORB and Harris-Affine are also close competitors. Examining the matching
5.1. DETECTORS

Figure 5.4: Rotation in a textured scene. (a) Repeatability relative the rotation angle, (b) the number of correspondences between test image and reference image, (c) matching score for detectors with LIOP descriptor and (d) the number of correct matches.

score with the LIOP descriptor, SURF is the top performer followed by SIFT and Hessian-Affine. SURF, SIFT and Hessian-Affine are considered to perform better than MSER as they indicate a more stable behaviour. As for the textured scene MSER has a higher performance compared to the structured scene. Here the top performers are SURF, SIFT, MSER and Hessian-Affine.

When examining the results of repeatability and matching score a majority of the detectors have peaks at 2x scale. This is thought of as a point at where the scale of the image coincides well with scales in e.g. the scale space pyramid leading to better similarities between extracted features in the reference image and test image of that particular scale.

Blur

In Figure 5.7 and Figure 5.8 are the results for the structured scene in Figure 4.1g and the textured scene in Figure 4.1h. The results clearly show that ORB out-
Figure 5.5: Scale in a structured scene. (a) Repeatability relative the scale factor, (b) the number of correspondences between test image and reference image, (c) matching score for detectors with LIOP descriptor and (d) the number of correct matches.

performs its relatives BRISK and FAST which quickly drop in both repeatability and matching score. At a blurring level of standard deviation larger than 3 no correspondences are found by FAST. This is also visible in the repeatability curves shown by no markers from standard deviation 4, a result from dividing by zero. Overall it is evident that the number of correspondences is greatly affected by blur as the slopes of the lines in Figure 5.7b and Figure 5.8b are steep.

MSER, described in [6] to be sensitive to blur, again shows sensibility to the deformation. This is due to borders, which enfold areas of interest, get smoothed out leading to too large area variations to become stable regions. In the structured scene the repeatability is especially irregular which is caused by its low number of correspondences. Its matching scores show similar behaviour in both scenes where it becomes irregular after standard deviation of 5. The peak which turns up in the repeatability curve in the structured scene at standard deviation 6 is caused by the very low number of correspondences found by MSER.

Best performance is obtained by SURF followed by Hessian-Affine and ORB.
5.1. DETECTORS

Figure 5.6: Scale in a textured scene. (a) Repeatability relative the scale factor, (b) the number of correspondences between test image and reference image, (c) matching score for detectors with LIOP descriptor and (d) the number of correct matches.

SURF is considered to be performing better than Hessian-Affine as it is slightly more horizontal and has a significantly higher matching score in the textured scene.

Noise

The results of detectors when affected by noise are presented in Figure 5.9 and Figure 5.10 for the structured and textured scenes in Figure 4.11 and Figure 4.13, respectively.

By examining the number of correspondences in both scenes, all detectors are consistently having almost horizontal curves except SIFT which has a small drop in performance. For small variances FAST is evolving differently than the other detectors with an increase in correspondences and repeatability. A reason might be that with induced noise more corners are typically detected and hence there is a possibility of regions overlapping by accident. The number of detected features in the reference image is then the lower bound for the possible amount of corre-
Figure 5.7: Blur in a structured scene. (a) Repeatability relative the kernel standard deviation, (b) the number of correspondences between test image and reference image, (c) matching score for detectors with LIOP descriptor and (d) the number of correct matches.

In the structured scene the performance in matching score does not differ much to the performances of the other detectors while in the textured scene a large difference is visible. This is thought to be due to low distinctiveness in the measurement regions extracted from the FAST detector as by nature it only detects features of a small fixed size. With added noise the risk of false positives is therefore increasing in a matching step.

Best performance in repeatability is achieved by Hessian-Affine followed by SURF and ORB, excluding FAST due to the improbable result. In matching score the best performance is obtained by ORB followed by Hessian-Affine and SURF with Harris-Affine closely behind. The worst performance is observed for SIFT.

**Downsampling**

The effect of downsampling on the performance can be studied in Figure 5.11 for the structured scene in Figure 4.1K and in Figure 5.10 for the textured scene in Fig-
5.1. DETECTORS

![Graphs showing performance of different detectors under various conditions.](image)

Figure 5.8: Blur in a textured scene. (a) Repeatability relative the standard deviation, (b) the number of correspondences between test image and reference image, (c) matching score for detectors with LIOP descriptor and (d) the number of correct matches.

We observe best performance by SURF both in repeatability and matching score. Consecutive performers in case of repeatability are SIFT, Hessian-Affine and Harris-Affine. However, when studying the matching score the consecutive top performers to SURF are Harris-Affine and Hessian-Affine with SIFT having a much lower matching score.

Just as for scale changes FAST has obvious poor performance. This is expected as downsampling an image also scales the content in the image. When downsampling an image information is lost as can be confirmed by the decrease in the number of correspondences for increase in downsampling factor. Just as FAST, ORB fails in extracting any IPs at a downsampling factor of 8 resulting in zero repeatability.

5.1.1 Summary

To summarize the performances of detectors in this section Table 5.1 displays the top three performers in each evaluation category. In order to give an idea on the
comparative performance, the top performers are given a score from one to three resulting in the highest score for the best detector. Examining the table we observe an overall good performance for Hessian-Affine while BRISK and FAST are bottom performers, confirmed by studying their low results in the evaluation. High performance is also achieved by SURF although with a high sensitivity to rotation.

As expected the number of correspondences varies between textured scenes and structured scenes with overall higher number of extracted points for textured scenes. It is also clear that the amount of extracted features differs for detectors from scene to scene.

In Table 5.2, we observe the computational efficiency of the detectors measured with compiled code. The increased efficiency among binary detectors is evident if studying the run time where especially FAST and ORB outperform other detectors. The run time is obtained by taking the average of the run time needed on the dataset we use in the evaluation.
5.1. DETECTORS

Figure 5.10: Noise in a textured scene. (a) Repeatability relative the white noise variance, (b) the number of correspondences between test image and reference image, (c) matching score for detectors with LIOP descriptor and (d) the number of correct matches.

<table>
<thead>
<tr>
<th></th>
<th>SIFT</th>
<th>SURF</th>
<th>Harris</th>
<th>Hessian</th>
<th>MSER</th>
<th>FAST</th>
<th>ORB</th>
<th>BRISK</th>
</tr>
</thead>
<tbody>
<tr>
<td>View-point</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>Rotation</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Scale</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>Blur</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Downsampling</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>2</td>
<td>12</td>
<td>5</td>
<td>13</td>
<td>1</td>
<td>0</td>
<td>7</td>
<td>0</td>
</tr>
<tr>
<td>Ranking</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>3</td>
</tr>
</tbody>
</table>

Table 5.1: Summary of performance of detectors. In order to give an idea on the comparative performance, the top performers are given a score from one to three resulting in the highest score for the best detector.
Figure 5.11: Downsampling in a structured scene. (a) Repeatability relative the downsampling factor, (b) the number of correspondences between test image and reference image, (c) matching score for detectors with LIOP descriptor and (d) the number of correct matches.

