An in-depth study of local image descriptors and their performance

Anders Boesen Lindbo Larsen – abll@diku.dk

February 10, 2012

Abstract

In this project, we study the performance of local image descriptors using a recently released dataset [2] that offers unique evaluation possibilities compared to previous datasets. Our investigation is twofold in that we evaluate current state-of-the-art descriptors and develop our own descriptors to explore different designs and their performance implications. The descriptors we develop are based on the locally orderless image representation and on higher-order differential structure. We show that we by the use of higher-order image structure are able to reduce the descriptor dimensionality and partly do away with the multi-local description, that has previously been considered crucial for achieving good performance. We test the descriptors in a variety of scenarios (namely, large changes in scale, viewing angle and lighting) and find that the choice of descriptor should be based with the image data and the end application in mind.

Contents

1 Introduction

For more than a decade of computer vision research, a notable innovation has been the development of local feature detectors and descriptors. Local features are important building blocks of many visual systems as they provide an effective way to represent an image. They are used in computer vision fields such as object tracking [55], object detection [34], object recognition [54], image retrieval [9], image mosaicing [40] and 3D reconstruction [3].

The handling of local image features depends on the application. When used for image registration, local image features are extracted from a set of detected *interest points* in the images to be compared. From the similarity of the feature descriptors between the images, it is possible to find matching points and calculating the coordinate transformation between the images such that corresponding feature points overlap. The process of determining feature descriptor correspondences between images is known as *feature matching*. When used for visual recognition, local feature descriptors are typically sampled densely from an image and vector quantized to *visual words* describing the image contents as a *bag-of-features* [35].

Local feature descriptors are effective mainly for three reasons. Firstly, the individual descriptors are robust to changes in illumination, viewpoint and image noise. Secondly, the set of local features representing an image is robust to occlusions and large scale perturbations because of its spatially orderless nature. Finally, local features offer a reduction in complexity as their representation of an image typically has much lower dimensionality than the original image.

In this project the goal is to explore a *scale-space* [18,24,53] based approach to constructing local feature descriptors using the *locally orderless image* (LOI) representation [20,47]. Most successful descriptors are formulated in an ad-hoc manner without rigorous justification for some of the design choices. This project tries to formulate descriptors in a more mathematically sound framework and to explore different descriptor designs.

Furthermore, this project seeks an understanding of how and why current feature descriptors work. Using the recently released DTU Robot image dataset [2] it is possible to evaluate descriptor performance in more detail than with previous datasets. The DTU dataset is more comprehensive in size and the images are taken in a more controlled and realistic environment. It offers support for perturbation scenarios where the viewing angle, the scale and the lighting changes.

Limitations

A limitation of this project is that we choose not to consider the ability of a descriptor to be rotation invariant, i.e. robust to image rotations around the optical axis of the camera. Camera rotations are not directly supported by the

DTU dataset that we use to evaluate descriptor performance. Furthermore, it should be noted the rotation invariance is not always a desirable feature since it decreases the discriminative power of a local image descriptor significantly and is therefore disabled in some visual systems.

Another limitation is that the functioning of local interest point detectors will not be considered in this project. While the detector has a significant influence on the performance of the local features [7], the focus of this report is to investigate descriptor construction and behavior.

Only gray-scale images will be considered in this project. This is a common limitation of local image descriptors since the geometric structure is mainly described by image intensity channel. Furthermore, capturing color features invariant to changes in illumination is hard to achieve in practice [37]. For an overview of color descriptors, see [46].

Finally, local image description is considered only in the context of feature matching. This is by far the most popular evaluation criterion of local feature descriptors used in the literature (except for [54]). However, it is not certain that this evaluation criterion reflects the performance of the descriptors when used in another context such as bag of features for object recognition. One could argue that in the general case, feature matching is a more representative performance measure. Its performance reflects the discriminative ability of a descriptor more directly compared to object recognition where the descriptor performance depends on the ability to be clustered into representative visual words.

Report structure

This report is organized as follows. In Section 2, an overview of the local feature description process is given including a review of popular descriptors in the literature. In Section 3 and 4, the theory behind image description is presented using the scale-space framework and the locally orderless image representation. In Section 5, we present our descriptor proposals based on the theory in the previous two chapters. We will also discuss the configuration of these descriptors and what elements of the descriptor design we wish to investigate further in our experiments. Section 6 contains a review of the dataset we are working with and the evaluation criteria used to quantify descriptor performance. In Section 7, we perform an evaluation of our descriptor proposals and investigate the performance implications of the different design choices. We also perform a comparative evaluation with other state-of-the-art descriptors. Finally, the results are discussed in Section 8.

2 Overview of local image description

Local feature detection and description was made popular by D. Lowe in his work on the SIFT algorithm [25]. Since then, a wealth of different methods for both detection and description has appeared. In a vision system, the typical handling of local features is composed of three steps.

1. Detect local interest points in the input images.
2. Describe the local image patches specified by the interest points.
3. Match descriptors between images to find correspondences according to some *matching strategy*.

In this section we present local image features with a focus on step 2 and 3 and review relevant literature.

2.1 Interest points

Interest points specify the local image regions to be described. The points are found by detecting image structure such as corners, blobs or extremal regions. The detector is supposed to output matching regions from images that differ by some perturbation, e.g. by a change in illumination. See [2, 45] for a survey on local interest point detectors.

Depending on the detector type, interest points are specified either as a circle with radius r or as an ellipsis given by the quadratic equation $ax^2 + bxy + cy^2 = 1$. The ellipsis is used by detectors that try to estimate affine transformations of image regions (assuming planar surfaces) [31].

For each interest point in an image, the image region is *canonized* by extracting the region and resampling it to an image patch of a fixed size. Elliptical interest regions are additionally transformed affinely to a circular shape. In this project, we use the code from [30] and resample image patches using bilinear interpolation to a patch size of 64×64 pixels. See Figure 2.1 for an illustration of interest points regions and their canonized image patches.

Notice that this canonization is not necessary to describe an interest point as the descriptors might use the image data directly from the image. However, the canonization step is practical since it establishes a standard from which to perform the description. Therefore, canonization is employed in this project.

2.2 Descriptor construction pipeline

The local feature descriptor algorithm can often be decomposed into the following stages (though not all stages are applicable to all descriptor types). This approach is inspired by the feature descriptor organization in [51].

1. Image feature extraction. Given the image patch, compute features capturing the geometric structure of the image.

Figure 2.1: Two examples of affine interest point regions and their canonized image patches.

2. Histogram binning. Represent the statistics of the feature response from the previous step as a histogram. The histogram has a fixed size and the bin contribution of the feature response is determined in this step.
3. Spatial pooling. Perform a multi-local description of the image patch by selecting several points/cells in the patch and describe them each separately.
4. Normalization. Concatenate the output of the previous step (usually histograms) into a vector and normalize it according to some manner.

For the rest of this report, local feature descriptors will be explained from this framework.

2.3 Image perturbations

The strength of local image features is their robustness towards nuisance factors. In this section we briefly review the typical image perturbations in the feature matching problem. These arise from the following:

Camera movement The position of the camera relative to the subject influence the appearance of the subject in terms of scale and perspective. It may also lead to occlusions.

Camera rotation Rotations around the optical axis of the camera affect the subject appearance severely. Changing the view direction of the camera leads to a perspective distortion of the subject.

Illumination The camera exposure, the light source positioning and the illumination brightness and color lead to changes in image contrast and overall appearance of the subject.

Noise Noise usually stems from image compression or over-sensitivity of a camera's CCD sensor.

Note that some of the nuisance factors cannot be handled in a local feature system (e.g. occlusions covering entire image regions). Note also that in a local feature system, many of the above nuisance factors are handled before the description step. For example, the detector estimates the scale of the interest points and thereby establishes invariance towards changes in distance between the camera and the subject. When we reach the description step, the influencing factors above manifest themselves as the following image perturbations.

Scale As already mentioned, the detection of the interest point scale and the canonization of the image region should ensure that the scale is the same even though the distance between the camera and the subject changes. However, the level of detail in the image region varies as the resolution of the image region changes.

Perspective Perspective changes lead to deformations that are hard to model. Some detectors try to ameliorate on this deformation by estimating the affine transformations of an image region under the assumption that it covers a planar surface.

Rotation Rotations usually requires the descriptor to detect a *dominating orientation* and rotate the image patch accordingly. However, this approach is problematic when the dominating orientation cannot be detected uniquely (i.e. with a large confidence) for an image region. The SIFT algorithm accommodates this problem by creating multiple descriptors for such an image region; one descriptor per dominant direction.

Translation The position of an interest point image region might not be centered at exactly the same location due to imprecision in the detector. Thus, we want some robustness towards spatial shifts of the image patch.

Illumination The variations in pixel intensity I between images is normally assumed to be modeled by $aI + b$. That is, we want to be invariant to changes in image contrast and brightness offset. Notice that this model does not handle nonlinear changes in illumination caused by changes to the light source position, e.g. shadows.

Noise Noise distortions of image patches are handled in the descriptor design whenever a signal is smoothed. The assumption on the noise model is specified by the smoothing function.

As we shall see in the discussion of local image descriptors, some of the design choices in a descriptor can be explained from these perturbations.

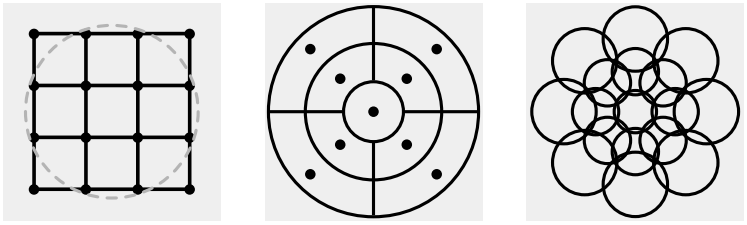
2.4 Popular descriptors

Among the wide variety of local image descriptors, we briefly survey a selection of the more popular ones. These descriptors will be used in our comparative performance evaluation (except the DAISY descriptor). In the following, we will assume that we are given canonized image patches.

SIFT

The *scale-invariant feature transform* (SIFT) descriptor is by far the most popular descriptor because of its good performance [7, 30] and its broad availability through different software implementations [49].

SIFT captures local image structure in a square 4×4 spatial grid of gradient orientation histograms with 8 bins each. The bin contributions of the gradient orientations are weighted in two turns. Firstly, each gradient is weighted



SIFT GLOH DAISY

Figure 2.2: Different descriptor layouts. The SIFT descriptor extracts 4×4 histograms in a square grid. The histograms are weighted using a Gaussian window illustrated by the dashed line. The GLOH descriptor uses a polar grid with 17 histograms (in the illustration above, we have shown a simpler grid with only 9 histograms). The DAISY descriptor is similar to the GLOH descriptor in that it uses a polar grid to distribute its gradient orientation histograms.

according to its L^2 norm such that large gradients have bigger influence than smaller gradients. Secondly, all gradients are weighted according to a Gaussian window centered in the image patch. The purpose of the Gaussian window is to lessen the influence of peripheral gradients that tend to be more perturbed. The contribution of a gradient to each of the 16 histograms in the grid is bilinearly weighted according to its distance from each histogram center. Moreover, the contribution to each bin in a histogram is again weighted bilinearly according to its distance from the bin center.

With a total of 16 histograms of 8 bins each, the SIFT descriptor achieves a dimensionality of 128. The final step in the SIFT description algorithm is to perform a normalization of the 128 dimensional vector to obtain illumination invariance. An illustration of the SIFT descriptor is shown in Figure 2.2.

The reason why down-sampling to 4×4 gradient orientation histograms works is because it offers a good tradeoff between robustness and distinctiveness. The robustness comes from the down-sampling and the bilinear weightings which results in a more graceful handling of perturbations such as translation and perspective transformations. The distinctiveness comes from the histograms being effective at representing the image structure when sampled side by side.

PCA-SIFT

PCA-SIFT [17] is, in spite of its name, not that related to SIFT. PCA-SIFT first samples gradients from the image patch along the x and y-axis at a resolution of 39×39. This results in a $39 \cdot 39 \cdot 2 = 3042$ dimensional vector that is normalized to unit magnitude to obtain illumination invariance. The descriptor is then reduced to 36 dimensions using PCA. The PCA projection matrix is generated from the covariance of 21,000 image patches extracted .

GLOH

The *gradient location-orientation histogram* (GLOH) [30] descriptor is inspired by both SIFT and PCA-SIFT. GLOH replaces the 4×4 grid of SIFT by a polar arrangement with a total of 17 gradient orientation histograms (see Figure 2.2 for a simplified depiction). Bilinear weighting similar to SIFT is used for both the orientation binning and for gathering histogram contributions. However, GLOH quantize gradient orientations in 16 bins and the bilinear weighting for the histogram contributions is calculated using polar coordinates. Because not all the histogram summation regions have the same size, each histogram is normalized by the area of its summation region. The 17 histograms of 16 bins each yields a descriptor with 272 dimensions. PCA is then used to reduce the dimensionality to 128.

DAISY

The DAISY descriptor is based on gradient orientation histograms arranged in a polar grid similar to GLOH. However, it uses Gaussian apertures for both the orientation binning and for gathering histogram contributions. The default DAISY descriptor uses 24 histograms of 8 bins each yielding a descriptor of 192 dimensions. A simplified illustration of the DAISY descriptor is shown in Figure 2.2. All histograms are normalized separately using the L^2 norm to make the descriptor more robust towards partial occlusions since the non-occluded might corrupt the non-occluded part if.

SURF

Speeded-up robust features (SURF) [5] uses a 4×4 grid similar to SIFT. Instead of using gradient orientation histograms, SURF gathers the sum of first-order Haar wavelet responses for each cell in the grid. This requires 4 dimensions per cell such that the dimensionality of SURF becomes 64. Similar to SIFT, a Gaussian window is applied to the Haar wavelet responses to decrease the importance peripheral image structure in the image patch. Contrast invariance is achieved by L^2 normalizing the SURF descriptor.

Moment invariants

Moment invariants [48] are generated up to the second order (excluding the zeroth order) and second degree. The moments are computed on the image patch derivatives along the x and y-axis respectively with $M_{ij}^a = \frac{1}{xy} \sum_x \sum_y x^p y^q (I_d(x,y))^a$, where $p+q$ is the order, a is the degree, and I_d is the image derivates along the direction d. This yields a 20-dimensional vector.

2.5 Performance evaluation

Image descriptor performance has been quantified using various datasets and evaluation criteria. Each method has its own bias, and descriptors are easily fitted to a specific dataset or to a certain application (e.g. observe the

performance discrepancy of the SURF descriptor between its original paper [5] and the DAISY paper [44]). The purpose of this section is to give an overview of existing evaluation methods and datasets to provide a perspective on the method used in this project. Recall that this project is limited to consider only performance of local descriptors with respect to the feature matching problem.

Matching strategy

The *matching strategy* specifies how we should predict whether two features match or not. It relies on a *similarity measure*, typically the distance between the feature descriptors. Often, the Euclidean distance is used as similarity measure. In [30], three possible matching strategies are listed:

Distance Given two features, we compute their distance according to the similarity measure and predict a match if their distance is below a certain threshold value.

Nearest neighbor distance Given a feature and a set of possible feature matches, a match is predicted only for the nearest neighbor in the set of features, and only if the distance to the nearest neighbor is below a certain threshold.

Nearest neighbor distance ratio Similar to the nearest neighbor distance, except the distance threshold is replaced by a threshold on the ratio between the distances to the nearest neighbor and the second nearest neighbor.

Compared to the neighbor-based strategies, the plain distance strategy works independently of the other features extracted, since it considers only the two features at hand. Neighbor-based strategies capture the discriminative ability relative to the extracted features as they only allow a single match between a feature and a set of features. The nearest neighbor distance ratio strategy reflect the descriptor's discriminative ability to an even greater extent as it requires the nearest neighbor match to stand out from the other features.

Following [7,30], we use the nearest neighbor distance ratio as matching strategy in this project as it achieves the best matching performance [30]. However, we remark that the plain distance matching may be useful in some situations where the computation required to find nearest neighbors is too demanding.

Evaluation criteria

The matching criteria provides us with matching predictions for the local features. If we know the true matching of the features, we can quantify the performance of a descriptor according to an *evaluation criterion*.

In [6], the *receiver-operating characteristics* (ROC) curve [12] was proposed as a performance measure of local image descriptors. The ROC curve is a plot of the *recall* vs. the *fall-out* for varying values of the discrimination threshold.

Recall and fall-out are defined as

$$\text{recall} = \frac{\text{\# of correct positives}}{\text{total \# of positives}} \quad , \quad \text{fall-out} = \frac{\text{\# of false positives}}{\text{total \# of negatives}} \quad . \tag{2.1}$$

Subsequently, a slightly different evaluation measure was proposed by Ke and Sukthankar in [17] to benchmark the PCA-SIFT descriptor. Instead of the ROC curve, they used the recall vs. $1 - \textit{precision}$ curve with the following argument. Compared to the fall-out, the $1 -$ precision is the probability of a misprediction when predicting a match.

$$1 - \text{precision} = \frac{\text{\# of false positives}}{\text{total \# of matches predicted (correct or false)}} \tag{2.2}$$

Ke and Sukthankar argue in favor of $1 -$ precision since the total number of negatives is not well-defined in the feature matching problem. In fact, the number of negatives is not a property of the image input; it is rather a property of the detector. For a more in-depth study of the relationship between the two curves, see [10].

The recall vs. $1 -$ precision measure has become the most popular criterion as it was also used in a broad comparative evaluation of different descriptors in [30] by Mikolajczyk and Schmid. Their framework for evaluating descriptors has since been used frequently as a basis for evaluating new descriptors [5, 33, 41].

Recently, Dahl et al. introduced a new dataset in [2, 7] and went back to the ROC measure based on recall vs. fall-out. However, instead of plotting the curve, they quantified the descriptor performance by measuring the area under curve (AUC) ending up with a single value. The AUC is equivalent to the probability that a randomly chosen true match will be ranked higher than a randomly chosen negative match [12]. We use the AUC evaluation measure in this project since we also use the accompanying dataset and this allows us to compare our results with those reported in [2, 7].

Datasets

There has been numerous local descriptor evaluations on different datasets. Most of these have been very specialized and therefore barely representative for the general application of descriptors [4–6, 32, 44]. For that reason, we focus on popular and more comprehensive datasets in this section.

In [31], Mikolajczyk and Schmid conducted the first broad comparative evaluation of different descriptors. For the benchmarking, they used a dataset consisting of eight different natural scenes exhibiting different combination of nuisance factors: view angle, rotation, scale, focus, exposure level and compression artifacts. Each scene is represented by six images yielding a total of 48 images for the entire dataset. The dataset is limited to planar scenes since the ground truth of the dataset is based on a manual specification of image correspondences approximating the homography between images. As the dataset comes with a framework for evaluating descriptors, it has become a popular benchmark in many later publications [5, 15, 33, 41, 44].

Obtaining ground truth when creating a dataset for descriptor evaluation is a major obstacle as it requires knowledge of the exact 3D structure being captured as well as the camera configuration. This is necessary in order to determine if two points in different images point to the same point on the 3D structure. Mikolajczyk and Schmid achieve ground truth by imposing a limitation on the 3D structure. However, this is unfortunate because it introduces a bias. For example, affine detectors perform well on their dataset [29] compared to a dataset with non-planar surfaces [1]. Another approach to achieve ground truth is taken in [52] where the 3D structure is estimated using wide baseline stereo matching and structure from motion on a large number of images. However, the dataset is somewhat limited as it consists of only three scenes and offers little control over individual nuisance factors.