<table>
<thead>
<tr>
<th>Detector</th>
<th>Run time [ms]</th>
<th># interest points</th>
</tr>
</thead>
<tbody>
<tr>
<td>SIFT</td>
<td>730</td>
<td>1313</td>
</tr>
<tr>
<td>SURF</td>
<td>169</td>
<td>894</td>
</tr>
<tr>
<td>MSER</td>
<td>612</td>
<td>281</td>
</tr>
<tr>
<td>Hessian-Affine</td>
<td>876</td>
<td>4483</td>
</tr>
<tr>
<td>Harris-Affine</td>
<td>900</td>
<td>2655</td>
</tr>
<tr>
<td>FAST</td>
<td>9</td>
<td>3179</td>
</tr>
<tr>
<td>ORB</td>
<td>20</td>
<td>495</td>
</tr>
<tr>
<td>BRISK</td>
<td>41</td>
<td>407</td>
</tr>
</tbody>
</table>

Table 5.2: Efficiency by the detectors as well as the average number of extracted IPs.
Figure 5.12: Downsampling in a textured scene. (a) Repeatability relative the downsampling factor, (b) the number of correspondences between test image and reference image, (c) matching score for detectors with LIOP descriptor and (d) the number of correct matches.
CHAPTER 5. EVALUATION RESULTS

5.2 Descriptors

In this part of the evaluation, descriptors are tested against image deformations and performance is measured by the recall and 1-precision measures. All descriptors are applied to the same extracted IPs by the Hessian-Affine detector due to its good performance in Section 5.1. Different from the evaluation of detectors we allow an overlap error of 50%, following the evaluation framework in [12]. The performances are presented by precision-recall curves and recall relative deformation changes, further called deformation-recall curves, which are similar to the figures of matching score in Section 5.1. Descriptors are titled by a concatenation of the detector and descriptor, e.g. hesorb for Hessian-Affine and ORB.

5.2.1 Precision-Recall Curve

Recall and 1-Precision are commonly combined to visualize the performance of descriptors. It is created by varying an acceptance threshold for the distance between two features in descriptor space. If the threshold is small, one is strict in acquiring correct matches which leads to high precision but low recall. A high threshold means that we accept all possible correspondences which leads to low precision, due to many false positives, and a high recall since all correct matches are accepted.

The interpretation of precision-recall curves is not always straightforward. In the ideal situation a recall equal to 1 is obtained for any precision. In real world applications this is not the case as noise might decrease the similarity between descriptors. Another factor arises as regions can be considered as correspondences with an overlap error up to 50%. Descriptors will then differ due to describing information in areas not covered by the other region. If recall is represented by a horizontal curve it announces that the recall is obtained with high precision. A descriptor with a slowly increasing curve indicates that it is affected by the image deformation.

View-Point

The performances of evaluated descriptors against view-point changes are presented in Figure 5.13 for the structured scene in Figure 4.1a and in Figure 5.14 for the textured scene in Figure 4.1b.

Especially by the results from the structured scene it is visible how the descriptors are affected by view-point changes as the curves continuously increases as the precision decreases. Best recall is obtained by LIOP followed by SIFT and BRISK. Observing the textured scene a higher performance is achieved as in the evaluation of detectors. Again the highest recall is achieved by LIOP followed by BRISK. In the precision-recall curve, Figure 5.14a, LIOP also indicates higher precision as it has a steep increase to then become more horizontal over a longer range. The peak of LIOP at image 5 in Figure 5.14b is due to a drop in the number of correspondences by half while LIOP manages to keep up the number of correct matches.
5.2. DESCRIPTORS

Figure 5.13: Performance due to view-point changes in a structured scene. (a) Precision-Recall curve and (b) Recall relative view-point increase.

Figure 5.14: Performance due to view-point changes in a textured scene. (a) Precision-Recall curve and (b) Recall relative view-point increase.

With the performance shown by BRISK it is the top performer among binary descriptors.

Rotation

In this section the performance results of descriptors in rotated images are presented in Figure 5.15, structured scene in Figure 4.1c, and Figure 5.16, textured scene in Figure 4.1d.

All descriptors reach a point from where the precision-recall curve becomes horizontal. Best performance is achieved by the LIOP descriptor with SURF and SIFT to come after. Both the precision-recall curves and deformation-recall curves indicate that FREAK is most affected by rotation. Overall the descriptors drop in performance in the textured scene compared to the structured scene, where LIOP only indicates a minor decrease. Noteworthy is the decrease in performance by BRISK where it in the textured scene shows a higher dependency to rotation. In
CHAPTER 5. EVALUATION RESULTS

Figure 5.15: Performance due to rotation changes in a structured scene. (a) Precision-Recall curve and (b) Recall relative increase in rotation.

Figure 5.16: Performance due to rotation changes in a textured scene. (a) Precision-Recall curve and (b) Recall relative increase in rotation.

Figure 5.16b we can observe BRISK and FREAK to have a continuous decrease in performance with increase in rotation. The poorest performance is achieved by FREAK.

Scale

The results for descriptors applied to images exposed to scale changes are visualized in Figure 5.17, structured scene in Figure 4.1e, and Figure 5.18, textured scene in Figure 4.1f.

All descriptors perform well on scaled images visualized by the horizontal precision-recall and deformation-recall curves. Best performance is achieved by LIOP followed by SIFT and BRISK. In Figure 5.18b we observe how BRISK reaches a higher recall when stagnated than SIFT but higher sensitivity to scale as it slowly increases while SIFT already reached its horizontal state. The stability of SIFT compared to BRISK is also visible in the deformation-recall curves.

42
5.2. DESCRIPTORS

Figure 5.17: Performance due to scale changes in a structured scene. (a) Precision-Recall curve and (b) Recall relative increase in scale.

Figure 5.18: Performance due to scale changes in a textured scene. (a) Precision-Recall curve and (b) Recall relative increase in scale.

The worst performance is again observed for FREAK.

**Blur**

The effect on the performance of descriptors caused by smoothing the images with a gaussian kernel is presented in Figure 5.20 for the textured scene in Figure 4.1h. For this scene the performance is overall poorer than for the other evaluated image deformations with the exception of BRISK outperforming the other descriptors. Studying the deformation-recall curve the best performance is confirmed to be achieved by BRISK followed by LIOP.

In the structured scene, Figure 4.1g, the result is presented in Figure 5.19 where the performance is overall good with many descriptors of equal performance indicated by coincidental horizontal lines at the same recall in the precision-recall curve. In the deformation-recall curve we can confirm that BRISK has the highest performance followed by LIOP as indicated in Figure 5.19b. We can also observe
CHAPTER 5. EVALUATION RESULTS

Figure 5.19: Performance due to blur in a structured scene. (a) Precision-Recall curve and (b) Recall relative increase in blur.

Figure 5.20: Performance due to blur in a textured scene. (a) Precision-Recall curve and (b) Recall relative increase in blur.

the irregular tendencies for the scene.

The reason for the big difference in performance in the textured and structured scene is likely due to many small regions detected in the textured scene. When the scene is smoothed the distinctiveness decreases and a majority of the descriptors become very similar leading to many false positives. In the structured scene a greater diversity of detected IPs is extracted with better recall as the result.

Noise

In this section the results of descriptors applied to images with induced noise are presented. The results are illustrated in Figure 5.21 for the structured scene in Figure 4.1i and in Figure 5.22 for the textured scene in Figure 4.1j.

All descriptors show good performance to the level of induced noise in the evaluation, hence are invariant or have high robustness to the image deformation. The results are similar in the structured and textured scene with LIOP and BRISK as the
5.2. DESCRIPTORS

Figure 5.21: Performance due to noise in a structured scene. (a) Precision-Recall curve and (b) Recall relative increase in noise.

Figure 5.22: Performance due to noise in a textured scene. (a) Precision-Recall curve and (b) Recall relative increase in noise.

top performers. When studying the deformation-recall curves in Figure 5.21b and Figure 5.22b FREAK suffers from a steeper decrease in performance than the other descriptors. Further, in the precision-recall curves we can observe how FREAK has among the higher uncertainties; an indication that another descriptor like SIFT has overall better performance.

Downsampling

The effect of downsampling on the performance of descriptors are shown in Figure 5.23 and Figure 5.24 for the scenes in Figure 4.1k and Figure 4.1l.

Broadly downsampling is handled well by the descriptors showing stable horizontal behaviour in the precision-recall curves. Best performance is obtained by LIOP and BRISK which outperform the other descriptors in the sense of recall relative precision. The other descriptors also tackle downsampling but not at the level of LIOP and BRISK. Studying the deformation-recall curves also indicate a
CHAPTER 5. EVALUATION RESULTS

Figure 5.23: Performance due to downsampling of a structured scene. (a) Precision-Recall curve and (b) Recall relative increased downsampling factor.

Figure 5.24: Performance due to downsampling of a textured scene. (a) Precision-Recall curve and (b) Recall relative increased downsampling factor.

higher performance by LIOP and BRISK, although LIOP drops in performance at the downsampling factor of 8 in Figure 5.23b. Among the descriptors with lower performance, SIFT performs the best.

5.2.2 Summary

The performances of the descriptors are summarized in a similar approach as detectors in Table 5.3. Marked in the table are the top three performing descriptors with corresponding score for each image deformation in the same approach described in Section 5.1.1. For blur, only the top two performers are displayed as the other descriptors were equally good.

Studying the result in the table it is evident that LIOP outperforms the other descriptors in almost all categories. Similarly BRISK is a good alternative as a binary point descriptor.

In Table 5.4 we present the memory requirements for each descriptor as well
as the computational efficiency. We can observe that binary descriptors overall require less memory than floating point descriptors. The efficiency is measured with compiled code which makes it possible to just measure the time needed for feature description without the time needed in the interface between MATLAB and C/C++. With this approach all descriptors but LIOP are applied to SURF features which instead is applied to Hessian-Affine features as the code of the descriptors origin from two different libraries. By calculating the time per feature it is possible to compare the descriptors efficiency as the total time is dependent on the number of extracted features which differs for the two detectors.

As can be observed the two libraries are considered not to be equally optimized as LIOP, described in [7] to be less computationally efficient than SIFT, in this test outperforms both SIFT and SURF in the sense of run time.

In the table we can again conclude that binary methods typically are faster than floating point methods as they are an order of magnitude faster than SURF and two orders of magnitude faster than SIFT.

<table>
<thead>
<tr>
<th>View-point</th>
<th>SIFT</th>
<th>SURF</th>
<th>LIOP</th>
<th>BRIEF</th>
<th>ORB</th>
<th>BRISK</th>
<th>FREAK</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rotation</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td></td>
<td></td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>Scale</td>
<td>2</td>
<td>3</td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Blur</td>
<td>-</td>
<td>-</td>
<td>2</td>
<td>-</td>
<td>3</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>Noise</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Downsampling</td>
<td>1</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>6</td>
<td>1</td>
<td>17</td>
<td>0</td>
<td>0</td>
<td>10</td>
<td>0</td>
</tr>
<tr>
<td>Ranking</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 5.3: Summary of performance of descriptors. In order to give an idea on the comparative performance, the top performers are given a score from one to three resulting in the highest score for the best detector.

<table>
<thead>
<tr>
<th>Descriptor</th>
<th>Memory size (byte)</th>
<th>Run time [µs/feature]</th>
</tr>
</thead>
<tbody>
<tr>
<td>SIFT</td>
<td>128</td>
<td>5547</td>
</tr>
<tr>
<td>SURF</td>
<td>64</td>
<td>524</td>
</tr>
<tr>
<td>LIOP</td>
<td>144</td>
<td>143</td>
</tr>
<tr>
<td>BRIEF</td>
<td>32</td>
<td>96</td>
</tr>
<tr>
<td>ORB</td>
<td>32</td>
<td>38</td>
</tr>
<tr>
<td>BRISK</td>
<td>64</td>
<td>33</td>
</tr>
<tr>
<td>FREAK</td>
<td>64</td>
<td>39</td>
</tr>
</tbody>
</table>

Table 5.4: Memory requirement and efficiency of descriptors.
CHAPTER 5. EVALUATION RESULTS

5.3 Combinations

This section presents the results of combinations of detectors and descriptors. It is divided into floating point detectors with floating point descriptors and binary point detectors with binary point descriptors, with some exceptions. Exceptions are Harris-Affine combined with ORB and BRISK, and SURF combined with BRIEF and FREAK. The combination of Harris-Affine and ORB showed good performance in [40], while BRISK is combined with Harris-Affine in this evaluation due to its good performance in the previous section. SURF is combined with binary descriptors as it is known to outperform other floating point detectors in speed. Combinations which also are performed in the evaluation on visual images in [7].

The evaluation is performed under the same guidelines as the descriptor evaluation allowing an overlap error of 50% and presenting the result in precision-recall curves, see Section 5.2.1, and deformation-recall curves. Last, evaluation combinations are entitled by a concatenation of the detector and descriptor with hes and har being short for Hessian-Affine and Harris-Affine. In case of no concatenation e.g. orb both ORB detector and descriptor are applied.

View-Point

In this section we evaluate the effect of view-point changes on different combinations with the results in Figure 5.25 for the structured scene in Figure 4.1a, and in Figure 5.26, for the textured scene in Figure 4.1b.

By a brief study of the figures it is clear that the performance varies depending on the scene and the combination. In the structured scene among floating point combinations mserliop achieves the highest score in precision-recall but shows dependency on perspective changes by slow continuous increase as well as a large variation in the deformation-recall curve. Top performers in the scene are surf, hesliop and harliop. In the textured scene among floating point combinations the best results are obtained by mserliop followed by hesliop and harliop when both precision-recall and deformation-recall are considered. Noteworthy are differences as for surf which in the structured scene is the second top performer while in the textured scene dropped to the middle.

Among binary combinations the results are easier to interpret. Best performance is obtained by orbbrisk, for both scenes, with results comparable with the best performers in the floating point family of combinations. Consecutive in performance are orb and orbfreak indicating how combinations with the ORB detector outperform BRISK, FAST as well as SURF. Worst performance to view-point changes is obtained by fastbrisk.

Rotation

The results of the combinations due to rotations of images are illustrated in Figure 5.27 for the structured scene in Figure 4.1c and in Figure 5.28, for the textured
5.3. COMBINATIONS

Figure 5.25: Performance due to view-point changes in a structured scene. (a) & (c) Precision-Recall curve and (b) & (d) Recall relative increased view-point angle.