A newer dataset introduced by Aanæs et al. [2] offers better ground truth as the scenes are scanned using structured light to obtain a 3D model. The dataset contains 60 different scenes containing various object types. Each scene is photographed from 119 camera positions to simulate change in view angle and scale. Additionally, for each camera position, the scene is photographed using 19 different light sources to simulate changes in illumination. Compared to previous datasets, this dataset is clearly superior in terms of size and control over nuisance factors. Though, the dataset does not contain any camera rotations around the optical axis and the color temperature of the scene lighting is constant. We use this dataset as it allows for a realistic and thorough evaluation of descriptor performance compared to the other datasets.

Recently, virtual scenes of photorealistic quality have been proposed [16, 50]. These datasets are superior in many regards because they offer better control over the image deformations. Furthermore, optimal ground truth is trivially achieved as the entire 3D structure is known. The only downside to photorealistic datasets is their questionable similarity to natural photographs which may cause a bias on the descriptor performance. Kaneva et al. [16] address this issue by comparing descriptor performance on a virtual model of the Statue of Liberty vs. the real world model and find a similar descriptor performance.

3 Differential image structure

The approach to local image description taken in this project is based on *scale-space theory*. This section provides an overview of the scale-space framework needed to construct the local image descriptors in later sections. The content of this section is based on [13, 14, 23, 24, 36, 42, 43]

The scale-space representation that we consider is the *linear scale-space* of a two-dimensional image, $I(\boldsymbol{r}) : \Omega \to \mathbb{R},\ \Omega \subseteq \mathbb{R}^2$, defined by

$$L(\boldsymbol{r};\sigma) = (G * I)(\boldsymbol{r};\sigma) \quad , \quad \boldsymbol{r} = (x, y) \tag{3.1}$$

Convolution is denoted by $*$ and G is the Gaussian aperture function

$$G(\boldsymbol{r};\sigma) = \frac{1}{(\sigma\sqrt{2\pi})^d} \exp\left(-\frac{\boldsymbol{r}\cdot\boldsymbol{r}}{2\sigma^2}\right) \quad , \quad \sigma \geq 0 \quad , \quad G(\boldsymbol{r};0) \equiv \delta \quad , \tag{3.2}$$

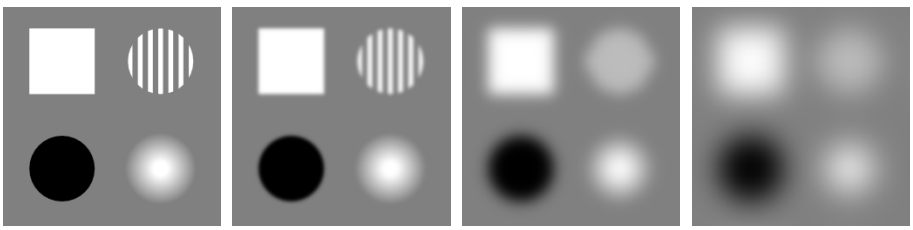
where δ is the Dirac delta function and σ (also known as the *inner scale* parameter) defines the width of the Gaussian kernel. The larger σ is, the more blurred the image will be yielding a lower resolution with fewer details, see Figure 3.1 for an example. The semicolon in $L(\boldsymbol{r};\sigma)$ indicates that the convolution is performed over the spatial coordinates $\boldsymbol{r}$ while the arguments following the semicolon are parameters of the observation.

The differential structure of the image is generated from the scale space derivatives calculated from

$$L_{x^n y^m}(\boldsymbol{r};\sigma) = \frac{\partial^{n+m}}{\partial x^n \partial x^m} (G * I)(\boldsymbol{r};\sigma) = \left(\left(\frac{\partial^{n+m}}{\partial x^n \partial y^m} G\right) * I\right)(\boldsymbol{r};\sigma) \quad , \tag{3.3}$$

where n and m indicate the order of the differential along the x and y axis respectively. Notice that differentiation commutes with convolution allowing us to differentiate the Gaussian kernel to get image derivatives. In fact, we cannot differentiate $I(\boldsymbol{r})$ since we cannot assume it is differentiable, however, $(G * I)(\boldsymbol{r};\sigma) \in C^\infty$ since $G(\boldsymbol{r};\sigma) \in C^\infty$.

The differentiated Gaussian kernels up to the third order are shown in Figure 3.2. Convolving these with the image from Figure 3.1 yields the image derivatives shown in Figure 3.3.



$L(\boldsymbol{r};\sigma = 0)$ $L(\boldsymbol{r};\sigma = 2)$ $L(\boldsymbol{r};\sigma = 8)$ $L(\boldsymbol{r};\sigma = 16)$

Figure 3.1: Example of a 2D image scale-space for increasing values of σ. The leftmost image is the original image $I(\boldsymbol{r})$ since $I(\boldsymbol{r}) = L(\boldsymbol{r}, \sigma = 0)$.

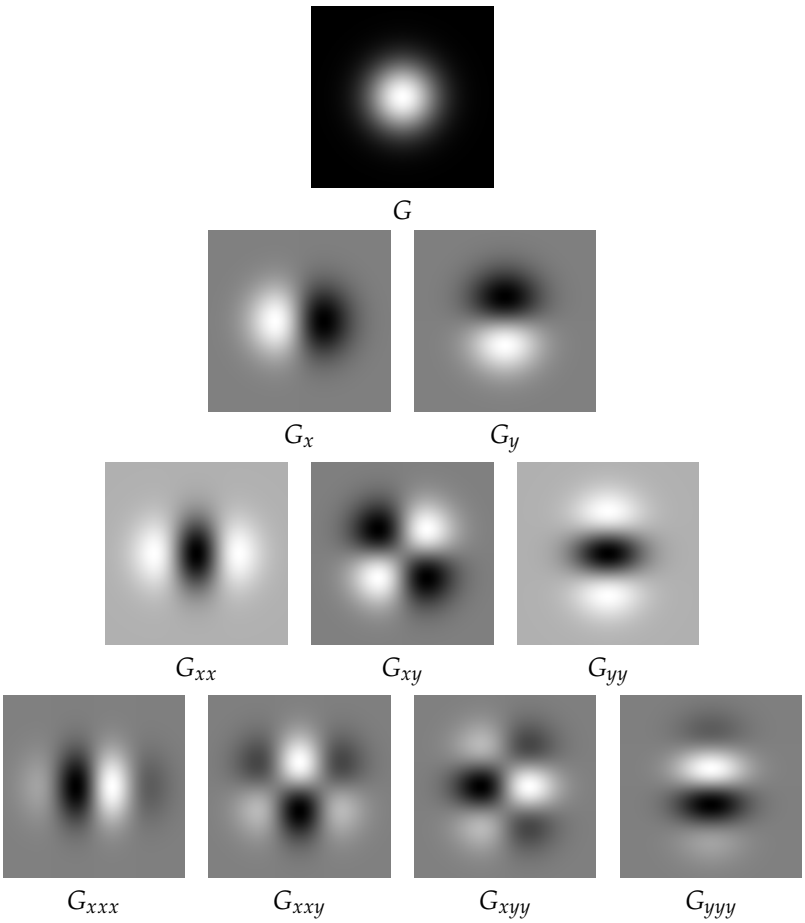


Figure 3.2: The differentials of the 2D Gaussian kernel up to the third order. Note that the order of the differentiation does not matter, eg. $G_{xxy} = G_{xyx} = G_{yxx}$.

In the following we shall cover different differential structure measures. For notational convenience we substitute $L_{x^n y^m}(\boldsymbol{r}; \sigma)$ with simply $L_{x^n y^m}$ and implicitly assume $L_{x^n y^m}$ to be computed at some scale σ and location $\boldsymbol{r}$.

3.1 The local k-jet

A simple method to describe an image patch would be to extract the pixels intensities in a vector and use that as a descriptor. Patches could then be compared using *normalized cross-correlation* to account for variability with respect to brightness and spatial shifting. However, this approach yields a high-dimensional feature descriptor with subpar discriminative abilities (see [30]).

Instead of representing the pixels raw, we could approximate them using the Taylor series. Note that image derivatives from Eq. 3.3 correspond to the coefficients of the Taylor series of the smoothed image function $I(\boldsymbol{r})$ as they describe the local geometry in a neighborhood of the image function.

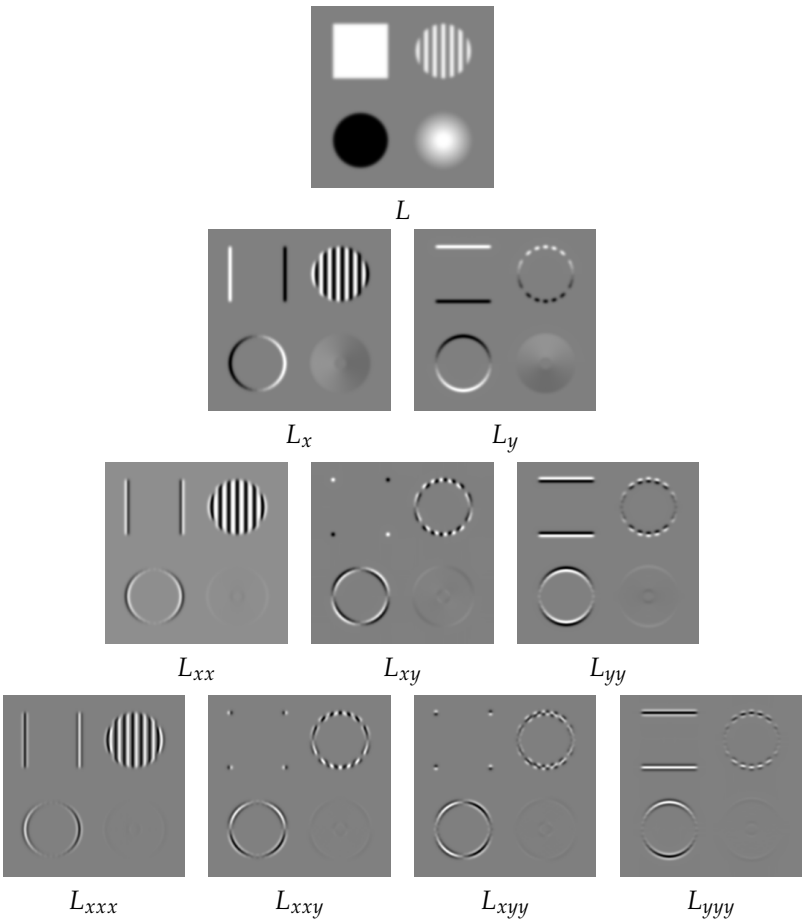


Figure 3.3: Image differentials up to the third order. The different derivatives capture different structures in the image.

For example, consider the neighborhood $\boldsymbol{a} = (x, y)$ described by the Taylor expansion around the point $\boldsymbol{r} = (0, 0)$:

$$L(\boldsymbol{a};\sigma) = L + L_x x + L_y y + \frac{1}{2}\left(L_{xx}x^2 + 2L_{xy}xy + L_{yy}y^2\right) + \frac{1}{6}\left(L_{xxx}x^3 + 3L_{xxy}x^2y + 3L_{xyy}xy^2 + L_{yyy}y^3\right) + \ldots \tag{3.4}$$

In practice, we approximate the local geometry by truncating the Taylor expansion to some degree k and excluding the zeroth order component. The zeroth order component L is uninteresting when we wish to be invariant to changes in brightness since it describes the average image intensity value of the local image patch. We refer to the vector of image derivatives as the *local k-jet* $\mathcal{J}_k$ [13]:

$$\mathcal{J}_k = \left(\{L_{x^n y^m} | 0 < n + m \leq k\}\right)^T \quad , \quad \mathcal{J}_k \in \mathbb{R}^K \quad , \quad K = \frac{(2+k)!}{2k!} - 1 \tag{3.5}$$

Thus, the local 3-jet $\mathcal{J}_3$ of an image is given by

$$\mathcal{J}_3 = (L_x, L_y, L_{xx}, L_{xy}, L_{yy}, L_{xxx}, L_{xxy}, L_{xyy}, L_{yyy})^T \quad . \tag{3.6}$$

An alternative more compact way to express the local Taylor expansion from Eq. 3.4 is

$$L(\boldsymbol{a}; \sigma) = L + \nabla L \boldsymbol{a} + \frac{1}{2} \boldsymbol{a}^T \nabla^2 L \boldsymbol{a} + \ldots \tag{3.7}$$

where ∇L and $\nabla^2 L$ respectively denote the gradient and the Hessian matrix

$$\nabla L = \begin{bmatrix} L_x & L_y \end{bmatrix} \quad , \quad \nabla^2 L = \begin{bmatrix} L_{xx} & L_{xy} \\ L_{yx} & L_{yy} \end{bmatrix} \quad . \tag{3.8}$$

In the following we shall see other ways to extract image geometry from these differentials.

3.2 The gradient orientation

The first order image structure, ie. the structure captured by ∇L, describes the linear change in image intensity. From L_x and L_y we can derive the *gradient orientation* $\theta \in]-\pi; \pi[$ from the angle of the gradient vector and the *gradient magnitude* m from the length of the gradient vector.

$$\theta = \text{atan2}\left(L_x, L_y\right) \quad , \quad m = \sqrt{L_x^2 + L_y^2} \tag{3.9}$$

The magnitude describe the strength of the change in image intensity along the direction θ. The gradient orientation weighted by its magnitude is used in histograms by many of the popular descriptors such as SIFT, DAISY, GLOH and HOG. However, θ may not be computed exactly as above, for example in the original HOG implementation, L_x and L_y are computed by simply filtering the image with the linear kernels $[-1, 0, 1]$ and $[-1, 0, 1]^T$ and θ is used *unsigned* spanning only 180°.

The gradient orientation becomes unstable for $L_x \to 0$ and $L_y \to 0$. However, in practice this is not problematic since θ is usually weighted by m to lessen the influence of small gradients. Furthermore, θ is undefined for $L_x = L_y = 0$. In practice, this problem is circumvented by defining $\text{atan2} \equiv 0$.

3.3 The shape index

Whereas the gradient describes linear intensity change of a point in the image, the Hessian matrix $\nabla^2 L$ describes the *curvature* around that point. Note that the Hessian matrix is square and symmetric allowing us to compute the pair of real eigenvalues κ_1 and κ_2 from

$$\begin{aligned} \kappa_1 &= \frac{1}{2}\left(L_{xx} + L_{yy} - \sqrt{4L_{xy}^2 + L_{xx}^2 + L_{yy}^2 - 2L_{xx}L_{yy}}\right) \\ \kappa_2 &= \frac{1}{2}\left(L_{xx} + L_{yy} + \sqrt{4L_{xy}^2 + L_{xx}^2 + L_{yy}^2 - 2L_{xx}L_{yy}}\right) \quad . \end{aligned} \tag{3.10}$$

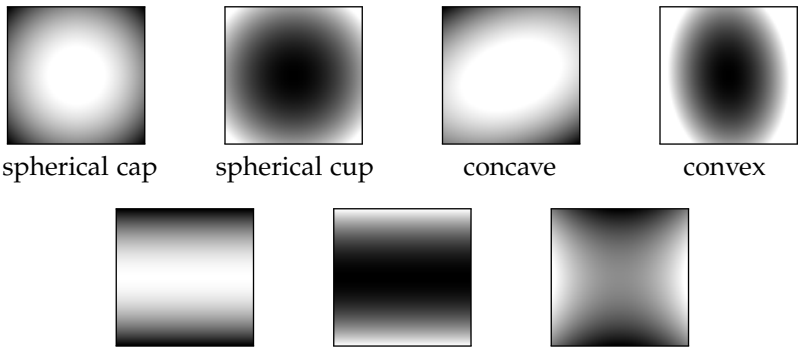


Figure 3.4: Examples of curvature. The spherical shapes are specializations of the convex and the concave shapes where the two principal curvatures are equally large. Ridges and ruts are known as cylindrical shapes that occur when one of the principal curvatures is zero. Saddle points occur when the principal curvatures have opposite sign.

Figure 3.5: Curvature examples sorted along the range of the shape index.

κ_1 and κ_2 are known as the *principal curvatures*. They describe the strength of the curvature along the *extremal directions* where the curvatures are minimal and maximal respectively. Examples of local curvature are concave, convex, saddle and cylindrical shapes as illustrated in Figure 3.4. The curvature in a point is symmetric, i.e. the curvatures in opposite directions are the same. Note that κ_1 and κ_2 do not capture the orientation of the curvature. The extremal directions are generated from the eigenvectors of $\nabla^2 L$, however, they are not used since we want to be invariant to rotation in the following.

In [19], Koenderink and van Doorn propose an image geometry measure based on the principal curvatures called the *shape index*. The shape index $s \in]-1;1[$ is defined as

$$s = \frac{2}{\pi} \arctan\left(\frac{\kappa_1 + \kappa_2}{\kappa_1 - \kappa_2}\right) \quad . \tag{3.11}$$

The shape index has the attractive property that it maps all curvature shapes onto a continuous range providing a smooth and intuitive transition between the shapes. See Figure 3.5 for an illustration of the relation between curvature and s.

Along with the shape index comes a *curvedness* measure c of the curvature:

$$c = \frac{1}{2}\sqrt{\kappa_1^2 + \kappa_2^2} \tag{3.12}$$

c indicates the strength of the curvature given by the shape index. It has an application similar to the magnitude of the gradient orientation since it can

be used to weight the contribution of s to e.g. a histogram. The geometric structure captured by the shape index is, however, much different from the structure captured by the gradient orientation. The shape index describes only the second order structure invariant to linear local structure and furthermore it is rotation invariant.

In the literature, no local image descriptors has used the shape index directly to capture local structure. However, the differential invariant descriptor [4] captures differential structure with similar invariants.

An alternative analytical formulation of s and c that skips the explicit calculation of the Hessian eigenvalues is given by

$$s = \frac{2}{\pi} \arctan \left(\frac{-L_{xx} - L_{yy}}{\sqrt{4L_{xy}^2 + \left(L_{xx} - L_{yy}\right)^2}} \right) \tag{3.13}$$

$$c = \frac{1}{2} \sqrt{L_{xx}^2 + 2L_{xy}^2 + L_{yy}^2} \quad . \tag{3.14}$$

4 Locally orderless images

The idea behind the *locally orderless image* (LOI) representation is to express the uncertainty of an observation as a probability distribution approximated by histograms instead of a single averaging value.

Koenderink and van Doorn [20] argue in favor of the LOI representation because of its close resemblance to the functioning of natural visual perception systems. An observed scene has infinite resolution (in theory) but the observer is only able to sample images of finite resolution of that scene. Instead of representing a point in the sampled image by a single value, a histogram is gathered to better capture the observation uncertainty of that image point.

In the original LOI formulation, the goal is to represent image intensities as histograms. To construct these, a set of bin intensity levels $B = \{b_i | i = 1, \ldots, q\}$ is chosen. The contribution to each intensity level b_i is computed using the Gaussian window

$$R(\boldsymbol{r}, b_i; \sigma, \beta) = \frac{1}{2\pi\beta^2} \exp\left(-\frac{(L(\boldsymbol{r};\sigma) - b_i)^2}{2\beta^2}\right) \quad , \tag{4.1}$$

where $L(\boldsymbol{r};\sigma)$ is the linear scale space from Eq. 3.1 and β adjusts the width of the Gaussian, ie. the bin width of the histogram. β is also referred to as the *tonal scale* parameter. We can think of $R(\boldsymbol{r}, b_i; \sigma, \beta)$ as a *soft isophote image* since it captures the smoothed level curves around the intensity b_i.

The LOI representation for an intensity level b at the image coordinates $\boldsymbol{r}_0$ is defined as

$$H(\boldsymbol{r}_0, b; \alpha, \beta, \sigma) = A(\boldsymbol{r}; \boldsymbol{r}_0, \alpha) * R(\boldsymbol{r}, b; \sigma, \beta) \tag{4.2}$$

where A is the neighborhood *aperture* function

$$A(\boldsymbol{r}; \boldsymbol{r}_0, \alpha) = \frac{1}{2\pi\alpha^2} \exp\left(-\frac{(\boldsymbol{r} - \boldsymbol{r}_0) \cdot (\boldsymbol{r} - \boldsymbol{r}_0)}{2\alpha^2}\right) \quad . \tag{4.3}$$

The *outer scale* parameter α defines the width of the smoothing window. Compared to the inner scale parameter that regulates image resolution, the outer scale parameter specifies a local Gaussian window from which the histogram contributions are gathered. Thus, going from $L(\boldsymbol{r}_0;\sigma)$ to $H(\boldsymbol{r}_0, b; \alpha, \beta, \sigma)$, we loose the local spatial ordering of the pixels around $\boldsymbol{r}0$ by replacing it with a histogram. The spatial ordering is, of course, still retained at scale α.

To reiterate, when constructing LOIs we must choose a suitable set of parameters to capture the observed image:

- The inner scale σ determines the scale of the details observed in the scene.
- The outer scale α determines the width of the window used to gather statistics about details in an observation point. If α is chosen too small,

the window fails to gather sufficient data for the statistics to become representative. Conversely, if α is chosen too large, the statistics reflect a more global structure rather than the local structure. Typically, $\alpha > \sigma$ since we want to capture some diversity in the local observation.

- The bin centers B regulates the construction of the histogram. When specifying B, an appropriate number of bins must be chosen as well as their distribution. Normally, the bin centers are distributed uniformly in the span of the observed values.
- The tonal scale β determines the amount of blur applied to the histogram bins as well as the overlap between adjacent bins. That is, the softness of the isophote images. Similar to α, we want to strike a balance that yields representative statistics.

The relationship between LOIs and local feature description is clear in terms of the descriptor construction pipeline from Section 2.2. Recall that, typically, feature description can be decomposed into four steps: Image feature extraction, histogram binning, spatial pooling and normalization. The LOI representation cover the first three steps since $L(\boldsymbol{r}, \sigma)$ can be regarded as the image feature extracted and $R(\boldsymbol{r}, b; \sigma, \beta)$ the histogramming step. The spatial pooling step is performed by $H(\boldsymbol{r}_0, b; \alpha, \beta, \sigma)$ where the histogram bin contributions are gathered according to the spatial aperture $A(\boldsymbol{r}; \boldsymbol{r}_0, \alpha)$. For multi-local description (e.g. the SIFT grid), we sample $H(\boldsymbol{r}_0, b; \alpha, \beta, \sigma)$ at multiple locations.

4.1 LOIs from image differentials

In the above review of the LOI representation, we have only considered histograms of image intensities. For local feature description, we typically wish to capture the differential structure of the image since it describes the local geometry better. This approach is similar to that of descriptors relying on multi-local histograms of image gradients (e.g. SIFT, DAISY and HOG) which has proven to be very efficient [7, 30]. An explanation of the efficacy of collecting first order image structure in multi-local histograms could be that the histograms provide a downsampling compared to the raw pixel representation. This downsampling is less sensitive to the precise locations of the local image elements and therefore offers robustness towards geometric distortions such as changes of perspective and translation. Moreover, the gradient orientation histograms are normalized making them robust to changes in brightness.

In the following, we extend the LOI formulation by including image differentials to imitate the above popular descriptors.

Gradient orientation

In the LOI context we use gradient orientations by replacing $R(\boldsymbol{r}, b; \sigma, \beta)$ from Eq. 4.2 with

$$R_\theta(\boldsymbol{r}, b; \sigma, \beta) = m(\boldsymbol{r}; \sigma) \frac{\exp\left(\beta^{-2} \cos(\theta(\boldsymbol{r}; \sigma) - b)\right)}{2\pi I_0(\beta^{-2})} \quad . \tag{4.4}$$

Compared to $R(\boldsymbol{r}, b; \sigma, \beta)$ from Eq. 4.1 we have performed three noteworthy changes. We have replacing the image intensity term $L(\boldsymbol{r}; \sigma)$ with the gradient orientation $\theta(\boldsymbol{r}; \sigma)$. Secondly, we use the gradient magnitude $m(\boldsymbol{r}; \sigma)$ as a coefficient to weight the histogram bin contribution of the gradients. This means that small gradients contribute less to the histogram than large gradients. Finally, since the gradient orientation domain is cyclic over the range $[-\pi, \pi]$ we replace the Gaussian bin aperture with the circular von Mises aperture where $I_0()$ is the modified Bessel function of order 0. The von Mises distribution is an approximation of the wrapped normal distrubution on the circle. Note that the support of the von Mises distribution is defined as any interval of length 2π, which fits with the range of θ. If this was not the case we should have rescaled θ to a range of length 2π.

The histogram contributions calculated in Eq. 4.4 are very similar to those of SIFT, DAISY, HOG, etc. However, in SIFT, the bin contributions are weighted by an additional Gaussian window placed at the center $\boldsymbol{r}_c$ of the local image patch such that the importance of peripheral gradients is diminished. In the LOI framework we can imitate the SIFT weighting $m_{\text{SIFT}}(\boldsymbol{r}; \sigma)$ of a histogram bin contribution from

$$m_{\text{SIFT}}(\boldsymbol{r}; \sigma) = \frac{m(\boldsymbol{r}; \sigma)}{2\pi\gamma^2} \exp\left(-\frac{(\boldsymbol{r} - \boldsymbol{r}_c) \cdot (\boldsymbol{r} - \boldsymbol{r}_c)}{2\gamma^2}\right) \quad , \tag{4.5}$$

where γ is the width of the Gaussian. Remark that this aperture is not to be confused with the neighborhood aperture $A(\boldsymbol{r}; \boldsymbol{r}_0, \alpha)$ around the location $\boldsymbol{r}_0$. Also, this additional weighting only makes sense for multi-local description like SIFT's grid of histograms because it offers a weighting of the bin-contributions across histograms. If we were to use only a single histogram placed at the center of of the image patch, the extra Gaussian window would align with $A(\boldsymbol{r}; \boldsymbol{r}_0, \alpha)$ causing a change of scale.

Shape index

Like with the gradient orientation in the LOI formulation, we use the shape index s as replacement for the image intensity values and its curvedness c to weight the histogram contributions of the shapes.

$$R_s(\boldsymbol{r}, b; \sigma, \beta) = \frac{c(\boldsymbol{r}; \sigma)}{2\pi\beta^2} \exp\left(-\frac{(s(\boldsymbol{r}; \sigma) - b)^2}{2\beta^2}\right) \tag{4.6}$$

Since the space of the shape index is not to be considered periodic, ie. $s = -1$ (concave) is not similar to $s = 1$ (convex), the Gaussian aperture and not the von Mises aperture is used to blur the bin contributions. This, in turn, leads to another problem at the boundaries of the interval $]-1; 1[$ because the area under the Gaussian is truncated outside this interval. See Figure 4.1 for an illustration of the situation. To compensate for this, we calculate the normalization factor z as the area under the Gaussian with variance β and mean b on the interval $]-1; 1[$,

$$z = \int_{-1}^{1} \exp\left(\frac{-(x - b)^2}{2\beta^2}\right) \mathrm{d}x \quad , \tag{4.7}$$

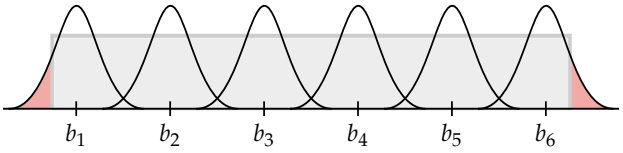


Figure 4.1: When constructing a histogram with Gaussian bin apertures on a finite interval (like the shape index), we should apply different normalization factors depending on the area under the Gaussian on the interval. This compensates for the truncation of the Gaussians that we observe at the boundaries of the interval above.

and the histogram contributions of the shape index are now calculated from

$$R_s(\boldsymbol{r}, b; \sigma, \beta) = \frac{c(\boldsymbol{r}; \sigma)}{z} \exp\left(-\frac{(s(\boldsymbol{r}; \sigma) - b)^2}{2\beta^2}\right) \quad . \tag{4.8}$$

The shape index collected in histograms is a novel idea for local image description (at least to our knowledge). Compared to histograms of gradient orientations, the shape index captures second order image structure. Thus, we consider it a supplement to the histograms of gradient orientations that might diminish the need for multi-local description since it captures more elaborate geometry.

Local k-jet

Similar to the above two measures of local image structure, we can use the local k-jet $\mathcal{J}_k$ to capture image geometry in histograms. However, the local k-jet is not well suited for histogramming for the following reasons.

The LOI histogram generated from $\mathcal{J}_k$ becomes high-dimensional since the local k-jet is multidimensional. If we choose to represent the joint distribution of $\mathcal{J}_k$, the dimensionality of the histogram grows exponentially with the dimensionality of $\mathcal{J}_k$. This quickly surpasses the dimensionality of e.g. the SIFT descriptor for even small k. If we choose to capture the marginalized distribution of $\mathcal{J}_k$, the exponential growth is circumvented, however, we cannot be sure that the marginal distributions are representative for the jet manifold. Moreover, the dimensionality of the marginalized histograms are still somewhat high. For example, consider the dimensionality of the marginalized histograms for $\mathcal{J}_4 \in \mathbb{R}^{14}$. If we generate a histogram of 8 bins for all 14 dimensions, the descriptor dimensionality will be 112.

Another problem regarding LOI from the local k-jet is that the derivative coefficients are not contained in a finite interval making it considerably harder to determine a representative set of intensity levels b.

In this project, LOIs based on the local k-jet have been attempted for local image description. However, the approach quickly turned out to be unsuccessful as we were not able to get good results. For this reason, and for the reasons stated above, no further experiments were carried out. We will, however, perform experiments with local k-jets on their own.

5 Proposed descriptors

In the previous two sections we have reviewed the scale-space framework for capturing differential image structure as well as LOIs for representing image structure statistics. In this section we describe how to create local feature descriptors using this framework. We also discuss what elements of the feature descriptor design we wish to investigate and present different descriptor proposals based on the discussion. Finally, we discuss different descriptor similarity measures. The actual evaluation of the descriptors is postponed until Section 7, where the descriptors will be compared to other state-of-the-art descriptors.

Recall from Section 2.1 that we use the canonized image regions (specified by the interest point detector) as input to the feature descriptors. Following [7,30], we extract image patches at three times the scale of the interest point region as depicted in Figure 5.1. This is done to include the image information around the border of the interest point region and to circumvent blurring errors along the image patch borders.

5.1 LOI descriptor

To construct a local image descriptor from LOIs we use Eq. 4.2,

$$H(\boldsymbol{r}_0, b; \alpha, \beta, \sigma) = A(\boldsymbol{r}; \boldsymbol{r}_0, \alpha) * R(\boldsymbol{r}, b; \sigma, \beta) \quad , \tag{5.1}$$

with R replaced by R_θ (Eq. 4.4) and R_s (Eq. 4.6) to capture the gradient orientation and the shape index, respectively. The LOI descriptor is simply constructed from a concatenation of these histograms. With the LOI framework as backbone in the descriptor construction pipeline from Section 2.2, we now have to decide on the following configuration options.

- The scale σ at which the image features θ and s are extracted.

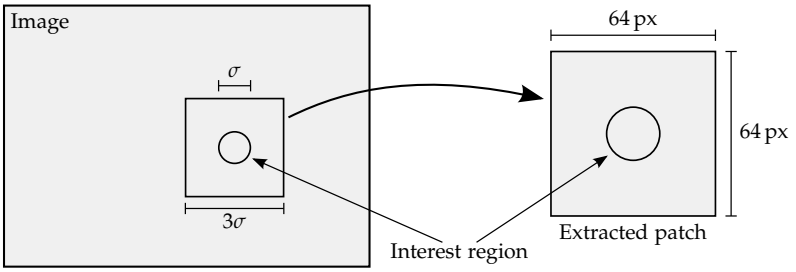


Figure 5.1: Canonization of an image region specified by an interest point of scale σ. We extract the image region at scale 3σ and resample it to 64×64 px.

- The construction of the histograms capturing the image features. Namely, the bin center locations B and their tonal aperture β.
- The spatial pooling strategy, i.e. how the histograms should be sampled from the local image region. This includes determining the number of histograms p, their locations $\boldsymbol{r}_j$ and their apertures α_j. Moreover, we have to find a good mix of gradient orientation and shape index histograms.
- How to normalize the histograms generated in the previous step.

Most of the parameters above are estimated using manual optimization, which will be described in Section 5.3.

Regarding the spatial pooling strategy, we suspect that the use of the shape index will reduce the need for multi-local description. Therefore, we will investigate if good performance can be achieved by (partly) replacing multi-locality by adding the shape index to the descriptor. The multi-local description we perform with the LOI descriptor will follow the grid structure of the SIFT descriptor. We could also have chosen a different sampling pattern, e.g. the polar grid of DAISY. However, there seems to be little difference in performance for different multi-local sampling patterns [7, 51].

Another angle we wish to investigate is the trade-off between the dimensionality of the descriptor vs. performance. We have seen that PCA-SIFT is able to achieve good performance compared to SIFT with the added convenience of a significantly lower descriptor dimensionality [15, 17]. Thus, we wish to explore if we can achieve SIFT-like performance by starting out with a very simple descriptor and increase the descriptor complexity (e.g. in terms of the number of histograms) until SIFT-like performance is reached. This approach contrasts the descriptor design by Winder et al. [51, 52] where the descriptor dimensionality is pushed as far as possible to improve performance.

Normalization

The last step in the LOI descriptor algorithm is to perform a normalization of the histograms. Recall from Section 2.3 that we assume the pixel intensity variations to follow the affine transformation $aI(\boldsymbol{r}) + b$. Both the gradient orientation and the shape index are invariant to varying offsets of image intensity since they do not rely on zeroth order image structure. The purpose of the normalization is to become invariant to contrast variations. For example, if the image intensity is multiplied by a constant factor a, the gradient magnitude will be multiplied by the same factor. A normalization of the gradient histogram will cancel such a transformation. The same argument goes for the shape index since the curvature measure c is affected similarly by linear transformations of image intensity.

Common normalization strategies are L^1 *norm*, L^2 *norm* (Euclidean norm) and *thresholded* L^2 *norm* [8, 25]. The thresholded L^2 norm is simply the L^2 norm followed by a clipping of values higher than a certain threshold, followed by a second L^2 normalization. According to Lowe [25], this ensures a better handling of certain nonlinear image intensity perturbations that leads to large changes in gradient magnitudes because the clipping reduces peaks in the histogram from such nonlinear transformations.

To complicate matters even further, we might choose to normalize the entire feature vector, the histograms independently or even the histograms group-wise (e.g. normalize all shape index histograms and all gradient orientation histograms separately).

The optimal choice of normalization might depend on the application. For example, Tola et al. [44] argue that for a stereo application, normalization should be performed independently per histogram such that occlusions covering some histograms do not affect the remaining histograms. In our evaluation of the LOI descriptors, we will investigates the importance of different normalization strategies. As default, we use the L^2-norm on individual histograms because it performs well which we will see in Section 7.

Proposals

The full list of descriptor proposals is given in Table 5.1 along with their parameters. The list contains selected LOI descriptors as we have removed the worst performing descriptor variants for clarity. We use a naming convention where 'GO' and 'SI' stand for gradient orientation and shape index respectively. If either of these are followed by the number 2 in subscript, it means that we sample them at two different aperture scales (a large α and a small α). If GO or SI are followed by the number 4 in subscript, it means that they are sampled at four different scales (e.g. small σ and small α; small σ and large α; etc.). If GO or SI are followed by '-grid$x$' it means that they are sampled multi-locally in a quadratic grid of width x. Note that due to time constraints in this project, we limit our descriptor proposals to not include combinations of multi-local and multi-scale samplings (this would significantly increase the complexity of the possible descriptor configurations).

Examples of the different variants of LOI descriptor proposals are illustrated in Figure 5.2. In the upper row, we try to do away with the multi-local histogram sampling by collecting gradient orientation and shape index histograms at different scales σ and apertures α. We call this approach *multi-scale* rather than multi-local. The sampling strategies in the bottom row are examples of our multi-local LOI descriptor proposals. Observe that the LOI-GO-grid4 descriptor is very similar to SIFT except for (mainly) three things: The LOI descriptor uses Gaussian apertures to smooth the histogram contributions spatially and to smooth the contribution over neighboring bins in a histogram whereas SIFT uses trilinear interpolation. Sencondly, SIFT uses the additional Gaussian window from Eq. 4.5. Finally, SIFT descriptor uses thresholded L^2 normalization where our LOI descriptor uses plain L^2 normalization.

5.2 Jet descriptor

In addition to the LOI descriptor we propose a descriptor based on the local k-jet. While descriptors based on differential invariants (functions of the local k-jet) have been proposed before [4, 39], no direct use of the local k-jet has been employed for local feature description.

The construction of our local k-jet descriptor (referred to as the *Jet descriptor*) is straightforward as we simply use $\mathcal{J}_k$ from Eq. 3.5. In addition, we scale

Variant name	Parameters	Dimensionality
LOI-GO	$\sigma_\theta = 1.2$, $\alpha_\theta = 26$	8
LOI-SI	$\sigma_s = 4.6$, $\alpha_s = 16$	8
LOI-GO_2	$\sigma_{\theta 1} = 1.2$, $\sigma_{\theta 2} = 1.2$, $\alpha_{\theta 1} = 6.8$, $\alpha_{\theta 2} = 28$	16
LOI-GO_4	$\sigma_{\theta 1} = 1.3$, $\sigma_{\theta 2} = 1.3$, $\sigma_{\theta 3} = 6.4$, $\sigma_{\theta 4} = 6.4$, $\alpha_{\theta 1} = 7.2$, $\alpha_{\theta 2} = 19.2$, $\alpha_{\theta 3} = 8.0$, $\alpha_{\theta 4} = 16.2$	32
LOI-SI_2	$\sigma_{s1} = 1.3$, $\sigma_{s2} = 6.4$, $\alpha_{s1} = 7.2$, $\alpha_{s2} = 19$	26
LOI-SI_4	$\sigma_{s1} = 1.3$, $\sigma_{s2} = 1.3$, $\sigma_{s3} = 6.4$, $\sigma_{s4} = 6.4$, $\alpha_{s1} = 7.2$, $\alpha_{s2} = 19.2$, $\alpha_{s3} = 8.0$, $\alpha_{s4} = 16.2$	32
LOI-GOSI	$\sigma_\theta = 1.1$, $\alpha_\theta = 28$, $\sigma_s = 1.6$, $\alpha_s = 18$	16
LOI-GO_4SI_2	LOI-GO_4 and LOI-SI_2 combined	48
LOI-GO_2SI_4	LOI-GO_2 and LOI-SI_4 combined	48
LOI-GO_4SI_4	LOI-GO_4 and LOI-SI_4 combined	64
LOI-GOSI-grid2	$\sigma_\theta = 1.0$, $\alpha_\theta = 9.6$, $\sigma_s = 2.6$, $\alpha_s = 10.2$	64
LOI-GOSI-grid4	$\sigma_\theta = 1.0$, $\alpha_\theta = 5.8$, $\sigma_s = 2.1$, $\alpha_s = 6$	256
LOI-GO-grid2	$\sigma_\theta = 0.8$, $\alpha_\theta = 9.6$	32
LOI-GO-grid4	$\sigma_\theta = 0.8$, $\alpha_\theta = 6$	128

Table 5.1: The LOI descriptor proposals and their parameters. All numbers are in pixel units. We allow ourselves to reuse the estimated parameters when combining the gradient orientation and the shape index, since the parameter space becomes too high dimensional to search through.

normalize the jet coefficients as described in [13, 36]. This gives us to the following options when configuring the descriptor:

- The scale σ at which the local k-jet features are extracted.
- The order k of the jet.
- The normalization method.