We observe that performances in rotated images are overall much higher than to view-point changes. The majority has high performance in both the structured and textured scene. Worst performance is obtained by the SURF detector in combination with SURF, FREAK and BRIEF descriptors, surf, surffreak and surfbrief. In Figure 5.27d is a clear example of how different detectors and descriptors perform in different set-ups. For example surfbrief has a poor performance as mentioned earlier while surffreak and surf, still indicating a dependence to rotation, has a greatly improved performance in comparison.

Overall best performance is achieved by hesliop followed by harliop and harsift among floating point combinations while for binary methods best performance is obtained by orbbrisk, orb and orbfreak.

Scale

The effect of scaled images on combinations of detectors and descriptors are shown by the results in Figure 5.29 and Figure 5.30 for the structured scene in Figure 4.1c and textured scene in Figure 4.1f respectively.
The combinations show a stable behaviour with similar performance in both scene types. Best performance is achieved by hesliop succeeded by surf, harliop and mserliop for floating point combinations. Studying the result of binary combinations in Figure 5.29c and Figure 5.30c shows that the relative order of combinations is the same for both scenes. Top performers within binary combinations are surfbrief, orbfreak and orbbrisk.

Blurr

In this section we present the results of combinations applied to images smoothed by a gaussian kernel in Figure 5.31 for the scene in Figure 4.1g and in Figure 5.32 for the scene in Figure 4.1h.

The effect of blur on combinations based on corner detectors as well as the blob detector MSER is evident in both scenes. All combinations where these are involved show poor performance in precision-recall curves. As visible in Figure 5.31d also known from Section 5.1 FAST’s performance decreases drastically with increasing smoothing level leading to zero detected interest points. The extensions to FAST, ORB and BRISK, show better performances due to the fact that they detect IPs.
5.3. COMBINATIONS

Figure 5.27: Performance due to rotation in a structured scene. (a) & (c) Precision-Recall curve and (b) & (d) Recall relative increased view-point angle.

at multiple scales.

Best performance is attained by surf outperforming other combinations in stability visible in both scenes with horizontal curves in precision-recall and deformation-recall curves. The top performer is followed by sift and hesliop. In the category of binary combinations we obtain surfbrief as the top performer with orbbrisk and orb to come after. Noteworthy is that for the high performance of orbbrisk, as well as the other combinations with ORB detector, in the structured scene only a very small number of IPs are extracted. For example at the standard deviation 6 in Figure 5.31d the recall reaches 100% where only 4 points are extracted compared to 24 by SURF and 35 by Hessian-Affine. Not to forget is the high precision shown by orbbrisk in the precision-recall curves by persistent horizontal curves.

Noise

We evaluate the performance of combinations applied to images with induced white Gaussian noise with the results presented in Figure 5.33, for the structured scene in Figure 4.1i, and in Figure 5.34, for the textured scene in Figure 4.1j.
CHAPTER 5. EVALUATION RESULTS

Figure 5.28: Performance due to rotation in a textured scene. (a) & (c) Precision-Recall curve and (b) & (d) Recall relative increased view-point angle.

Just as in the evaluation of descriptors the overall performance is high for various combinations when affected by noise. Best performance for floating point combinations are attained by hesliop and surf stagnating at about the same high level of recall in precision-recall curves for both scenes. Studying the deformation-recall curves we see how hesliop has a higher performance than surf in the textured scene. The combinations to come after are mserliop, harliop and harbrisk which all are about equally good when weighing the results in both scenes together. The overall best performance in case of induced noise is achieved by orbbrisk followed by orb and surfbrief showing better performance than the floating point family. The worst performance is given by brisk.

Downsampling

Last, we evaluate the effect on combinations caused by downsampled images and present the results in Figure 5.35 and in Figure 5.36 for the structured scene in Figure 4.1k and the textured scene in Figure 4.1l. As can be observed in Figure 5.36d the ORB detector fails in extracting IPs at
Figure 5.29: Performance due to scale in a structured scene. (a) & (c) Precision-Recall curve and (b) & (d) Recall relative increased view-point angle.

a downsampling factor of 8, as concluded in Section 5.1, resulting in zero recall. Taking both precision-recall curves and deformation-recall curves into account, the best performers in downsampled images are surf, hesliop, harbrisk and harliop. The combination of surf shows a good performance in the form of high stable precision-recall and deformation-recall curves. About equal performance is obtained by harbrisk and hesliop only swapping positions in performance order in the two scenes.

Among binary methods the best performance is obtained by surfbrief, with similar results to surf, and surfbreak to come after. If we only consider the downsampling factors of 2 and 4, as no correspondences are found by ORB detector combinations for factor 8, then orbbrisk would position in top three.

MSER does in Figure 5.35a reach a 100% recall, but if we observe its corresponding deformation-recall curve it is clear that it is only at this specific downsampling factor of 4 as no IPs are extracted at factor 8.

The worst performance is obtained by FAST which is a result from not detecting any correspondences, visualized by no data in the graphs. Low performance is also obtained by brisk and sift.
CHAPTER 5. EVALUATION RESULTS

Figure 5.30: Performance due to scale in a textured scene. (a) & (c) Precision-Recall curve and (b) & (d) Recall relative increased view-point angle.

5.3.1 Summary

In Table 5.5 the results are summarized from the evaluation of combinations. The table is, as the evaluation figures in this section, divided into two groups depending on if the detector in the combination is of floating or binary point. The exception is for SURF which occurs in both groups and is divided by its descriptor in an equivalent manner. Top three performers are given scores in both groups between 1-3 in the same approach as for detectors and descriptors, the higher the number the higher performance. Considering the best performers by weighing the number of occurrences in the table as well as the positions marked, the top three combinations of this evaluation are `hesliop`, `surf` and `orbbrisk`. As in other evaluations `surf` is a top performer, but again with the big drawback of being very sensitive to rotation. Hessian-Affine together with LIOP shows an overall high performance in this evaluation with blur being its most challenging deformation. Among the binary combinations best performance is achieved by ORB detector together with BRISK descriptor which overall outperforms other binary combinations and in many cases has comparable results with Hessian-Affine combined with LIOP.

In Table 5.6 the computational efficiency of the evaluated combinations is pre-
5.3. COMBINATIONS

![Graphs](a) & (c) Precision-Recall curve and (b) & (d) Recall relative increased view-point angle.

Figure 5.31: Performance due to blur in a structured scene. (a) & (c) Precision-Recall curve and (b) & (d) Recall relative increased view-point angle.

sent. Just as for detectors and descriptors the efficiency is measured with compiled code and calculated as the average of the the run time for the images in the dataset. As a few combinations origin from two libraries these run times are estimated by the corresponding run time for the detector and descriptor.

We observe that the highest efficiency is obtained by binary combinations with orbbrisk as the most efficient with an average total run time of 37 ms for an image in this test. Longest total run time is obtained by hesliop followed by sift. Noteworthy is the effect of the scene in Figure 4.1h on the average number of extracted points and run time for hesliop. In this textured scene the detector extracts at least four times as many features compared to in the other scenes which penalizes the total average of the extracted points and the run time. If not taking this scene into account the average run time is 1.71 s. For the floating point top performers the average run time is about 2.3 s for hesliop and 0.63 s for surf. This makes orbbrisk 62 times more efficient than hesliop and 17 times more efficient than surf.
Figure 5.32: Performance due to blur in a textured scene. (a) & (c) Precision-Recall curve and (b) & (d) Recall relative increased view-point angle.