Compared to the LOI descriptor, the Jet descriptor is significantly simpler in terms of configuration possibilities. We perform a manual optimization of the parameters in the following Section 5.3.

Similar to the LOI descriptor, we wish to investigate to what extent a multi-scale or multi-local sampling strategy can improve performance. Namely, if we through multi-local sampling of low-order k-jets can achieve higher performance than sampling a single high-order jet.

Whitening and normalization

We perform a whitening of the local k-jet coefficients according to a statistical derivation of their covariant structure [26, 36]. This yields a vector where the elements are uncorrelated and of the same magnitude allowing a standard

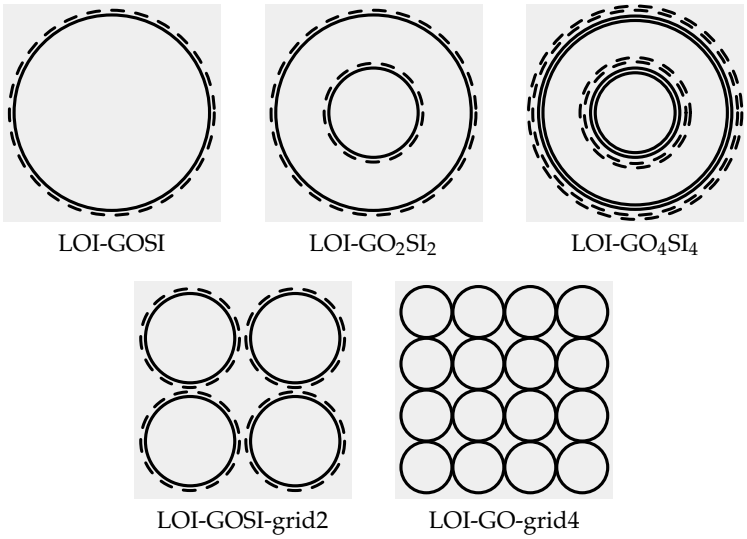


Figure 5.2: Examples of LOI descriptor proposals. The grey squares illustrate canonized image patches. Circles with solid and dashed lines indicate gradient orientation and shape index histograms respectively. The width of the circles represents the size α of the aperture. For LOI-GO_4SI_4 there are histograms with (approximately) the same aperture α. This indicates that their inner scale σ varies.

distance measure (e.g. the Euclidean distance) to be used. It is beyond the scope of this project to derive the local k-jet covariance matrix and explain its exact functioning. However, according to [36], the covariance between the jet coefficients $L_{x^i y^j}$ and $L_{x^k y^l}$, where both $n = i + j$ and $m = k + l$ are even, is given by

$$\operatorname{cov}(L_{x^i y^j}, L_{x^k y^l}) = (-1)^{\frac{n+m}{2}+k+l} \frac{\beta^2}{2\pi\sigma^{n+m}} \frac{n!m!}{2^{n+m}(n+m)\left(\frac{n}{2}\right)!\left(\frac{m}{2}\right)!} \quad . \tag{5.2}$$

σ is the scale parameter of the local k-jet, and β is a model parameter that is irrelevant in our context. If not n and m are even, the covariance is 0. In [36], the k-jet covariance is estimated empirically from jets on natural images and is found to be similar to the above theoretical result. Finally, we remark that this normalization method is related to the descriptor similarity measure proposed in [4].

After the whitening, the descriptor is L^2 normalized to achieve image contrast invariance. This places the vectors on the unit D-sphere where D is the dimensionality of the jet.

Proposals

The jet descriptors we propose are listed in Table 5.2. Their naming is following the same convention as for the LOI descriptors. 'Jet-k' refers to the local k-jet while the suffix '-grid#' and the subscript '$_\#$' indicate the degree of multi-local

Variant name	Parameters	Dimensionality
Jet-3	$\sigma = 10.6$	9
Jet-4	$\sigma = 10.6$	14
Jet-5	$\sigma = 10.6$	20
Jet-6	$\sigma = 10.6$	27
Jet-7	$\sigma = 10.6$	35
Jet-4_2	$\sigma_1 = 7.5$, $\sigma_2 = 16$	28
Jet-5_2	$\sigma_1 = 7.5$, $\sigma_2 = 16$	40
Jet-3-grid2	$\sigma = 6.8$	36
Jet-3-grid4	$\sigma = 5.2$	144
Jet-4-grid2	$\sigma = 6.8$	56
Jet-5-grid2	$\sigma = 6.8$	80

Table 5.2: The Jet descriptor proposals and their parameters. All numbers are in pixel units. Observe that the σ parameter is significantly lower than the neighborhood aperture α for the LOI descriptors in Table 5.1. This means that the Jet descriptor does not rely on the data outside the interest point image region. Recall that we extract the image region at a scale three times larger than the image region of interest to include image geometry around the border. This extra information does not seem to be useful for the Jet descriptor.

and multi-scale sampling respectively. We do not illustrate the Jet descriptors as the LOI illustrations in Figure 5.2 are similar because they follow the same multi-local and multi-scale sampling strategy.

5.3 Parameter optimization

Both the Jet and the LOI descriptors contain free parameters for which we must find suitable values that maximize the performance of the descriptors. To estimate these values we perform optimization on the Oxford dataset using the code made available by Mikolajczyk and Schmid [30][1]. Because we do not consider rotation, we optimize only on the *Leuven* scene from the dataset since it contains no significant rotations around the optical axis. Note that we evaluate descriptor performance on the dataset by Aanæs et al. [2] but that we optimize parameters on the Oxford dataset to avoid fitting our parameters to the evaluation data. Furthermore the Oxford code uses the recall vs. $1 - \text{precision}$ measure whereas the DTU dataset is evaluated with the recall vs. fall-out measure (as discussed in Section 2.5).

We employ a somewhat ad-hoc manual optimization method to estimate the descriptor parameters. We search through the range of parameter values and use the configuration that yields the best performance (see Figure 5.3 for an example of the performance of the LOI-GO descriptor as a function of σ and α). In practice, however, the dimensionality of the parameter space is too high for us to search through all possible parameter combinations.

[1] Dataset and evaluation code available at `http://www.robots.ox.ac.uk/~vgg/research/affine`.

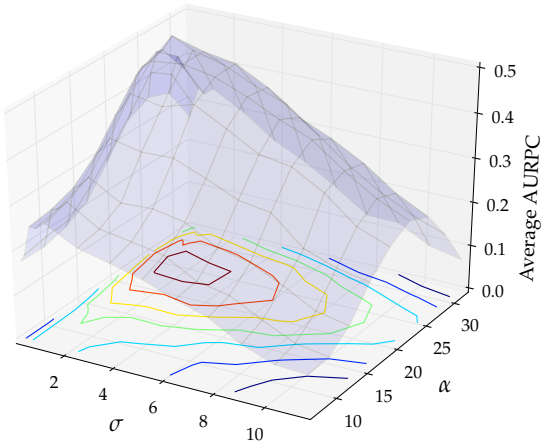


Figure 5.3: Typical scenario when optimizing parameters for a descriptor. In this case, we optimize σ and α for the LOI-GO descriptor and find the best performance around $\sigma = 1.2\,\text{px}$, $\alpha = 26\,\text{px}$. AURPC stands for the area under the recall vs $1 -$ precision curve.

Parameter	**Value(s)**
Tonal scale β for the gradient orientation	0.34
Tonal scale β for the shape index	0.14
Bin center locations B for the gradient orientation	$\{-2.749, -1.964, -1.178, -0.393, 0.393, 1.178, 1.964, 2.749\}$
Bin center locations B for the shape index	$\{-0.875, -0.625, -0.375, -0.125, 0.125, 0.375, 0.625, 0.875\}$
Multi-local sampling points $\boldsymbol{r}$ for '-grid2'	$\{(x,y)\|x,y \in \{20,43\}\})$
Multi-local sampling points $\boldsymbol{r}$ for '-grid4'	$\{(x,y)\|x,y \in \{15,26,37,48\}\})$

Table 5.3: Fixed parameters for LOI and Jet descriptors.

Therefore, we allow ourselves to fixate certain parameters and search through the conditional parameter space spanned by the remaining parameters. The parameters we fixate are $\boldsymbol{r}$ (the locations of the histogram sampling points), β (the tonal aperture), B (the bin intensity levels) and q (the number of bins). For example, q is found to be optimal at 8 for both the gradient orientation histograms and the shape index histograms (this is similar to the findings in [8, 25, 44]). The full list of fixed parameters are listed in Table 5.3.

For each of the proposed LOI descriptors we estimate the parameters σ and α per histogram across multi-scale samplings. For multi-local description we estimate the same σ and α for all histograms in a grid. The estimated parameters are shown in Table 5.1. For the Jet descriptors, we estimate the parameters σ and k and these are listed in Table 5.2.

Admittedly, this optimization scheme is far from perfect. The most critical

points are:

- The performance measure used with the Oxford dataset is the area under the recall vs. $1 - \text{precision}$ curve whereas the measure on the DTU dataset is based on recall vs. fall-out.
- We use only a single scene from the Oxford dataset which increases the risk of overfitting to the training set. To investigate the gravity of this situation we have performed additional sporadic checks on the test set and found that the parameters appear to be close to optimal on the DTU dataset.
- The optimization method does not consider the entire parameter space due to limitations on computational resources.

A more automatic parameter optimization method is proposed in [51] where parameters are learned using conjugate gradient descent on the AUC measures. Though, note that this approach assumes that the performance as a function of the parameters is convex which is not necessarily true.

5.4 Similarity measures

With the feature descriptor representations in place, we now only need a similarity measure between descriptor vectors to be able to discriminate between them. The Euclidean distance between two feature vectors $\boldsymbol{p}, \boldsymbol{q} \in \mathbb{R}^n$,

$$D_{\mathrm{L}^2}(\boldsymbol{p}, \boldsymbol{q}) = \sqrt{\sum_{i=1}^{n} (\boldsymbol{p}_i - \boldsymbol{q}_i)^2} \quad , \tag{5.3}$$

is the de facto standard similarity measure used for feature matching [7,25,30]. In some cases, the Mahalonibis distance is used to take the correlation of the feature vectors into consideration [30]. We can, however, regard the Mahalonibis distance as the Euclidean distance on feature vectors that have been decorrelated beforehand (like the whitening of the Jet descriptor). To our knowledge, the only example of an alternative distance applied to the feature matching problem of the stability-based measure in [4], which is also closely related to the covariance of the descriptor. Furthermore, it has been shown that alternative distance measures can improve performance of SIFT descriptors when used for descriptor clustering in a bag-of-features context [38].

We find the absence of experiments with different distance measures remarkable, especially since most descriptors rely on histograms for which the Euclidean distance is just one out of many distance measures. Therefore, we propose to use the following similarity measures when comparing $\boldsymbol{p}$ and $\boldsymbol{q}$.

L^1 distance The L^1 distance very similar to the Euclidean distance except that all bin differences are weighted the same.

$$D_{\mathrm{L}^1}(\boldsymbol{p}, \boldsymbol{q}) = \sum_{i=1}^{n} |\boldsymbol{p}_i - \boldsymbol{q}_i| \tag{5.4}$$

Thus, D_{L^1} is less sensitive to large bin differences as it does not amplify their contributions by squaring them.

χ^2 distance The χ^2 distance between two histograms is defined by

$$D_{\chi^2}(\boldsymbol{p}, \boldsymbol{q}) = \frac{1}{2} \sum_{i=1}^{n} \frac{(\boldsymbol{p}_i - \boldsymbol{q}_i)^2}{\boldsymbol{p}_i + \boldsymbol{q}_i} \quad . \tag{5.5}$$

Compared to D_{L^1} and D_{L^2} where all bin differences are weighted equally, D_{χ^2} reduces the influence of large bins (where the discrepancy may be disproportionately larger than for smaller bins) which may be favorable in some situations.

Kullback-Leibler divergence The Kullback–Leibler divergence [21] is an asymmetric measure of the difference between two probability distributions

$$D_{\mathrm{KL}}(\boldsymbol{p}||\boldsymbol{q}) = \sum_{i=1}^{n} \boldsymbol{p}_i \ln \frac{\boldsymbol{p}_i}{\boldsymbol{q}_i} \tag{5.6}$$

D_{KL} behaves similar to the χ^2 distance as the logarithm ensures that heavy bins are less significant than bins with little content. We might also say that this gives more importance to the tails of a distribution. Since D_{KL} is asymmetric it is not strictly a distance measure. This is easily circumvented by taking the sum of the divergence in both directions.

$$D_{\mathrm{KLsym}}(\boldsymbol{p}, \boldsymbol{q}) = D_{\mathrm{KL}}(\boldsymbol{p}||\boldsymbol{q}) + D_{\mathrm{KL}}(\boldsymbol{q}||\boldsymbol{p}) \tag{5.7}$$

Jensen-Shannon divergence The Jensen-Shannon divergence [22] is based on the Kullback–Leibler divergence as it computes the total divergence to the mid-point $\boldsymbol{m}$ of $\boldsymbol{p}$ and $\boldsymbol{q}$.

$$D_{\mathrm{JS}}(\boldsymbol{p}, \boldsymbol{q}) = D_{\mathrm{KL}}(\boldsymbol{p}||\boldsymbol{m}) + D_{\mathrm{KL}}(\boldsymbol{q}||\boldsymbol{m}) \quad , \quad \boldsymbol{m} = \frac{1}{2}(\boldsymbol{p} + \boldsymbol{q}) \tag{5.8}$$

A significant difference from D_{KL} is that D_{JS} is always a finite value. Furthermore, if we take the square root of D_{JS}, it satisfies all the properties of a metric [11].

$$D_{\sqrt{\mathrm{JS}}}(\boldsymbol{p}, \boldsymbol{q}) = \sqrt{D_{\mathrm{JS}}(\boldsymbol{p}, \boldsymbol{q})} \tag{5.9}$$

6 Dataset and evaluation criteria

In this section we present the dataset used to evaluate the descriptor performance as well as the evaluation measure.

6.1 DTU robot dataset

The DTU robot dataset[1] [2,7] consists of 60 different scenes containing objects such as miniature buildings, cloth, books, cans and groceries. See Figure 6.1 for examples of the different scene types. Each scene has been photographed systematically with varying configurations of camera position and light source. An overview of the camera and light configuration possibilities is shown in Figure 6.2.

There are 119 different camera positions along four different paths. Three of these are horizontal arcs at different distances to the scene. The fourth camera path is a linear path along the depth axis (z-axis) in front of the scene. For all camera positions the camera is pointed to the center of the scene. Example images from these camera positions are shown in Figure 6.3. Note that there are no vertical variations in the placement of the camera, nor are there any rotations along the optical axis.

The lighting possibilities cover 28 different light source positions as well as diffuse lighting. The 28 light sources are placed along two linear paths following the horizontal axis (the x-axis) and the depth axis (z-axis) respectively. Example images from different light source positions are shown in Figure 6.4.

[1] Dataset and code available at `http://roboimagedata.imm.dtu.dk`.

Figure 6.1: Example images from different scenes in the DTU robot dataset.

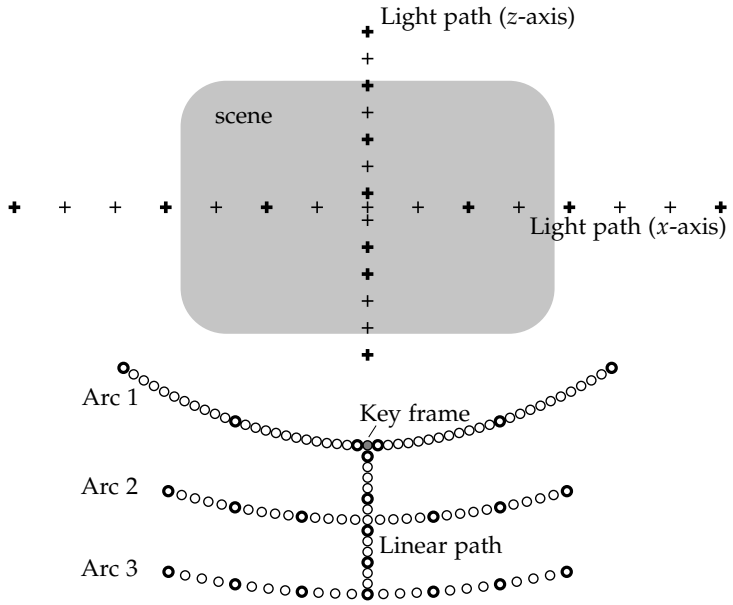


Figure 6.2: Top-down view of all camera and light configurations for the DTU robot dataset. Each scene is photographed from all the different camera positions (circles) and for each camera position, we have different light source positions (plusses). Arc 1 is placed 0.5 m from the scene and spans $\pm 40^\circ$, Arc 2 has distance 0.65 m and spans $\pm 25^\circ$ and Arc 3 has distance 0.8 m and spans $\pm 20^\circ$. The linear path spans the range [0.5 m; 0.8 m]. The circles and plusses with thick strokes indicate that the corresponding camera and light positions are part of the reduced dataset.

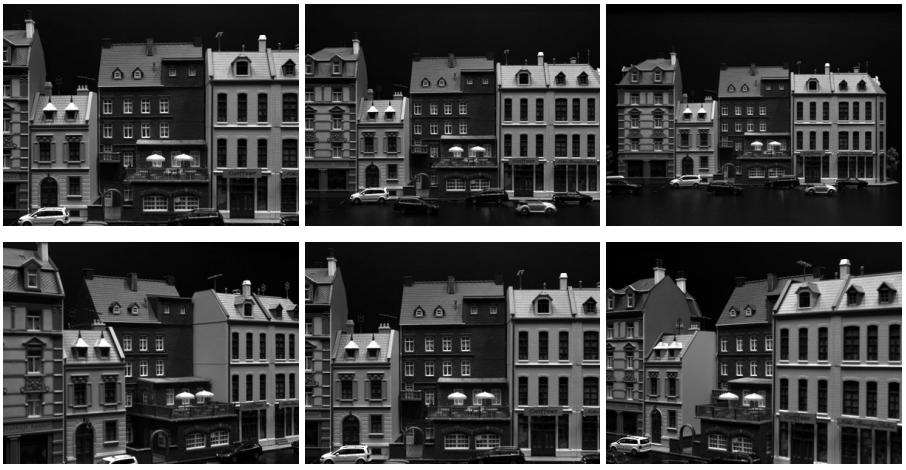


Figure 6.3: Examples images from different camera positions in the DTU robot dataset. The upper row shows the scale variations as the camera moves on the linear path along its optical axis. The lower row shows the view angle variations when moving the camera along Arc 1.

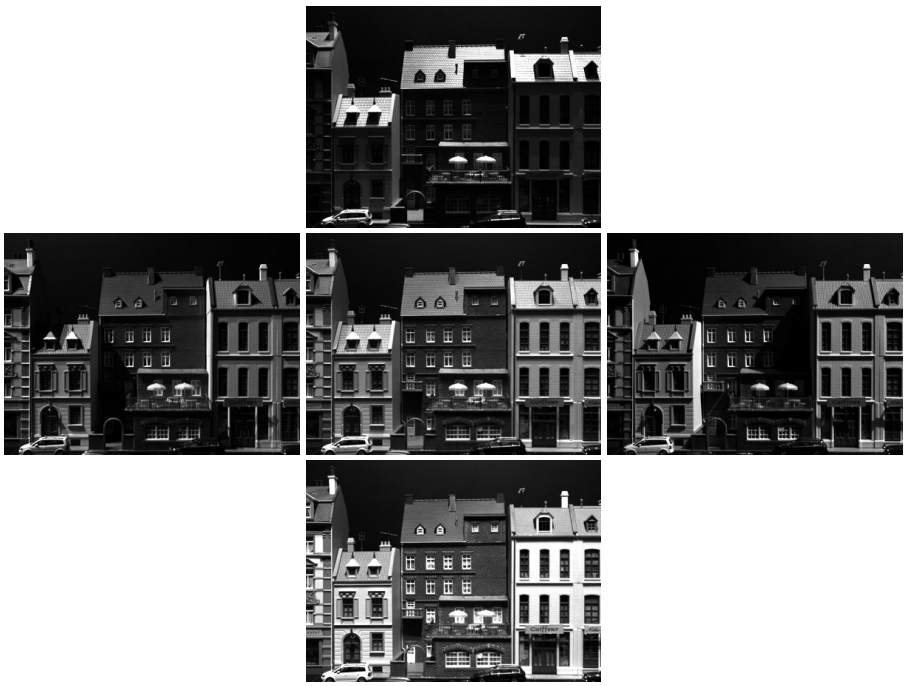


Figure 6.4: Examples of light source positions. The light path along the x is shown horizontally and the light path along the z axes is shown vertically).

Note that the light sources are created artificially from the original dataset as explained in detail in Appendix A.

For every scene, a 3D reconstruction of the surface geometry has been captured from a surface scan using structured light. Since the camera position is known for each image, we can map the 2D image points to the surface of the 3D structure - and vice versa. This is used to determine the true image correspondences allowing us to verify whether 2D image points from different camera positions map to the same location on the 3D model of the scene.

When evaluating descriptor performance, we match the descriptors extracted from a frame against the descriptors extracted from a *key frame* chosen as reference point. For all experiments with camera position variations, the center front position is chosen as key frame (see Figure 6.2) with all images illuminated by diffuse lighting. For all lighting experiments, diffuse lighting is used as reference and the camera position is locked to its reference position.

Because of time constraints in this project, we limit our descriptor evaluation by only using a reduced set of camera and light configurations (23 camera positions and 13 light source positions). These configurations are selected to be representative for the entire dataset, see Figure 6.2. While this is a significant reduction in size compared to the full dataset, the reduced set is still many orders larger than the popular Oxford dataset in terms of number of images (notice that we still have 60 scenes per configuration).

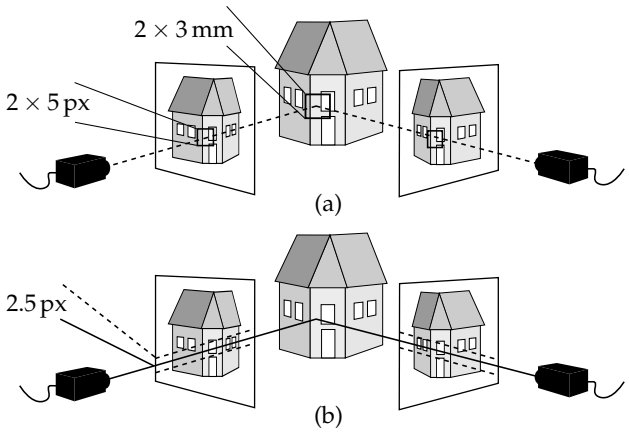


Figure 6.5: The two matching criteria for determining a true feature match. (a) shows that the feature points on the 3D model must be within 3 mm radius of each other (corresponding to approximately 5 pixels in the image). (b) shows the epipolar requirements that require the epipolar line of a feature point to be within 2.5 px of the other feature point. This illustration is reproduced from [7].

6.2 Evaluation criteria

We follow the evaluation method described in [7]. Interest points are extracted from two images and are matched pairwise based on the similarity of their feature descriptors. A match between two features points is correct if the feature points satisfy the following two criteria (illustrated in Figure 6.5). The Euclidean distance between the 3D feature points (mapped from the 2D images onto the 3D structure) must be within a distance of 3 mm. Secondly, the epipolar line of a feature point from one image should be within a 5 px distance to the feature point in the other image.

Recall from Section 2.5 that we perform a ROC analysis based on the *nearest neighbor distance ratio* matching strategy. This means that given two images (one is the reference image), we evaluate the performance of a descriptor from the following. For each feature in the reference image, we go to the other image and find the distances δ_b and δ_s to the *best* matching and the *second best* matching features respectively (in terms of the Euclidean distance between their feature descriptors). We then calculate the ratio $r = \delta_b / \delta_s$ and use it to predict the correctness of a match; if $r > t$, we predict a positive match p', if $r \leq t$ we predict a negative match n'. The prediction is verified against the true value as determined by the previously described matching criteria. Each prediction outcome then falls within one of the four categories: true positive, false positive, true negative and false negative as illustrated by the following 2×2 confusion matrix.

		True value	
		p	n
Prediction	p'	True positive (TP)	False positive (FP)
	n'	False negative (FN)	True negative (TN)

The number of outcomes in each category is counted, and we can calculate the true positive fraction (TPR) and the false positive fraction (FPR) from

$$\text{TPR} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad , \quad \text{FPR} = \frac{\text{FP}}{\text{FP} + \text{TN}} \tag{6.1}$$

TPR is also known as the *recall* and it tells us how good we are at identifying positive feature matches correctly. FPR is known as the *fall-out* indicating the ability to identify negative feature matches correctly. Recollect that these are the same terms as mentioned in the review of different evaluation criteria in Section 2.5. By varying the value of the threshold t and computing the TPR and the FPR for each threshold, we can generate the *receiver operating characteristic* (ROC) curve [12]. The ROC curve shows the tradeoff between true positives and false positives as we adjust our threshold criteria t.

In order to get a single value measure to quantify the descriptor performance on an image-pair, we compute the *area under the curve* (AUC) using the trapezoidal approximation method. Perfect descriptor performance is achieved when AUC $= 1$ while descriptor performance similar to a random guess gives AUC $= 0.5$. For each configuration of camera position / lighting we compute the mean AUC over the 60 scenes and use it to quantify the descriptor performance.

6.3 Sensitivity analysis

To complement the above evaluation method, we propose a sensitivity analysis of the image descriptors. The goal of the sensitivity analysis is to study in detail how different image perturbations affect the local feature description for different descriptors. Therefore, we try to limit as many influencing factors as possible in an attempt to better reveal the sensitivity of the descriptors when exposing the image patches to a single perturbation factor. The main differences between the sensitivity analysis and the feature matching evaluation are:

- Instead of measuring descriptor performance using the ROC analysis, we measure the discrepancy between the perturbed feature description versus the unperturbed reference feature description. The discrepancy measure is described in detail below.
- Instead of finding a different set of interest points for each image, we find a set of interest points in the reference image and map them to the other images using the 3D scene model (more on this below). Thereby, we ensure that corresponding interest points from different images point at exactly the same location on the 3D model.

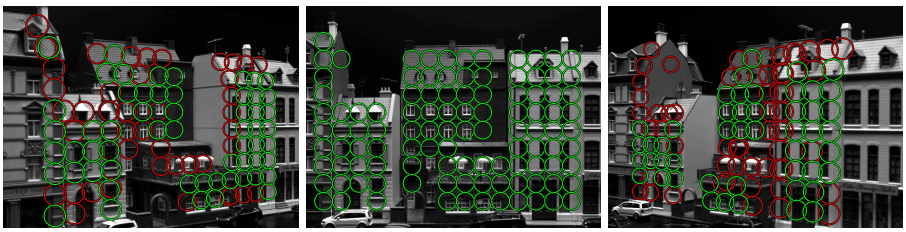


Figure 6.6: Feature point transformations for the sensitivity analysis. The center image shows the reference image from which the interest points are generated. Green circles indicate valid interest points while red circles indicate invalid interest points (due to an unwanted perturbation, e.g. occlusion).

- We do not use an interest point detector to generate interest points. Instead, we extract interest points in a grid resulting in an arbitrary sampling of image patches. This ensures that our results are not biased by the detector.
- We disregard interest points that are subject to unwanted perturbations, e.g. occlusions. This gives a more optimistic view on the descriptor performance compared to what we can expect in practice. Again, we do this to reduce the effect of unwanted nuisance factors to give a more pure view of descriptor sensitivity when perturbing the images.

Generating interest points

Interest points are generated only for the reference image using an arbitrary grid sampling. We want to fixate these interest points to make them point at the same locations on the 3D scene model across different images. For images where the camera position is stationary, fixating the interest points is trivially done using the same set of interest points for all images. However, when the camera position changes, we have to adjust the location and the scale of the interest points. See Figure 6.6 for an example of these interest point transformations as the view angle changes.

To map interest points between images, we project all interest points from one image onto the 3D scene model using the camera transformation matrix of that image. Hereafter, these 3D points are mapped to the other image using the camera transformation for that image. Finally, we adjust the scale of the interest point using the ratio between the camera distances to the 3D point. Examples of the interest points that we generate using this method are shown in Figure 6.7.

Since we want to minimize the influence of irrelevant factors, we prune the interest points on two occasions. Firstly, we remove interest points in the reference image at unsuitable locations in the scene, i.e. interest points pointing at scene elements for which we have little or no 3D structure available. This is done by attempting to map the interest points onto the 3D structure. If no or very little 3D structure is available for an interest point (that is, if the point cloud that make up the 3D structure is sparse around the interest point), it is discarded. Secondly, we discard interest points mapped to other images

Figure 6.7: Example of image patch perturbations caused by change in view angle (top row) and lighting source (bottom row). The image patch is taken from the scene shown in Figure 6.6.

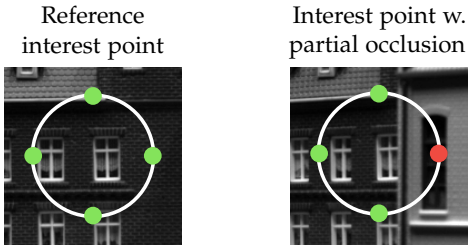


Figure 6.8: Interest point pruning for the sensitivity analysis. To the left, we have placed an interest point in the reference image. To the right, the same interest point has been mapped to an image taken from a different view angle. We see in the right image that the interest point has become partly occluded. The green and red dots indicate points that are mapped to the 3D structure to check if corresponding points are in accordance (by measuring their Euclidean distance in 3D). The red dot fails this check causing the interest point to be discarded for that view angle.

if they have become partly occluded (e.g. as a result of a view angle change). Occlusions are detected by mapping the left-, right-, upper- and lowermost point on the circular interest point border to the 3D model and checking if they each are within a certain distance of their corresponing points from the reference interest point mapped to 3D. The second pruning is explained further in Figure 6.8. Both pruning steps can also be observed in Figure 6.6, where the missing interest points in the grid are caused by the first pruning and where the red interest points indicate discarding from the second pruning.

For the sensitivity analysis, we extract interest points at three different scales ending up with approximately 225 interest points per scene (after the first pruning). Around 10-20% of these points are discarded by the second pruning. Thus, we extract around 180 interest points for each camera position in a scene.

Evaluation criteria

From the interest points above, we extract image patches for all images and generate feature descriptors. We then measure the discrepancy between the descriptor vectors from the reference image and their corresponding descriptor vectors from the perturbed images using the Euclidean distance. However, this discrepancy measure alone is not sufficient to reason about descriptor sensitivity for two reasons. It does not tell us anything about the discriminative

ability of a descriptor since it does not reflect the feature vectors ability to stand out from the feature vectors extracted from the image. Furthermore, the discrepancy measure does not allow us to compare the sensitivity between different descriptors since descriptor distances have different units.

To provide an evaluation criteria that reflects the discriminative ability of a descriptor and allows for comparisons between descriptors, we propose the following. For each patch description in a perturbed image we measure its distance d_r to the reference patch description. Furthermore, we measure its distances d_o to all other patches extracted from the image. We repeat this for all scenes for a fixed camera / light configuration and end up with two distributions of distances; one distribution $p(d_r)$ of distances to reference feature descriptors, and one distribution $p(d_o)$ of distances to all other descriptors in the image. The descriptor sensitivity is estimated by performing Welch's t-test,

$$t = \frac{\overline{X}_r - \overline{X}_o}{\sqrt{\frac{s_r^2}{N_r} + \frac{s_o^2}{N_o}}} \quad , \tag{6.2}$$

where $\overline{X}$, s^2, and N represent the sample mean, sample variance and sample size of $p(d_r)$ and $p(d_o)$ respectively as indicated by the subscripts. Welch's t-test assumes that both $p(d_r)$ and $p(d_o)$ follow the Gaussian distribution, but it does not assume they have the same standard deviation. The t value expresses the discrepancy between the two distributions and thereby the discriminative ability of a descriptor.

Note that we could also use t above to perform a null hypothesis test to see if e.g. the mean of $p(d_o)$ is greater than $p(d_r)$. In practice, however, this would almost always be the case since there is a significant difference between the two distributions.

7 Descriptor evaluation

In this section, we will evaluate the performance of our descriptor proposals from Section 5 on the dataset presented in Section 6. Firstly, we evaluate the performance of our LOI and Jet descriptors and investigate what descriptor configurations works best for each them. Secondly, we perform a comparative study of our descriptor proposals with other state-of-the-art descriptors from the literature. Finally, we study the descriptor performances using our sensitivity analysis.

Note that we perform most of the following benchmarks using interest points from the *Maximally stable extremal regions* (MSER) detector [27] since it performs [2,7,45]. However, we will also investigate the influence of different detectors on descriptor performance in Section 7.3.

7.1 LOI descriptor

The performance of our LOI descriptor proposals (listed in Table 5.1) is shown in Figure 7.1. In general, we see that the ordering of the descriptors is very consistent across all configurations of camera and light source position. Thus, we do not need to consider performance with respect to a certain perturbation scenario in the following.

Multi-scale description

We see that the low-dimensional multi-scale descriptors (LOI-GO, LOI-GO_2, LOI-SI, LOI-SI_2 and LOI-GOSI) do not achieve good performance compared to the other LOI descriptor variants. When increasing the dimensionality of the multi-scale descriptors, performance is significantly ameliorated (see LOI-GO_4, LOI-GO_2SI_2, LOI-GO_2SI_4, LOI-GO_4SI_2 and LOI-GO_4SI_4). The multi-scale gradient orientation histograms are clearly more discriminative than the multi-scale shape index histograms (observe LOI-GO_4 vs. LOI-SI_4 and LOI-GO_4SI_2 vs. LOI-GO_2SI_4). In fact, it seems that multi-scale shape histograms gathered at more than two different scales do not capture additional image structure since LOI-SI_2 and LOI-GO_4SI_2 achieve approximately the same performance as LOI-SI_4 and LOI-GO_4SI_4 respectively.

Multi-local description

Regarding the multi-local LOI variants, we see that they are able to achieve higher performance than the multi-scale descriptors. Our hypothesis, that multi-locality can be (partly) substituted with the second order information captured by the shape index, seems to hold to some degree as we observe a similar performance between LOI-GOSI-grid2 and LOI-GO-grid4 (the SIFT-like descriptor). Notice also the large difference in descriptor dimensionality,

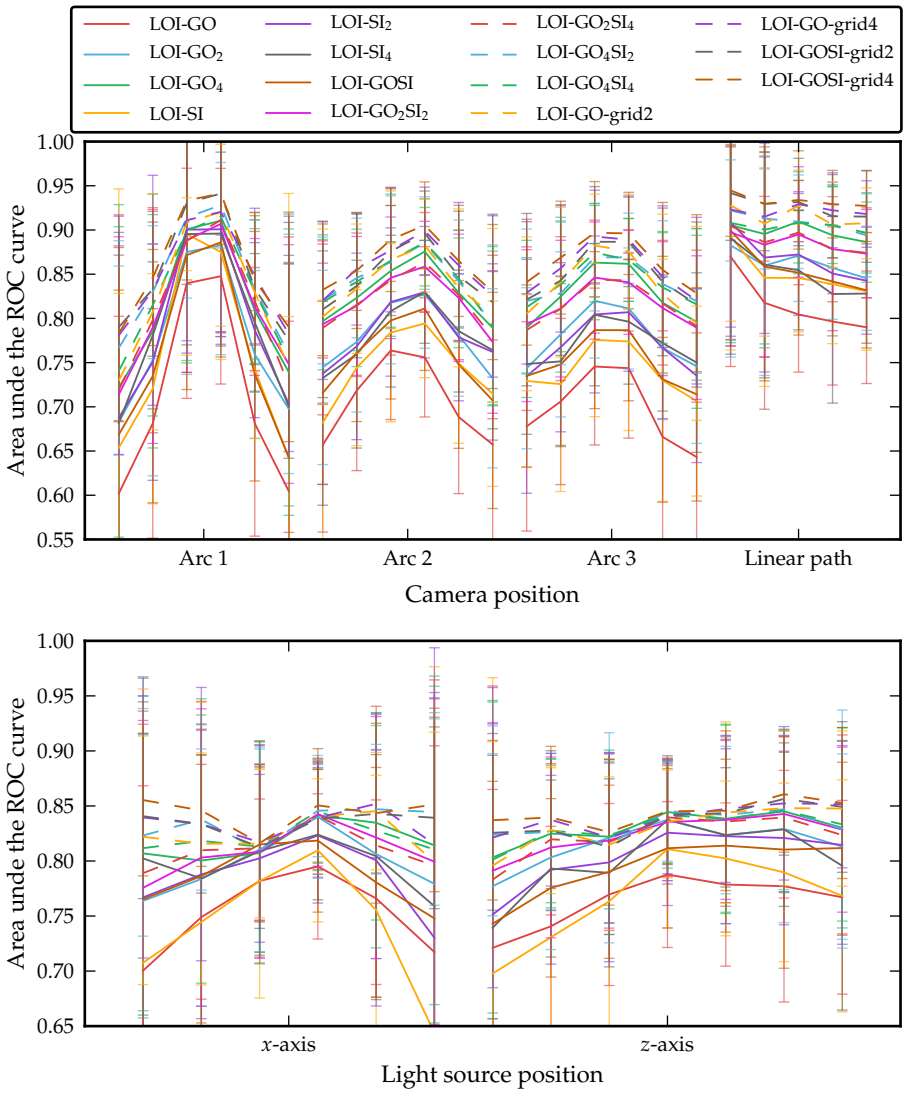


Figure 7.1: Performance of the different configurations of the LOI descriptors (from Table 5.1) on the DTU dataset. Interest points are generated using the MSER detector. The error bars indicate the standard deviation.