<table>
<thead>
<tr>
<th>View-point</th>
<th>Rotation</th>
<th>Scale</th>
<th>Blur</th>
<th>Noise</th>
<th>Downsampling</th>
<th>Total</th>
<th>Ranking</th>
</tr>
</thead>
<tbody>
<tr>
<td>harorb</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>harbrisk</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>harsift</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>harliop</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>8</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>hesliop</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>3</td>
<td>13</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>sift</td>
<td></td>
<td></td>
<td>2</td>
<td></td>
<td></td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>surf</td>
<td></td>
<td></td>
<td>3</td>
<td>2</td>
<td>10</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>mserliop</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>orb</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>7</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>brisk</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>surfbrief</td>
<td></td>
<td></td>
<td>3</td>
<td>3</td>
<td>10</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>surffreak</td>
<td></td>
<td></td>
<td>2</td>
<td></td>
<td></td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>orbbrisk</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>(1)</td>
<td>12(13)</td>
<td>1</td>
</tr>
<tr>
<td>orbfreak</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td></td>
<td></td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>fastbrisk</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

Table 5.5: Summary of performance of combinations. In order to give an idea on the comparative performance, the top performers are given a score from one to three resulting in highest score for the best detector. Within parentheses the position if not downsampled by factor 8.
5.3. COMBINATIONS

Figure 5.33: Performance due to noise in a structured scene. (a) & (c) Precision-Recall curve and (b) & (d) Recall relative increased view-point angle.

<table>
<thead>
<tr>
<th>Combination</th>
<th>Run time [ms]</th>
<th># interest points</th>
<th>Run time [µs/feature]</th>
</tr>
</thead>
<tbody>
<tr>
<td>hesliop</td>
<td>2291</td>
<td>6334</td>
<td>361</td>
</tr>
<tr>
<td>harliop</td>
<td>1407</td>
<td>2655</td>
<td>539</td>
</tr>
<tr>
<td>harorb*</td>
<td>1160</td>
<td>2655</td>
<td>437</td>
</tr>
<tr>
<td>harbrisk*</td>
<td>1147</td>
<td>2655</td>
<td>432</td>
</tr>
<tr>
<td>mserliop*</td>
<td>612</td>
<td>281</td>
<td>2179</td>
</tr>
<tr>
<td>sift</td>
<td>1889</td>
<td>1372</td>
<td>1377</td>
</tr>
<tr>
<td>surf</td>
<td>627</td>
<td>894</td>
<td>702</td>
</tr>
<tr>
<td>orbbrisk</td>
<td>37</td>
<td>495</td>
<td>75</td>
</tr>
<tr>
<td>orbfreak</td>
<td>44</td>
<td>495</td>
<td>88</td>
</tr>
<tr>
<td>orb</td>
<td>49</td>
<td>495</td>
<td>99</td>
</tr>
<tr>
<td>brisk</td>
<td>73</td>
<td>407</td>
<td>180</td>
</tr>
<tr>
<td>briskfreak</td>
<td>67</td>
<td>407</td>
<td>166</td>
</tr>
<tr>
<td>fastbrisk</td>
<td>95</td>
<td>3179</td>
<td>30</td>
</tr>
<tr>
<td>surfbrief</td>
<td>272</td>
<td>894</td>
<td>304</td>
</tr>
<tr>
<td>surffreak</td>
<td>266</td>
<td>894</td>
<td>297</td>
</tr>
</tbody>
</table>

Table 5.6: Efficiency by the combinations. Combinations marked with * are estimated values based on the detector and descriptor.
Figure 5.34: Performance due to noise in a textured scene. (a) & (c) Precision-Recall curve and (b) & (d) Recall relative increased view-point angle.
Figure 5.35: Performance due to downsampling in a structured scene. (a) & (c) Precision-Recall curve and (b) & (d) Recall relative increased view-point angle. As can be observed in (d) there is no data for combinations with the ORB detector for downsampling factor 8 which is due to no detected correspondences.
Figure 5.36: Performance due to downsampling in a textured scene. (a) & (c) Precision-Recall curve and (b) & (d) Recall relative increased view-point angle. As can be observed in (b) & (d) there are no data for the combinations with ORB and MSER detectors for downsampling factor 8 which is due to no detected correspondences.
Chapter 6

Application Analogies

In this chapter we discuss the evaluation results in Chapter 5 in connection to the applications, image stitching and image retrieval.

Image Stitching

In image stitching the goal is to combine two or more images to one unified larger image. This is commonly realized by an invariant IP detector and descriptor which extract and match features between images.

When IPs and descriptors have been extracted one approach is to use RANSAC which is implemented to randomly select matches used to estimate a homography. The estimated homography is the best consensus with the used matches.

The obtained homography is then used to transform the images to the final stitched image. Worth to mention is that the steps explained are only main features of an image stitching process. In a real world application more steps are included to increase the quality.

When performing image stitching desired properties of a detector and descriptor combination are high distinctiveness of extracted features, large enough quantity and a distributed set of features such that not all are located in one part of the image. Distribution of features is also important since images to be stitched often visualize the same content close to the borders. Another factor, in the choice of detector and descriptor, is that depending on the situation the choice of method might vary. If the image stitching is to be performed in an image editor software running on a standard computer the execution time might not be the most crucial factor but high accuracy. In this situation Hessian-Affine with LIOP is the suggested method to use as it shows best performance in the evaluation and has a large number of correspondences. If the stitching is to be performed on a hand-held device, as a mobile phone in combination with a FLIR One IR camera, the choice of ORB and BRISK is recommended. It is the superior binary combination with high speed and has among the larger number of correspondences although restricted to extract at most 500 features. Also it has a low memory requirement of 64 bytes per feature.
compared to 144 bytes used by LIOP.

**Image Retrieval**

In image retrieval the goal is to find which images contain the same scene or object as a reference image or a keyword. One approach is to have a database storing descriptors from various scenes and objects. When an image is to be matched, descriptors are extracted and further matched with the descriptors in the database. Out from the matched descriptors the retrieved images can e.g. be chosen as the ones with most corresponding descriptors.

Similar to image stitching, combinations used in image retrieval are desired to have high distinctiveness of extracted features and large enough quantity of interest points. The quantity of interest points needs to be high enough to extract sufficient information such that it is possible to separate a correct match from a false match.

Since the appearances of scenes and objects may look very different in images, the usage of an invariant detector and descriptor is of interest. Just as in image stitching recommendations are to use Hessian-Affine with LIOP or ORB together with BRISK. If the implementation is for academic research a good alternative is SURF, which for commercial usage needs a license as it is patented. It has proven to be a top performer together with above mentioned combinations and has an efficiency between the two.
Chapter 7

Comparisons to Results in Earlier Work

In this chapter we discuss the results in this work compared to results in related work, both for the visual domain and the LWIR domain.

Detectors

Comparing the results of detectors in the LWIR band obtained in this work to the results from the thorough work in the visual band in [6], a clear difference is the performance by MSER. In the visual spectral band it did in many cases achieve the highest score followed by Hessian-Affine while in the LWIR spectral band it overall showed worse performance than many other detectors in this evaluation. A reason might be that MSER is very sensitive to blur which more commonly occurs in IR images than visual images visualized by less distinct borders.