64 vs. 128. However, we cannot completely do away with multi-local description since our best performing multi-scale descriptor (LOI-GO_4SI_4) is only able to tangent the worst performing multi-local descriptor (LOI-GO-grid2).

SIFT's additional Gaussian window

As described in 4.1, one of the main differences between the LOI and the SIFT descriptor is that for SIFT, an additional Gaussian window is used to weight to the magnitude of the gradient orientation. We wish to investigate

whether this has any influence on the discriminative abilities of our LOI descriptor. Figure 7.2 shows the performance for the LOI-GO-grid4 descriptor (our SIFT approximation) with different Gaussian window scales γ. We see that the additional Gaussian window has no visible effect on the matching performance; except that it degrades the performance when the Gaussian spread γ becomes too narrow. This is an interesting result that contradicts Lowe's claims in [25] (which Lowe never supports by any empirical results). The result corresponds reasonably well with the findings in [8] where the Gaussian window yields an improvement of only 1% for the HOG descriptor used in a person detection system.

Normalization

We want to investigate which of the following normalization strategies perform best: L^1, L^2, and thresholded L^2 normalization. Moreover, we wish to investigate whether it is beneficial to normalize the histograms separately vs. normalizing the entire feature vector.

We have experimented with the different normalization schemes on LOI-GO-grid4 and the LOI-GO_4SI_2, and the results are shown in Figure 7.3. For both descriptors, it is clear that some normalization scheme should be used since the unnormalized descriptors perform worse than the normalized, especially for changes in illumination (which makes good sense as the contrast level is highly dependent on the illumination).

For the SIFT-like descriptor, LOI-GO-grid4, we see that the choice of normalization method is indifferent since all normalized descriptors perform approximately the same. We also see that the use of thresholding has no effect on the descriptor performance. When the threshold is lowered to 0.1, the performance drops since the clipping is too aggressive. Thus, we are not able to support Lowe's claims in [25], where he states that 0.2 yields the best performance.

For LOI-GO_4SI_2, we see that the histograms should be normalized individually and not by normalizing the entire feature vector. This makes sense because deflections of a gradient orientation histogram should not affect a shape index histograms in the same descriptor. We do not experiment with thresholded normalization on LOI-GO_4SI_2 since it would require different threshold values per histogram type, and since it does not look promising for LOI-GO-grid4.

7.2 Jet descriptor

The performance of our Jet descriptor proposals (listed in Table 5.2) is shown in Figure 7.4. For the single-scale jets, we see that the best performance is achieved around Jet-6 and Jet-7. The lower order jets, Jet-3 and Jet-4, yield subpar performance compared to the other descriptors. This is somewhat unexpected since no other local image descriptors in the literature have relied on image derivatives higher than the third order. Thus, we had expected higher-order structure above the third order to be inefficient at describing local geometry.

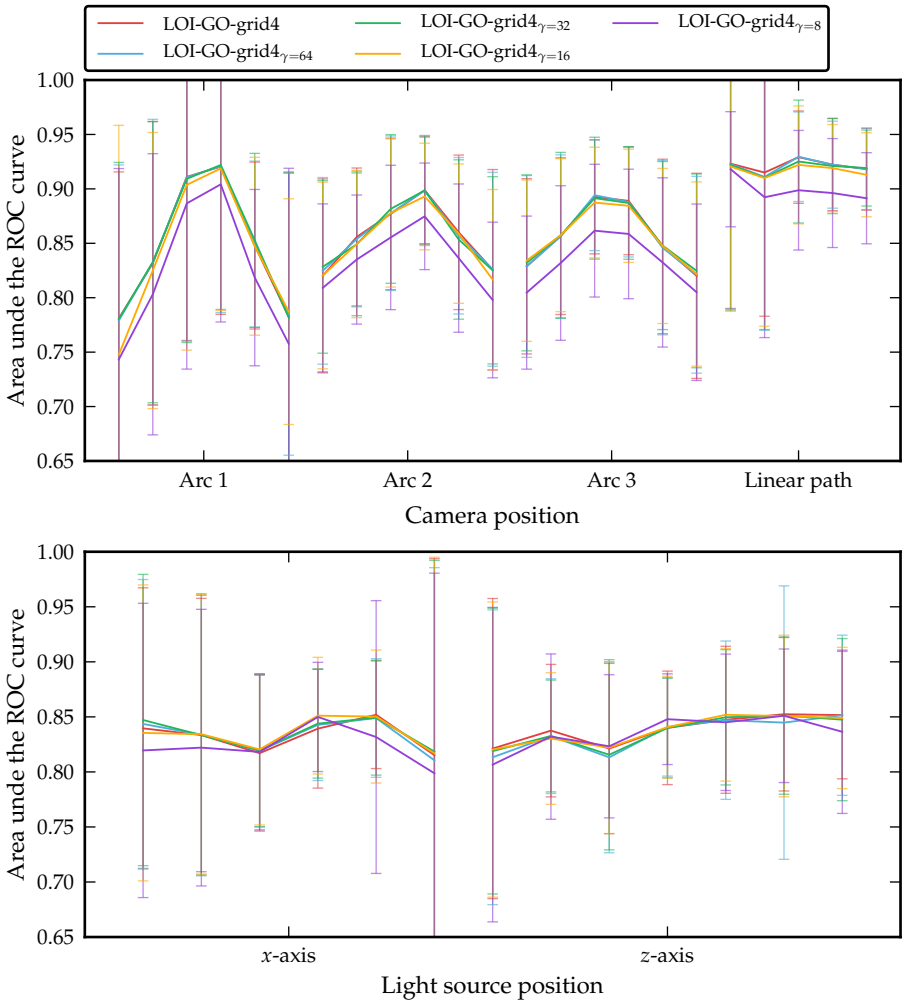


Figure 7.2: Performance of the LOI-GO-grid4 descriptor with an additional Gaussian window on the gradient magnitudes to imitate the SIFT descriptor. The plain LOI-GO-grid4 descriptor has $\gamma = \infty$. We see that the Gaussian window has very little influence on the descriptor performance. When the Gaussian window becomes too narrow ($\gamma = 8\,\text{px}$), the performance drops.

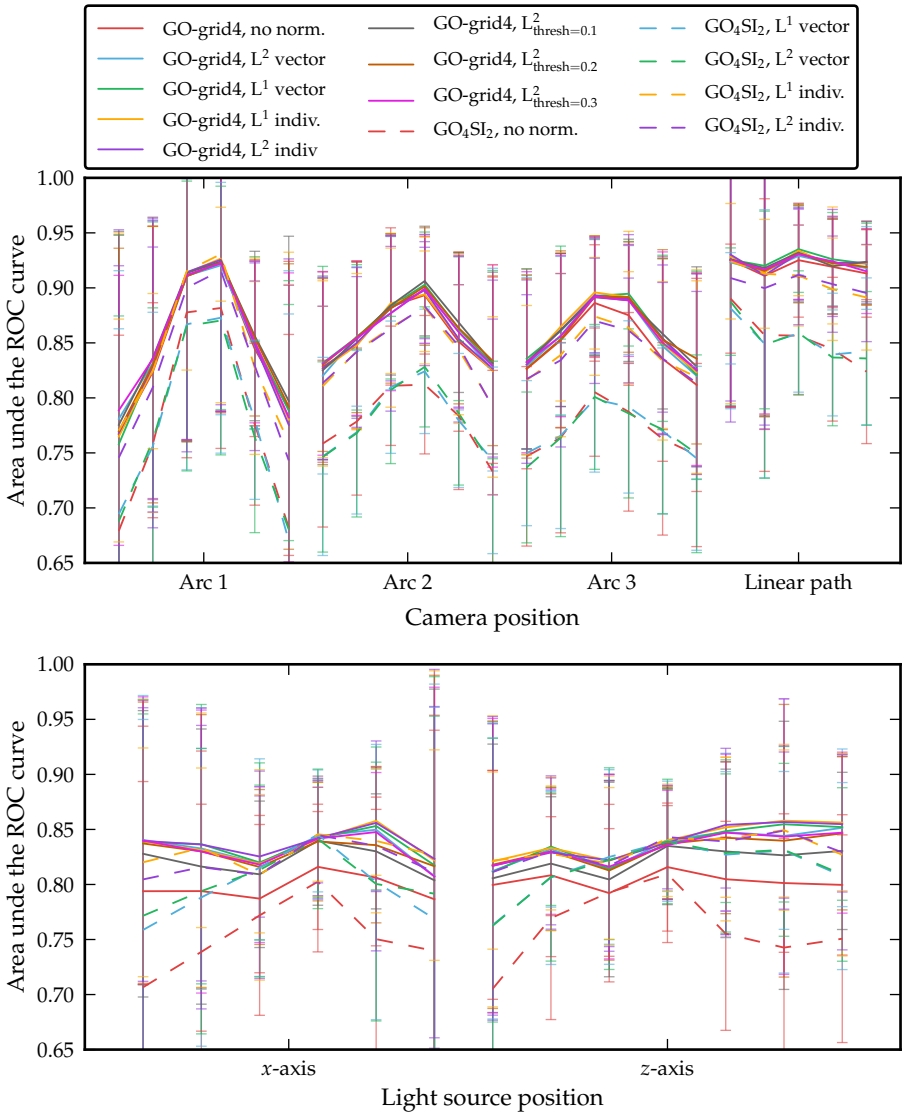


Figure 7.3: The performance implications of different normalization schemes on the LOI-GO-grid4 and the LOI-GO_4SI_2 respectively. 'no norm' indicate that the descriptors are used without normalization. 'L^p vector' indicate that the entire feature vector has been normalized using the L^p norm. 'L^p indiv.' indicate that each histogram has been normalized separately using the L^p norm. Finally, '$L^2_{\text{thresh}=x}$' indicate thresholded L^2 norm with x as threshold value.

Multi-scale description

Combining Jet descriptors at different scales gives limited improvement in performance when we compare Jet-4 and Jet-5 to Jet-4_2 and Jet-5_2 respectively. It seems that the multi-scale descriptors are slightly more discriminative than than single-scale descriptors when the camera position changes. However, the multi-scale approach fails to make the descriptor more robust for illumination variations since the performance is no better than for the single-scale descriptors.

Multi-local description

Like for the LOI descriptors, multi-locality is able to improve the Jet descriptor performance. Compared to both single and multi-scale descriptors, the multi-local descriptors show some improvement for camera position perturbations. The multi-local descriptors show a visibly better robustness towards illumination perturbations. An explanation of this behavior could be that illumination changes may not affect all the multi-local sampling points which leads to a smaller change of the entire descriptor since the non-affected sampling points remain the same.

With the 2×2 grid, the best performance is achieved using local 4-jets. Interestingly, Jet-4-grid2 achieves a performance similar to Jet-3-grid4, which shows that we are able to substitute some multi-locality with higher-order image structure.

Normalization

The influence of different normalization methods is similar to the influence on our LOI descriptors. That is, descriptor normalization increases performance significantly, and the choice between L^1 and L^2 normalization is indifferent as they both yield the same results. For this reason, we do not show any performance graphs for the different normalization schemes on the Jet descriptor.

7.3 Comparison with other descriptors

In the following we evaluate our Jet and LOI descriptors against state-of-the-art descriptors from the literature. These descriptors (SIFT, PCA-SIFT, GLOH, SURF and moment invariants, see Section 2.4 for an overview) are generated using the code made available by the Oxford vision group[1] and from the official website of SURF[2]. We disable rotation invariance for all descriptors to make them comparable to our descriptors. To avoid cluttering the performance graphs, we select a total of six descriptor variants to represent our descriptor proposals. These descriptor variants are chosen to represent the Jet and LOI descriptors at their best performance and at their best performance when descriptor dimensionality is taken into consideration. We still want

[1] http://www.robots.ox.ac.uk/~vgg/research/affine
[2] http://www.vision.ee.ethz.ch/~surf

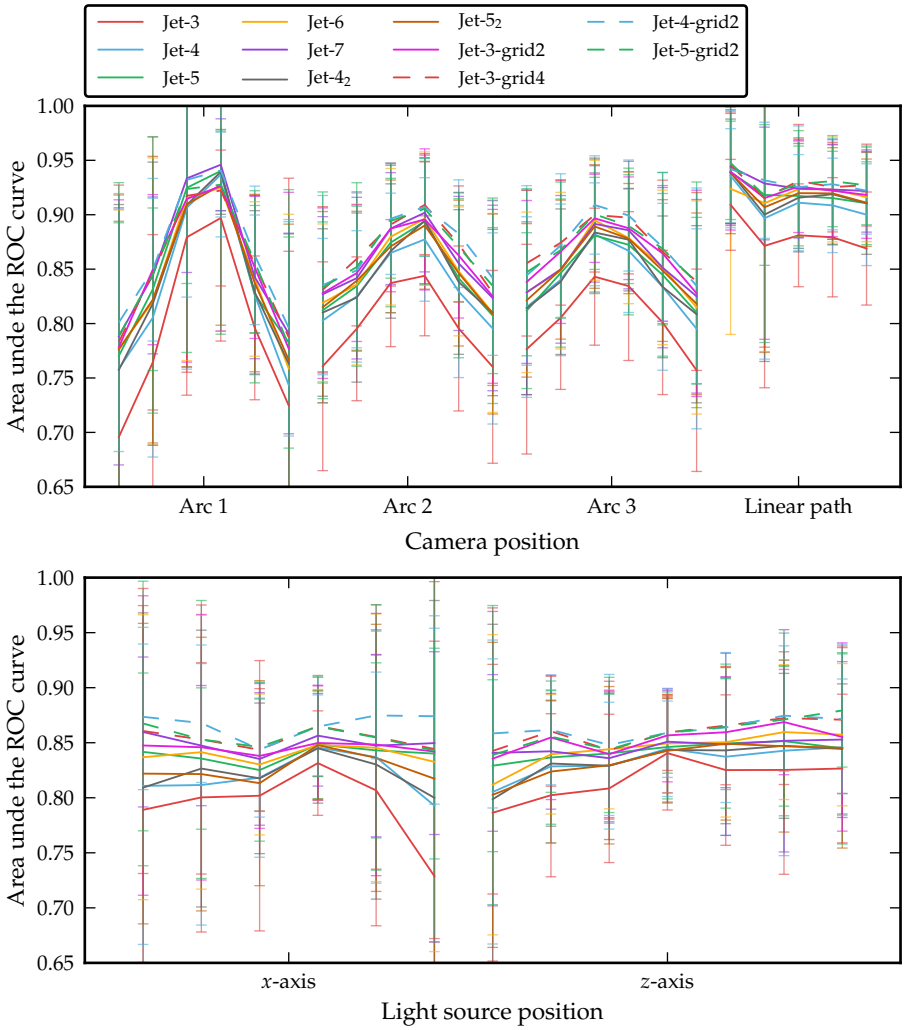


Figure 7.4: Performance of different configurations of the Jet descriptor on the DTU dataset. Interest points are generated using the MSER detector.

Descriptor	Dimensionality
SIFT [25]	128
GLOH [30]	128
SURF [5]	64
PCA-SIFT [17]	36
Moment inv. [32]	20
LOI-GO_4SI_2	48
LOI-GOSI-grid2	64
LOI-GOSI-grid4	256
Jet-5	20
Jet-7	35
Jet-4-grid2	56

Table 7.1: The different descriptors that we use for our comparative performance study.

to investigate if good matching performance can be achieved using a low dimensional descriptor. The complete list of descriptors for our comparative study is shown in Table 7.1.

In Figure 7.5, the performance of the descriptors is shown. We see that top performers are PCA-SIFT, SIFT, GLOH and Jet-4-grid2. PCA-SIFT stand out by showing good robustness towards both camera position and illumination changes. The Jet-4-grid2 performance is slightly under par for camera position changes, however, for illumination perturbations it is significantly better than SIFT and GLOH. Thus, our hypothesis that multi-locality can be partly substituted with higher-order information seems to hold. Considering the dimensionality of SIFT and GLOH, Jet-4-grid2 is able to achieve competitive performance with under half the descriptor size.

The performance for both SURF and the moment invariant descriptor are below average. The SURF descriptor is especially sensitive towards differences in lighting. This is somewhat surprising compared with the results in [5] where the descriptor is shown to outperform SIFT, GLOH and PCA-SIFT on images from the Oxford dataset (using a different detector, though). Both SURF and the moment invariant descriptors are relatively low-dimensional at 64 and 20 dimensions respectively. Compared to these, Jet-5 and Jet-7 are performing quite well at a comparable dimensionality.

Our LOI descriptors fail to compete against the top performing descriptors since LOI-GO_4SI_2, LOI-GOSI-grid2 and LOI-GOSI-grid4 all show mediocre results. These results are unexpected since the descriptors are very similar to SIFT. We have previously shown that none of SIFT's extra features (thresholded L^2 normalization and Gaussian window weighting of gradient magnitudes) yield no performance improvements. Thus, we rule out these conditions as the reason for the difference in performance. A more likely explanation could be that the LOI descriptor parameters have been overfitted to the very small Oxford dataset as described in Section 5.3. We remark that a similar subpar performance of DAISY descriptors is found in [7] when evaluated on the same

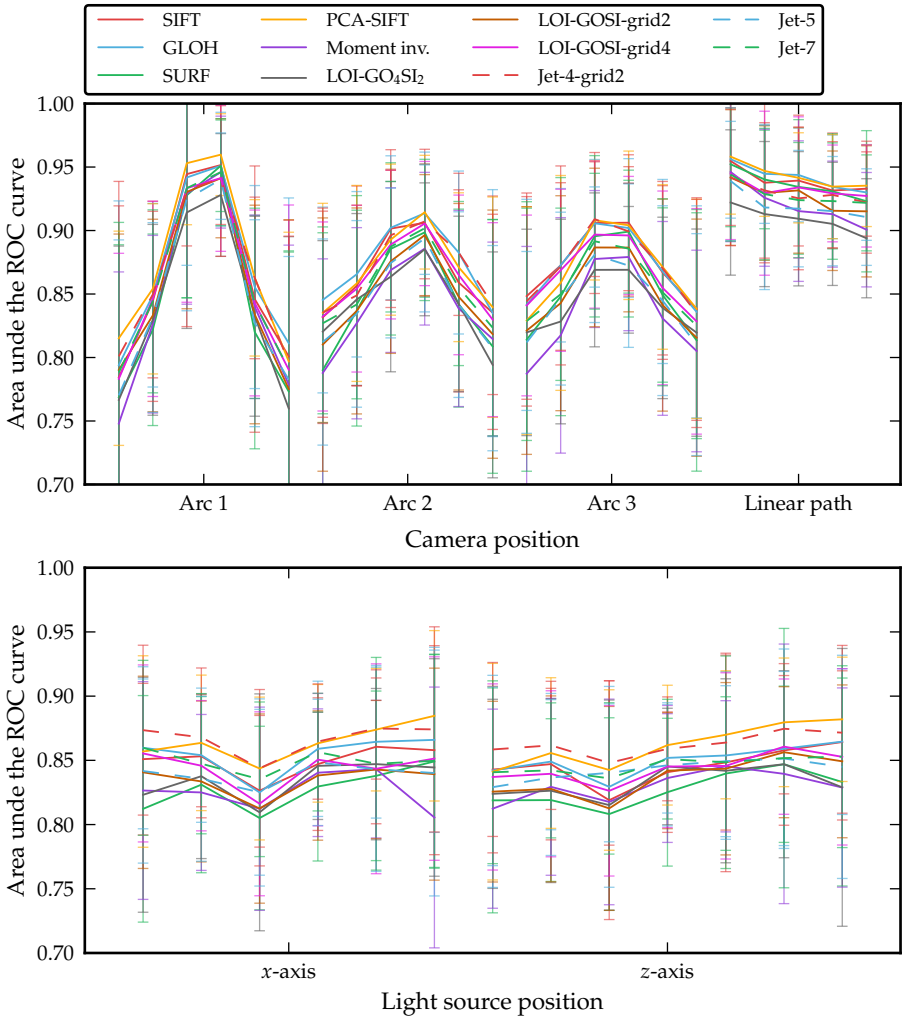


Figure 7.5: Performance of selected LOI and Jet descriptors compared to other state-of-the-art descriptors on MSER interest points.

dataset against SIFT. Since the DAISY descriptor is also similar to SIFT on many points, this leads us to believe that histogram-based descriptors with their relatively large number of parameters are hard to configure properly.