Hessian-Affine and Harris-Affine which are popular to use in combination with descriptors in the visual spectrum show good performance in IR images as well, especially Hessian-Affine which shows best overall performance.

More efficient detectors as ORB and SURF, not invented at the time of above discussed evaluation, are two commonly used detectors in visual images. SURF which is very popular due to its balance of high accuracy and efficiency in visual images also shows high performance in LWIR images. In the evaluation presented in [41], the best ratio between robustness and efficiency was obtained by BRISK followed by ORB while in this work we found ORB and SURF to have the highest ratios. We can also conclude that BRISK’s performance in IR images are different from the visual images as it has among the worst performances in this work.

Descriptors

In [7,35] it is concluded that LIOP outperforms state-of-the-art descriptors as SIFT and SURF, a result we also have received for IR images in this work. It is also presented how the binary descriptors BRIEF, ORB and BRISK have comparable results to SURF and SIFT. In the LWIR spectral band BRIEF and ORB do in
some categories have comparable results to these two descriptors while BRISK consistently performs better.

A result which points at better performance in IR images than in visual images by BRISK relative to most descriptors.

Last we can conclude that the popular SIFT descriptor still is one of the top performers in IR images.

**Combinations**

In the evaluation of binary methods for visual images in [42], it is obvious how the combination of detector and descriptor might affect the performance. For example in tests of descriptors with their default detectors, BRISK and FREAK perform much worse than when combined with the ORB detector. Best overall performance was obtained by ORB detector in combination with FREAK or BRISK descriptors and ORB combined with FREAK is the suggested combination to use. In this work we can confirm that BRISK with its default detector performs much worse, in most categories even the worst, compared to when in combination with the ORB detector while the combination of ORB and FREAK has lower performance in the LWIR spectral band. With the high performance of the combination of ORB and BRISK we can conclude that the choice of combination has large effect on the performance both in visual images and IR images.

Another similarity is the high performance by Hessian-Affine with LIOP in [23] and in this work as well as by the combination of SURF which shows high performance in both spectral bands.

**IR images**

The most related work in the LW spectral band [34] shows both similarities and differences to the results in this work. Best performance against blur is in both evaluations obtained by SURF. For rotation best performance is achieved by SIFT in the compared evaluation while LIOP, not included in the earlier evaluation, shows highest robustness to the deformation in this work. In this evaluation SIFT struggles with rotation changes between $90-180^\circ$ when combined with its default detector, DoG, while it has a more stable behaviour when combined with Hessian-Affine and Harris-Affine. The same tendency, to suit better with the mentioned detectors, is found when we compare the performance of SIFT against scale changes, where it had the highest score in [34].

Among binary combinations [34] presents a low performance for ORB and BRISK with their default detectors. In this work the low performance of the combination of BRISK is confirmed while the combination of ORB is a top performer among binary methods. A difference shown to be important between these two evaluations is the comparisons of different detectors and descriptors in combinations performed in this work which have led to the conclusion of a good match of ORB detector and BRISK descriptor.
Chapter 8

Conclusions

In this chapter we present our conclusions based on the performed evaluations.

The objective of this work was to evaluate the performance of the state-of-the-art detectors and descriptors developed for visual images on IR images. Detectors and descriptors performances were tested against the image condition changes: view-point, rotation, scale, blur, noise and downsampling. The evaluation results underlie the conclusions and recommendations of which methods should be used in applications as image stitching and image retrieval on IR images.

Detectors

From the evaluation we can conclude that for IR images the most suitable detectors are Hessian-Affine, SURF and ORB. The detector with overall best performance in repeatability is Hessian-Affine. If an application is to circumvent patented detectors Hessian-Affine and ORB are suggested. Further, for real time systems or devices with less computational power the binary detector ORB is recommended.

Descriptors

We can conclude that best performance among descriptors are achieved by LIOP, BRISK and SIFT which outperform the other descriptors. Similar to evaluations performed on visual images, as in [7], LIOP outperforms the state-of-the-art descriptor SIFT in all categories on IR images in both precision and recall. Among binary descriptors BRISK shows an overall better performance than SIFT and outperforms the other descriptors. The benefit of BRISK compared to LIOP and SIFT is its high computational efficiency and low memory requirement.

Combinations

From the high number of combinations tested we find that best performance is achieved by Hessian-Affine combined with LIOP. Comparable to its result is the
binary combination of ORB detector with BRISK descriptor which outperforms other alternatives. These two combinations are also the only ones to perform well in all categories. An alternative to these combinations is the combination of SURF which has high scores but also a great weakness to rotation changes.

Therefore, in applications where the highest repeatability and accuracy is of interest Hessian-Affine together with LIOP is suggested. On the other hand, if speed, less computational power and memory usage are concerned, the ORB detector with BRISK descriptor is recommended.

**Image Stitching and Image Retrieval**

From the conclusions in the previous section together with the discussion in Chapter 6 we recommend Hessian-Affine combined with the LIOP descriptor for both image retrieval and image stitching. As binary alternative, with comparable results, we recommend the combination of ORB detector and BRISK descriptor.
Bibliography


BIBLIOGRAPHY


[33] L. Ferraz and X. Binefa, “A Scale Invariant Interest Point Detector for Discriminative Blob Detection,” in Pattern Recognition and Image Analysis, vol. 5524


Appendix A

List of Considered Detectors and Descriptors

In this appendix are considered detectors and descriptors from the literature study presented. They are displayed in the tables below with brief information of year published, authors, invariances etc.
### Floating Point Methods