Influence of interest point detectors

So far we have only considered descriptor performance on interest points generated by the MSER detector. While the MSER is considered a good detector [7,45], it is not representative for the many other detectors available in the literature. Therefore, we choose to evaluate the descriptors using the *Difference of Gaussians* (DoG) [25], the *Harris-Laplace* (HarLap) and the *Harris-affine* (HarAff) [29] detectors. They are all reported to work well in [7,31].

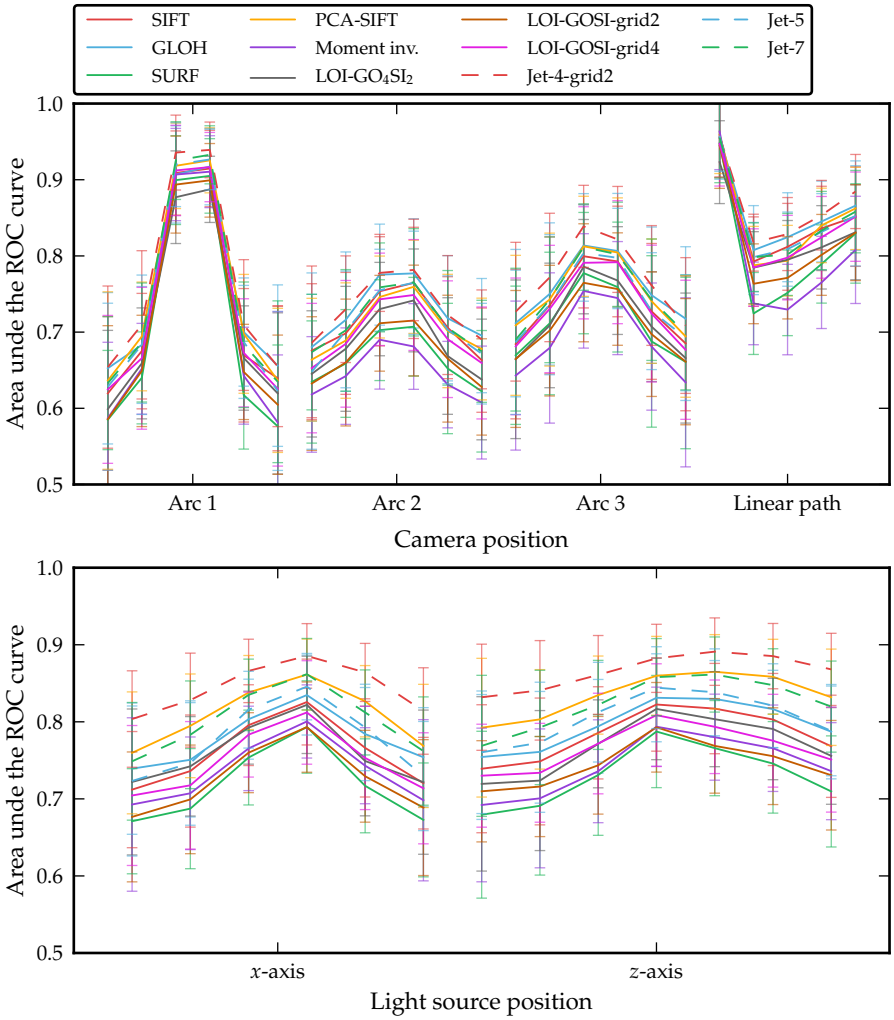


Figure 7.6: Descriptor performance on DOG interest points.

DoG In Figure 7.6 we see the descriptor performance using the DoG detector. The ordering of the descriptors has changed such that Jet-4-grid2 descriptor shows a significant advantage over all the other descriptors, especially under illumination perturbations. We also see that the plain Jet descriptors are able to achieve a slightly better performance than both SIFT and GLOH. This is quite impressive with the dimensionality of Jet-5 and Jet-7 taken into consideration.

HarLap In Figure 7.7, descriptor performance is shown for the HarLap detector. We are beginning to recognize a pattern as LOI, SURF and the moment invariant descriptors consequently are the worst performers. Concerning the SURF descriptor, this result contradicts the performance reported in [5]. PCA-SIFT, SIFT, GLOH and Jet-4-grid2 are all top performers and we note that the Jet-4-grid again show superior robustness to illumination perturbations.

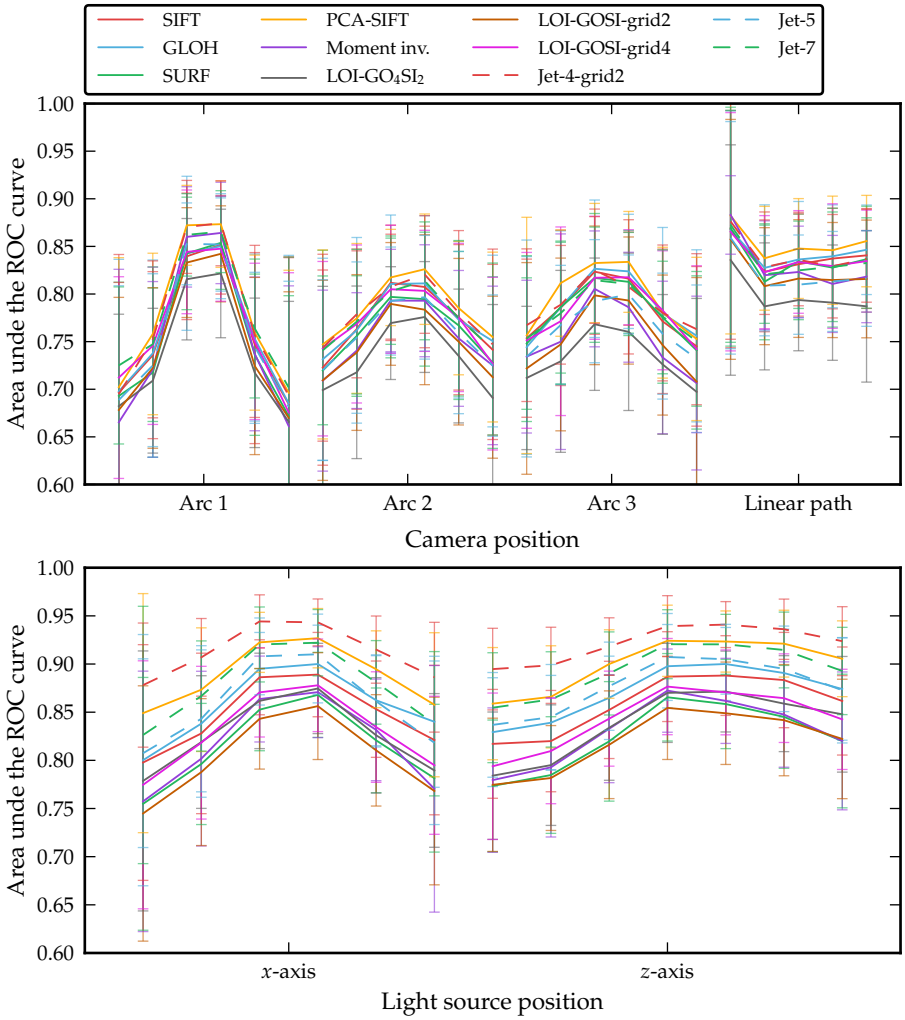


Figure 7.7: Descriptor performance on HarLap interest points.

For the HarLap detector, Jet-7 performs on par with SIFT and GLOH.

HarAff Finally, in Figure 7.8 we see the descriptor performance using the HarAff detector. The pattern is much similar to that of HarLap interest points with PCA-SIFT, SIFT, GLOH, Jet-4-grid2 achieving the best performance.

Similarity measures

In Section 5.4, we discussed the lack of experimentation with different distance measures and presented alternatives to the standard Euclidean distance. We have evaluated descriptor performance of SIFT and LOI-GO_4SI_2 using these distance measures. Note that for the distance measures D_{χ^2}, D_{KL}, D_{KLsym}, D_{JS} and $D_{\sqrt{\mathrm{JS}}}$, we perform a L^1 normalization of the individual histograms in the descriptor vectors to match the assumptions of the distance measures.

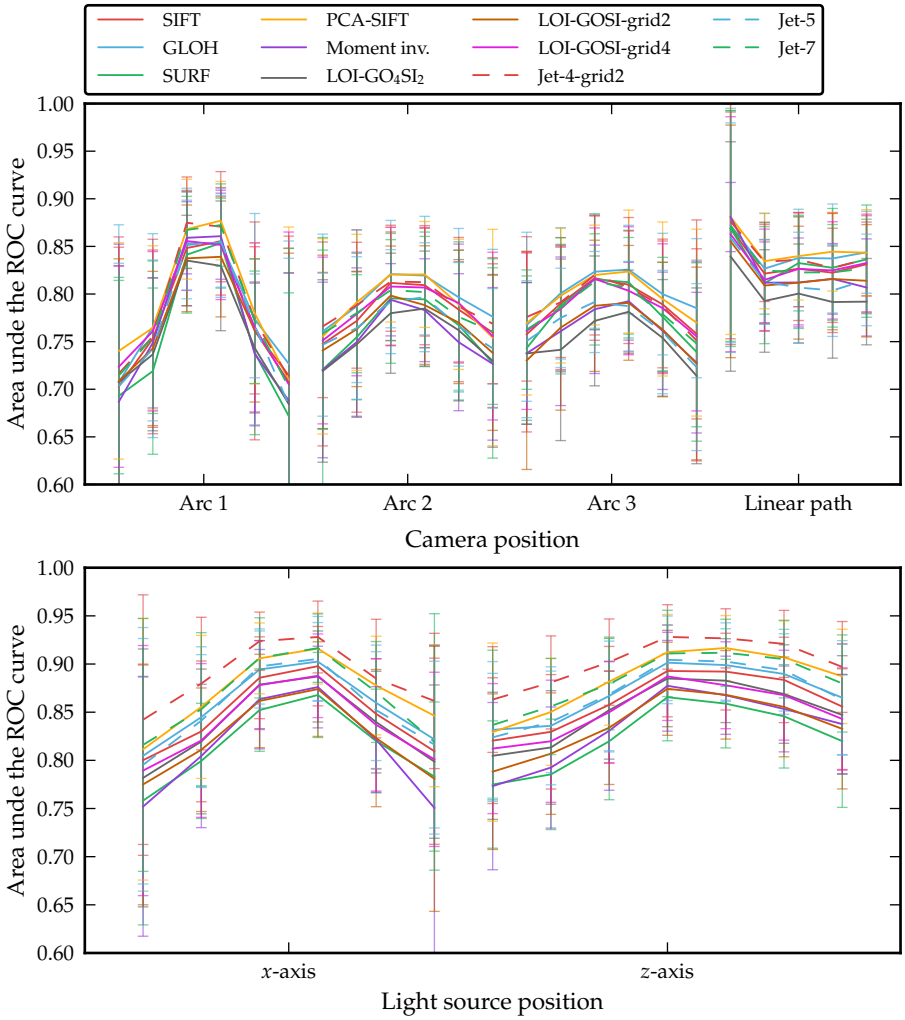


Figure 7.8: Descriptor performance on HarAff interest points.

The results are shown in Figure 7.9. We see immediately, that there is no difference in performance when using the other distance measures since no single distance measure has a visible advantage in any of the perturbation scenarios. Therefore, we do not explore alternative distance measures further in this project.

Planar vs. non-planar surfaces

As the DTU dataset contains different scene types, we will examine to what extent descriptor performance is influenced by the scene content. We limit our investigation to consider only the difference between planar and non-planar surfaces. To represent planar surfaces, we use a total of 17 scenes containing miniature houses, books and building materials (wooden planks and bricks). To represent non-planar surfaces, we use a total of 22 scenes containing fabric,

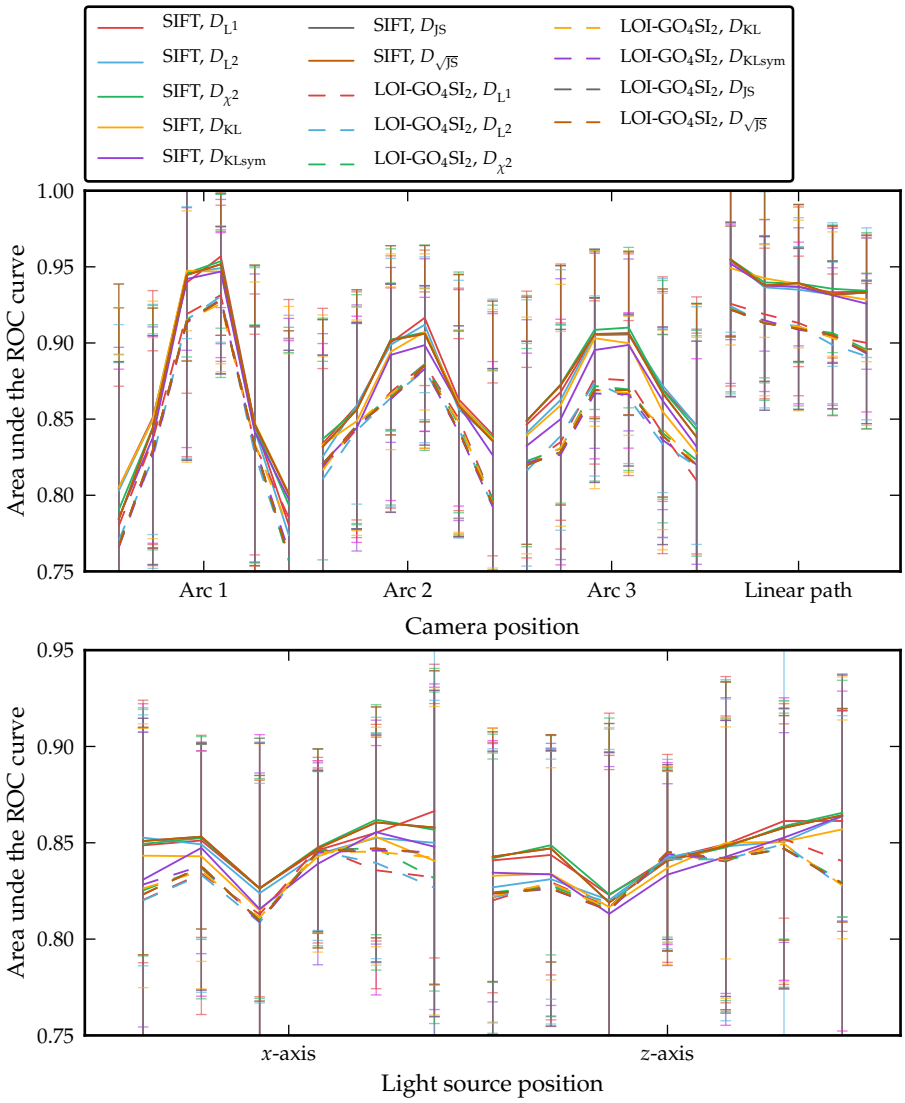


Figure 7.9: Alternative distance measures for evaluating SIFT and LOI-GO$_4$SI$_2$ on interest points generated by the MSER detector.

plush toys, vegetables and beer cans. We evaluate the descriptor performance using DoG and MSER interest points and the results are shown in Figure 7.10. The evaluation results for illumination changes have been omitted to save space and since they are similar to the camera displacement results.

We see a notable difference in descriptor performance depending on the scene type. Non-planar scenes are more difficult for feature matching as we see that the AUC is generally lower than for planar scenes. It seems that the Jet descriptors are slightly more robust to non-planar scenes as their performance decreases slightly less than the other descriptors (especially for large view angle changes). We speculate that it is an advantage not to rely on too many small multi-local sampling points for non-planar structures since the outermost sampling points may be exposed to heavy perturbations.

Recall vs. $1-$precision evaluation measure

As described in Section 2.5, the choice of descriptor evaluation measure has been debated in the literature. For the DTU dataset, the ROC analysis (recall vs. fall-out) has been employed. However, for the majority descriptor evaluations, the recall vs. $1-$precision is used. We wish to investigate the importance of choosing one measure over the other.

In Figure 7.11 and 7.12, we see recall vs. $1-$precision descriptor performance on MSER and DoG interest points respectively. The figures should be compared to Figure 7.5 and 7.6 showing recall vs. fall-out performance. The difference between the two measures is quite clear. We see that the recall vs. $1-$precision measure yields significantly more consistent results as the variance is visibly lower and the graphs more smooth. As the camera distance from the scene increases, we also observe that the descriptor performance drops significantly for $1-$precision when compared to fall-out. This is because the detectors produce fewer interest points as the camera moves away from the scene. Because there are fewer interest points, the fall-out ratio is improved as the number of negative occurrences drop (remember that the fall-out is given as # false positives / total # of negatives). The lower number of detected interest points does not improve the $1-$precision measure because it is computed as the ratio of false positives vs. the total number of matches predicted. Thus, the number of detected interest points has less influence on the $1-$precision measure making it more representative when reasoning about descriptor performance in situations where the number of interest points changes. We observe the same effect for changes in illumination where the number of interest points also drops as the light source is moved away from the center position (see Figure 7.5 vs. Figure 7.11). From the fall-out, it is not completely clear that the discriminative performance of the descriptors goes down as the light source changes since the number of interest points drops. From the $1-$precision measure, we clearly see that the interest points are perturbed and that it gets harder to discriminate between them.