<table>
<thead>
<tr>
<th>Detector:</th>
<th>Type</th>
<th>Shape</th>
<th>Invariance</th>
<th>Robustness</th>
<th>Other</th>
<th>Available</th>
<th>Year</th>
<th>Author/Authors</th>
<th>License</th>
<th>FPGA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Harris-Laplace</td>
<td>Corner</td>
<td>Circle</td>
<td>Scale, rotation and camera noise.</td>
<td>Illumination</td>
<td>Used for Laplacian-based scale selection.</td>
<td>OpenCV in C++, use OpenCV for Matlab.</td>
<td>2011</td>
<td>Michael J. Black and Tonio Kröger</td>
<td>BSD</td>
<td>Matlab, OpenCV</td>
</tr>
<tr>
<td>Harris-Laplace</td>
<td>Blob</td>
<td>Circle</td>
<td>Scale, rotation and camera noise.</td>
<td>Returns slightly more non-interested regions than Harris-Laplace.</td>
<td>Illumination</td>
<td>Used for Laplacian-based scale selection.</td>
<td>OpenCV via Matlab.</td>
<td>Michael J. Black and Tonio Kröger</td>
<td>BSD</td>
<td>Matlab, OpenCV</td>
</tr>
<tr>
<td>Harris-Affine</td>
<td>Corner</td>
<td>Ellipses</td>
<td>Affine and higher order affine shape estimation.</td>
<td>Illumination, blur</td>
<td>Performed as good as Harris-Affine-2 in 2001. Good for precisely localized interest points.</td>
<td>OpenCV in C++, also on Github by Jaechul Kim and Kristen Grauman</td>
<td>2003</td>
<td>Michael J. Black and Tonio Kröger</td>
<td>BSD</td>
<td>Matlab, OpenCV</td>
</tr>
<tr>
<td>Harris-Affine</td>
<td>Blob</td>
<td>Ellipses</td>
<td>Affine and higher order affine shape estimation.</td>
<td>Illumination, blur</td>
<td>Performed second best in 2005 after MSER. Good for precisely localized interest points.</td>
<td>OpenCV via Matlab.</td>
<td>Michael J. Black and Tonio Kröger</td>
<td>BSD</td>
<td>Matlab, OpenCV</td>
<td></td>
</tr>
<tr>
<td>EBR</td>
<td>Edge Based Region, Blob, Boundary based</td>
<td>Parallelogram</td>
<td>Scale, rotation, affine</td>
<td>Illumination, blur</td>
<td>Start from Harris corner.</td>
<td>As binaries</td>
<td>1999</td>
<td>Toivonen and Van Gool</td>
<td>BSD</td>
<td>Matlab, OpenCV</td>
</tr>
<tr>
<td>R-SIFT</td>
<td>Blob, Boundary based</td>
<td>Arbitrary</td>
<td>Scale, rotation, affine</td>
<td>Illumination, blur</td>
<td>Keypoint detector and image matching according to authors.</td>
<td>Matlab, given by the authors</td>
<td>2011</td>
<td>Jaechul Kim and Kristen Grauman</td>
<td>BSD</td>
<td>Matlab, OpenCV</td>
</tr>
<tr>
<td>MEBR</td>
<td>Blob, Internally based</td>
<td>N/A</td>
<td>Not Found</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>SIFT</td>
<td>Corner, Spatial features</td>
<td>Arbitrary</td>
<td>Scale, rotation</td>
<td>Illumination</td>
<td>Waikey keypoint detector and image matching. Checks the entropy in pdf's for intensity to find interest points.</td>
<td>Matlab, C++</td>
<td>2005</td>
<td>Wolfgang Förstner, Tomaž Sonar, and Falko Schindler</td>
<td>BSD</td>
<td>OpenCV</td>
</tr>
<tr>
<td>Harris</td>
<td>Corner, Point</td>
<td>Rotation, illumination, noise</td>
<td>view-point</td>
<td>Common</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>KLT, Kanade-Lucas-Tomasi</td>
<td>Corner, Point</td>
<td>Rotation, illumination, noise</td>
<td>view-point</td>
<td>Common</td>
<td>OpenCV, Matlab</td>
<td>1988</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>SUSAN</td>
<td>Corner, Template based</td>
<td>N/A</td>
<td>More efficient than Harris corner detector but more sensitive to noise.</td>
<td>N/A</td>
<td>Matlab</td>
<td>1998</td>
<td>Andrew Zisserman</td>
<td>BSD</td>
<td>Matlab, OpenCV</td>
<td></td>
</tr>
<tr>
<td>FAST</td>
<td>Corner, Template based</td>
<td>Circle</td>
<td>Rotation</td>
<td>Builds on SUSAN. Fast about a three times faster than the SFM. Scale to detect dominant orientation.</td>
<td>Matlab, Python, C</td>
<td>2003</td>
<td>Edward Rosten and Tomaž Sonar</td>
<td>BSD</td>
<td>OpenCV, Matlab, Python, C</td>
<td></td>
</tr>
<tr>
<td>FAST-ER</td>
<td>Corner, Template based</td>
<td>Circle</td>
<td>Rotation</td>
<td>Not as efficient as FAST-9 but has better repeatability.</td>
<td>Matlab</td>
<td>2004</td>
<td>Edward Rosten, David Porter and Tony O. Brox</td>
<td>BSD</td>
<td>OpenCV, Matlab, Python, C</td>
<td></td>
</tr>
<tr>
<td>FAST+ (FAST-9)</td>
<td>Corner, Template based</td>
<td>Circle</td>
<td>Rotation</td>
<td>High repeatability. Quite commonly used.</td>
<td>Matlab, Python, C</td>
<td>2005</td>
<td>Edward Rosten, David Porter and Tony O. Brox</td>
<td>BSD</td>
<td>OpenCV, Matlab, Python, C</td>
<td></td>
</tr>
<tr>
<td>DoG</td>
<td>Blob, PDE based</td>
<td>Blob, circular</td>
<td>Scale, rotation</td>
<td>Approximation of the second derivatives of LBG.</td>
<td>Matlab</td>
<td>1999</td>
<td>H. Bergström, H. Lindeborg</td>
<td>BSD</td>
<td>Matlab, Python, C</td>
<td></td>
</tr>
<tr>
<td>DoG</td>
<td>Blob, PDE based</td>
<td>Blob, circular</td>
<td>Scale, rotation</td>
<td>Approximation of the second derivatives of LBG.</td>
<td>Matlab</td>
<td>1999</td>
<td>H. Bergström, H. Lindeborg</td>
<td>BSD</td>
<td>Matlab, Python, C</td>
<td></td>
</tr>
<tr>
<td>DoG</td>
<td>Blob, PDE based</td>
<td>Blob, circular</td>
<td>Scale, rotation</td>
<td>Approximation of the second derivatives of LBG.</td>
<td>Matlab</td>
<td>1999</td>
<td>H. Bergström, H. Lindeborg</td>
<td>BSD</td>
<td>Matlab, Python, C</td>
<td></td>
</tr>
<tr>
<td>DoG</td>
<td>Blob, PDE based</td>
<td>Blob, circular</td>
<td>Scale, rotation</td>
<td>Approximation of the second derivatives of LBG.</td>
<td>Matlab</td>
<td>1999</td>
<td>H. Bergström, H. Lindeborg</td>
<td>BSD</td>
<td>Matlab, Python, C</td>
<td></td>
</tr>
<tr>
<td>DoG</td>
<td>Blob, PDE based</td>
<td>Blob, circular</td>
<td>Scale, rotation</td>
<td>Approximation of the second derivatives of LBG.</td>
<td>Matlab</td>
<td>1999</td>
<td>H. Bergström, H. Lindeborg</td>
<td>BSD</td>
<td>Matlab, Python, C</td>
<td></td>
</tr>
<tr>
<td>DoG</td>
<td>Blob, PDE based</td>
<td>Blob, circular</td>
<td>Scale, rotation</td>
<td>Approximation of the second derivatives of LBG.</td>
<td>Matlab</td>
<td>1999</td>
<td>H. Bergström, H. Lindeborg</td>
<td>BSD</td>
<td>Matlab, Python, C</td>
<td></td>
</tr>
<tr>
<td>DoG</td>
<td>Blob, PDE based</td>
<td>Blob, circular</td>
<td>Scale, rotation</td>
<td>Approximation of the second derivatives of LBG.</td>
<td>Matlab</td>
<td>1999</td>
<td>H. Bergström, H. Lindeborg</td>
<td>BSD</td>
<td>Matlab, Python, C</td>
<td></td>
</tr>
<tr>
<td>DoG</td>
<td>Blob, PDE based</td>
<td>Blob, circular</td>
<td>Scale, rotation</td>
<td>Approximation of the second derivatives of LBG.</td>
<td>Matlab</td>
<td>1999</td>
<td>H. Bergström, H. Lindeborg</td>
<td>BSD</td>
<td>Matlab, Python, C</td>
<td></td>
</tr>
</tbody>
</table>

**Note:** The above table lists various feature detectors and their capabilities, along with their respective authors and licenses. The table is structured to provide a clear comparison of features such as invariance, robustness, and availability across different detectors and their respective years of publication. Each detector is categorized under specific types and shapes, with varying degrees of invariance to changes such as scale, rotation, and affine shape. The robustness factor varies, with some detectors being more sensitive to noise or blur. The table also highlights the availability of the detectors either as binaries or as part of larger packages such as OpenCV and Matlab. The license information indicates whether the software is open-source (BSD) or proprietary. The FPGA column indicates whether the detector is suitable for hardware implementation. The information is sourced from various academic and technical resources, some of which are referenced in the footnotes.
Blob (Corner, PDE based)

Blob Circle

Abrupt and strong variations of images. Uses weighted rank order LoG filter. A non-biased filter. Shows better result than Hessen-Affine, Harris-Affine and MSER in the paper. His implementation found though it is implemented in Matlab according to author. Quite straightforward to how to implement it.

Author/Authors: Zhenwei Mei, Xudong Jiang
Year: 2012
License: Patented

ASIFT Blob Circle

Scale, rotation, affine

Invariance: Most robust descriptor. OpenCV, VLFeat

OpenCV, VLFeat

ASIFT

Author/Authors: Guoshuo Yu, Jean-Michel Morel
Year: 2006
License: Patented

SURF Blob Circle

Scale, rotation, illumination

Approximated version of SIFT for increased efficiency.

OpenCV, Matlab

OpenCV, Matlab

SURF

Author/Authors: Herbert Bay, Tony Tuytelaars, and Luc Van Gool
Year: 2006
License: Patented

KAZE Corner-like or Blob-like depending on the conductivity function in diffusion step. (Blob, Template based)

Rectangular

Scale, rotation, illumination.

Non-linear diffusion, non-linear scale space. Uses MSURF adopted for non-linear scale space.

OpenCV via Github, or available in OpenCV library from version 3.0

Pablo Fernandez Alcaraz, Adrian Barreto, and Andrew J. Davison

Year: 2010
License: BSD

DART Blob, PDE based

Circle

View-point, illumination.

Scale, rotation, affine?

Its descriptor is influenced by DAISY. Better precision and recall than SIFT & SURF and better Approximation of the Grad by piece-wise triangular filters.

David Martin, Arno, Felix Tomasz and Roger Grimes

Year: 2011
License: Patented

MDPS Detector: Type

Shape

Invariance

Robustness

Other

Available

Year

Authors

License

FPGA

SIFT Blob

Circle

Scale, rotation

Invariance

Most robust descriptor.

OpenCV, VLFeat

OpenCV, VLFeat

SIFT

Author/Authors: Lowe
Year: 1999
License: Patented

ASIFT Blob

Circle

Scale, rotation, affine

Invariance: Better than SIFT under affine transformations but performs like SIFT under the other situations.

Source code given by authors on webpage.

Qiang Wu, Yang and Zhanyi Wang, Bin Fan, Fuchai Wu

Year: 2011
License: BSD

SURF Blob

Circle

Scale

Invariance

Approximated version of SIFT.

OpenCV, Matlab

OpenCV, Matlab

SURF

Author/Authors: Herbert Bay, Tony Tuytelaars, and Luc Van Gool
Year: 2006
License: Patented

KAZE Corner-like or Blob-like depending on the conductivity function in diffusion step. (Blob, Template based)

Rectangular

Scale, rotation, illumination.

Non-linear diffusion, non-linear scale space. Uses MSURF adopted for non-linear scale space.

OpenCV via Github, or available in OpenCV library from version 3.0

Pablo Fernandez Alcaraz, Adrian Barreto, and Andrew J. Davison

Year: 2010
License: BSD

DART Blob, PDE based

Circle

View-point, illumination.

Scale, rotation, affine?

Its descriptor is influenced by DAISY. Better precision and recall than SIFT & SURF and better Approximation of the Grad by piece-wise triangular filters.

David Martin, Arno, Felix Tomasz and Roger Grimes

Year: 2011
License: Patented

License
<table>
<thead>
<tr>
<th>Detector</th>
<th>Type</th>
<th>Shape</th>
<th>Invariance</th>
<th>Robustness</th>
<th>Other</th>
<th>Available</th>
<th>Year</th>
<th>Author/Authors</th>
<th>License</th>
<th>FPGA</th>
</tr>
</thead>
<tbody>
<tr>
<td>BRIEF</td>
<td>Intensity</td>
<td>Scale or rotation depending on the version.</td>
<td>Rotation, blur and perspective distortion.</td>
<td>Use a fast detector such as CenSurE. Performs well to viewpoint changes in an evaluation, better than ORB and BRISK. Works best if dataset is similar in scale and orientation.</td>
<td>Matlab</td>
<td>2010</td>
<td>Michael Calonder, Vincent Lepetit, Tomas proakis, Christoph Stroha, and Pascal Fua</td>
<td>GPL v2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>OSRI</td>
<td>Blob, Template based</td>
<td>Uses Hessian-Affine as detector</td>
<td></td>
<td></td>
<td></td>
<td>2014</td>
<td>Xianwei Xu, Lu Tan, Jianliang Feng, Je Zhou</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FREAK</td>
<td>Blob, Template based</td>
<td>Uses the same detector as BRISK. Authors claim that it is more robust than SURF. Quite well cited.</td>
<td></td>
<td></td>
<td>OpenCV</td>
<td>2012</td>
<td>Alexandre Kalcheff, Raphael Ortiz, Pierre Vandergheynst</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LDB</td>
<td>Corner</td>
<td>Circle</td>
<td>Rotation</td>
<td>Resistant to noise. Performs quite well in affine transformations.</td>
<td>Uses OFAST as detector. Some as ORB. Achieves greater robustness and discriminative ability than e.g. ORB and BRISK. Code given by author as libLDB, a C++ library.</td>
<td>Code given by author as libLDB, a C++ library.</td>
<td>2014</td>
<td>Xin Yang and Kwang-Ting (Tim) Cheng</td>
<td>GNU General Public License version 3.</td>
<td></td>
</tr>
<tr>
<td>LBP</td>
<td>Blob, Intensity based</td>
<td>Based on LBP. Multi scale LBP.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RLBP</td>
<td>Rotation</td>
<td>Based on LBP.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAND (BRISK)</td>
<td>Blob, Intensity based</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Other Detectors

<table>
<thead>
<tr>
<th>Detector</th>
<th>Type</th>
<th>Shape</th>
<th>Invariance</th>
<th>Robustness</th>
<th>Other</th>
<th>Available</th>
<th>Year</th>
<th>Author/Authors</th>
<th>License</th>
<th>FPGA</th>
</tr>
</thead>
<tbody>
<tr>
<td>GFTTDetector</td>
<td>Corner</td>
<td>Point</td>
<td>Also known as Shi-Tomasi corner detector.</td>
<td>OpenCV</td>
<td>~1990</td>
<td>Tomasi, Shi</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MFD</td>
<td>Blob, Intensity based</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MCBR</td>
<td>Blob, Intensity based</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Saliency detection</td>
<td>Blob, Intensity based</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>