Finally, we note that the ordering of the different descriptors only changes slightly between the two evaluation measures. For example, PCA-SIFT does not stand out as the best performer for all camera position perturbations anymore on MSER interest points. In general, though, the change in de-

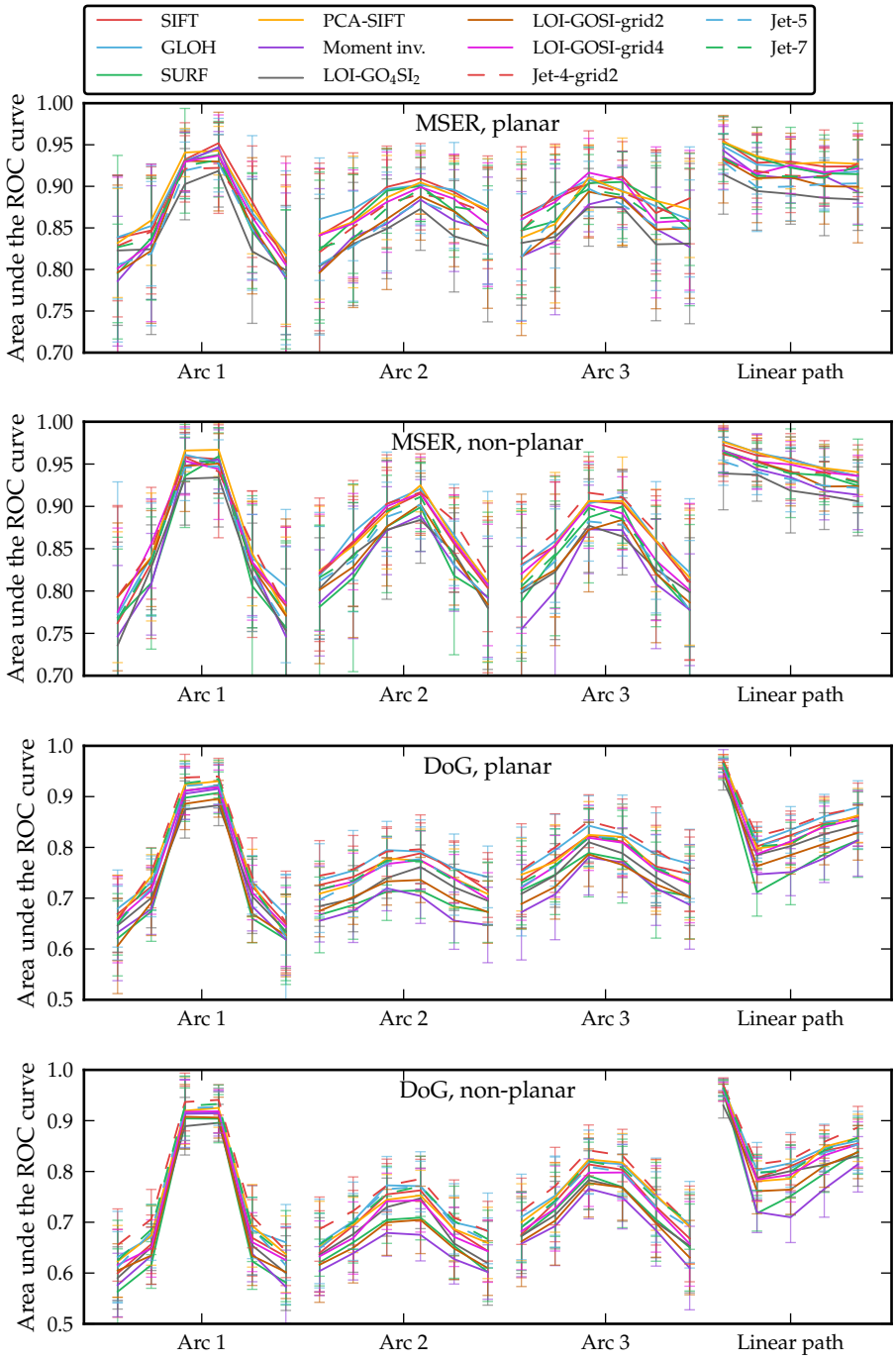


Figure 7.10: Planar vs. non-planar scene structures used for evaluating descriptor performances on DoG and MSER interest points.

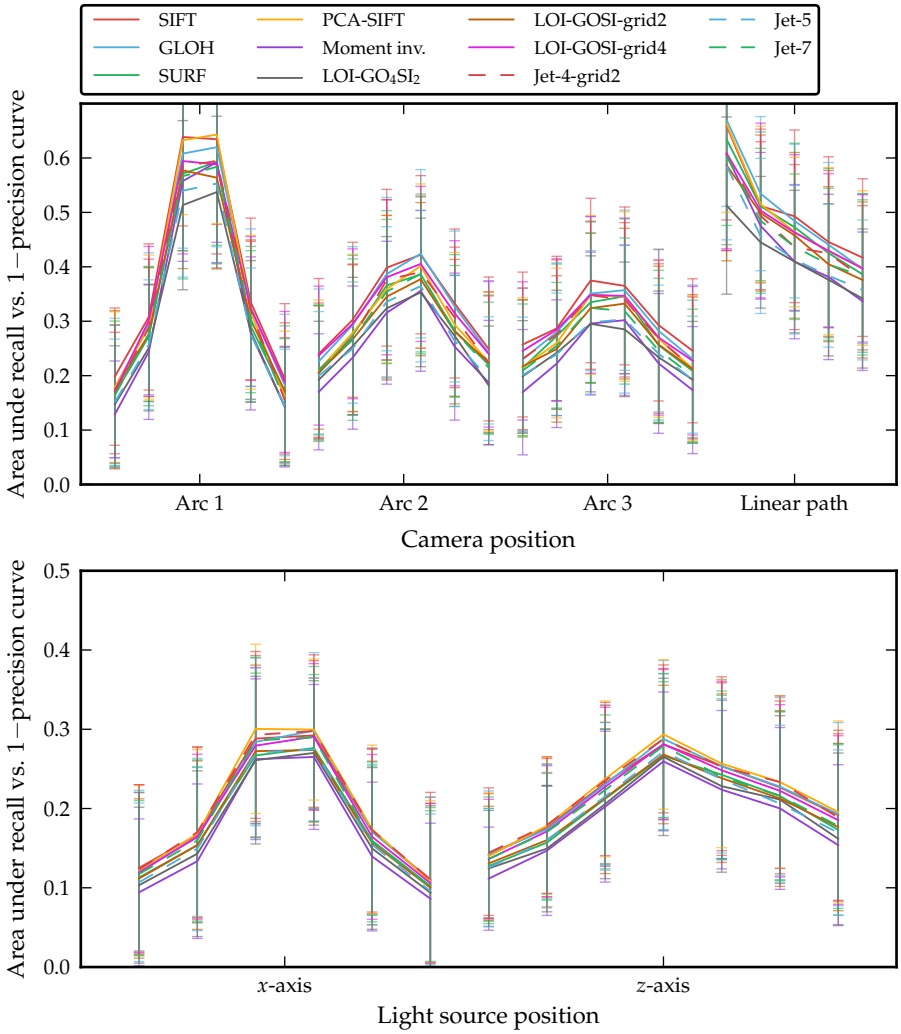


Figure 7.11: Recall vs. 1 − precision performance on MSER interest points. This graph should be compared to Figure 7.5 showing recall vs. fall-out performance.

scriptor ordering is not sufficiently significant for us to change our previous conclusions about descriptor performance. We do, however, advocate the usage of the recall vs. 1 − precision measure as it yields results that are more consistent and interpretable.

7.4 Sensitivity analysis

In Section 6.3, we presented an alternative evaluation method for local image description where the number of nuisance factors is reduced to better reveal the sensitivity towards a single perturbation factor. With this method we replace the ROC analysis with Welch's t-test on the distribution of distances to the reference descriptors and the distribution of distances to all other

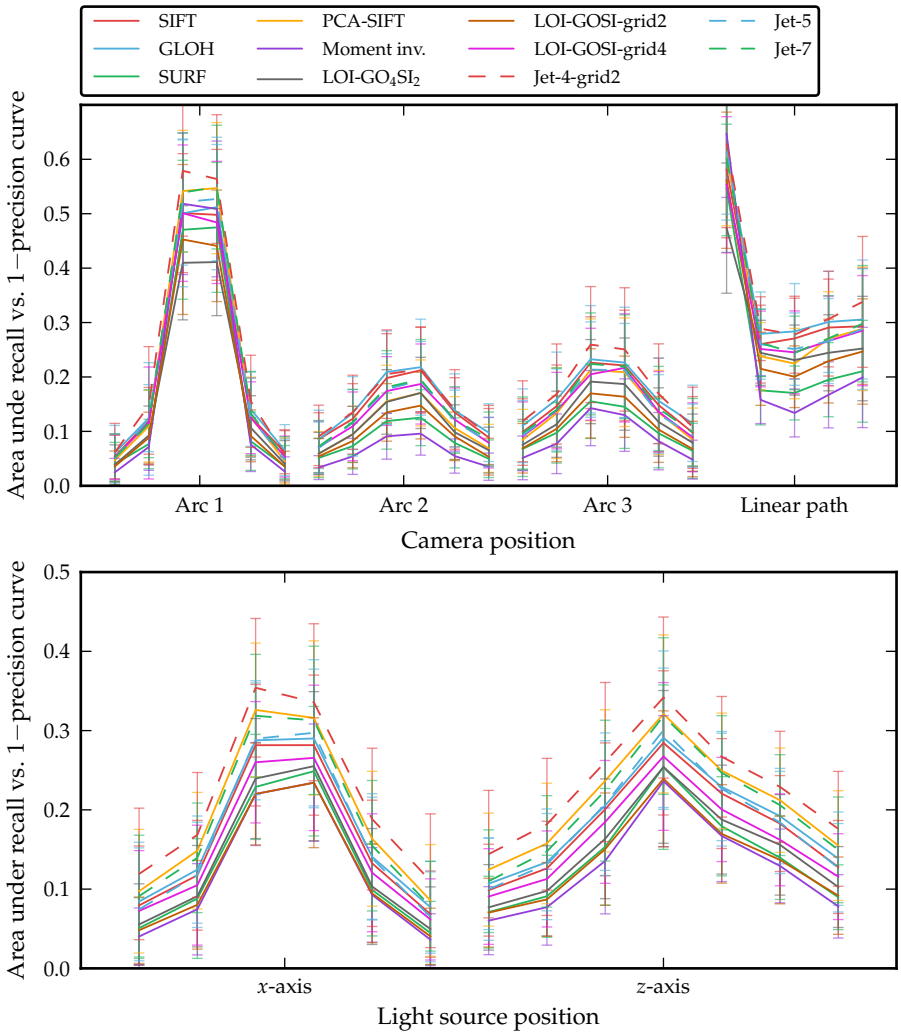


Figure 7.12: Recall vs. 1 − precision performance on DoG interest points. This graph should be compared to Figure 7.6 showing recall vs. fall-out performance.

descriptors in the image. We have plotted these results in Figure 7.13. A large t-value means large discrepancy and thereby better discriminative abilities. As we see, this method yields quite different results from the previous evaluation measure since LOI-GO_4SI_2 and SIFT seems to be most robust to camera displacements. For illumination changes, SURF stands out as being robust to more high-contrast areas (the light source positions where it outperforms the others are the light source positions in middle and in front of the scene). These results are interesting as we see descriptors, that have previously performed poorly, being less sensitive in some of the perturbation scenarios. The results are also somewhat problematic as we would like our evaluation criteria to reflect the discriminative ability in order to remain relevant to the feature matching scenario. To accommodate this problem, we propose a slight alteration of our evaluation criteria.

In Figure 7.14, we have plotted the t value from Welch's t-test on the distribution of distances to the reference descriptors and the distribution of distances to the *nearest descriptor* (in terms of feature vector distance). Thus, we require that descriptors stand out similar to the nearest neighbor distance ratio matching as described in Section 2.5. For this revised evaluation measure, we see that descriptor performance is more similar to the matching scenario. We find the sensitivity of the SIFT descriptor remarkable as it clearly outperforms the other descriptors for camera displacements. When comparing SIFT to LOI-GOSI-grid4 we see that LOI-GOSI-grid4 is more sensitive to changes in view angle and in scale, since it performs well for small perturbations but quickly gets surpassed by SIFT on Arc 1 and the Linear path. We might accredit this behavior to the shape index that may be more sensitive to camera displacements. For lighting variations, we see a repetition of the pattern that the Jet-4-grid-2 descriptor followed by PCA-SIFT are superior at discriminating the local feature vectors

We conclude the sensitivity experiments by remarking that this evaluation criteria needs further work if we want to obtain more specific results. For example, it would be relevant to investigate the above suppositions in detail in order to pinpoint exactly what descriptor designs that increases the sensitivity of a descriptor towards the different perturbations. However, because of time constraints, this has not been carried out.

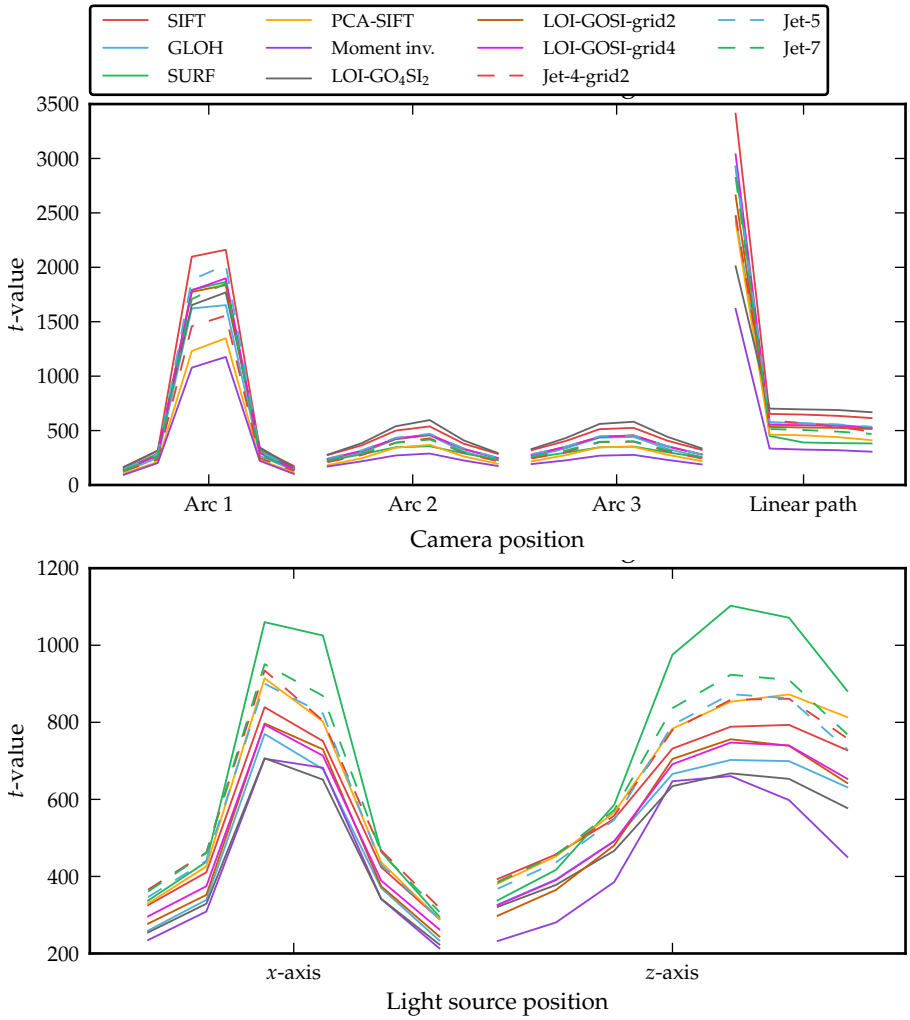


Figure 7.13: Sensitivity analysis. t is the result of Welch's t-test on the distribution of distances to the reference descriptors and the distribution of distances to all other descriptors in the image.

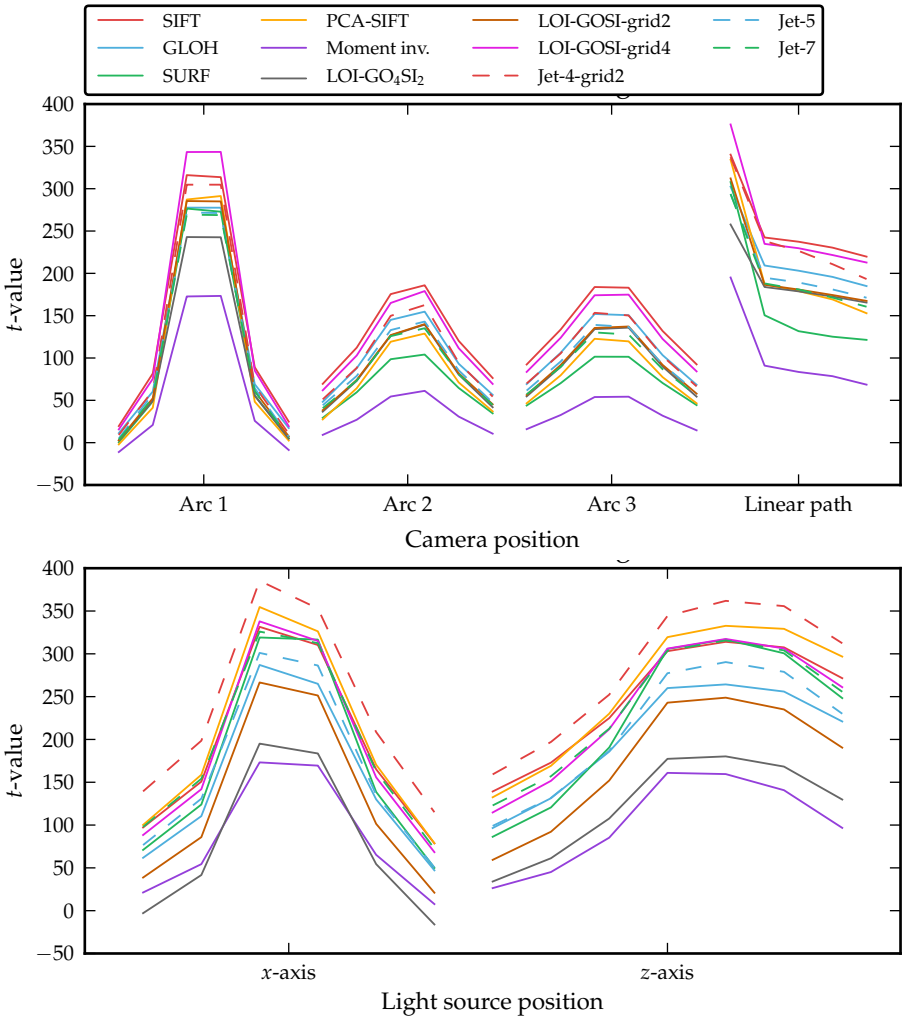


Figure 7.14: Welch's t-test on the distribution of distances to the reference descriptors and the distribution of distances to the nearest descriptor (in terms of feature vector distance).

8 Discussion

In the previous two sections we have studied descriptor performance on a large variety of situations. We remark that all our experiments have been carried out on a superior dataset in terms of realism compared to previous datasets. This allows us to reason about descriptor performance from a significantly better average than previous evaluations. While our dataset is a good representative for the general descriptor performance, it may not be reflect the performance in very specific situations or for a specialized application. Nevertheless, we still believe that our results are more representative than the majority of previous descriptor evaluations in the literature [5, 28, 41, 44].

8.1 Descriptor design

In this project, we have dissected the typical descriptor design and tried to analyze what makes a good local descriptor. We have done so to better illuminate the importance of the different steps in a description algorithm which has lacked a thorough coverage in the literature.

Many design choices are made in a feature description algorithm, and for most descriptors in the literature, these choices are often taken without proper justification (e.g. in the form of empirical results). For example, the GLOH descriptor [30] uses 17 histograms gathered from a polar grid, but the performance implications of adjusting the number of histograms is not shown. This leaves the reader unknowing about the importance of the different design choices and makes it harder to pinpoint what works when comparing different descriptor algorithms. Another problem with the GLOH descriptor (and for most other local feature descriptors in the literature) is that no details are provided regarding the parameter optimization of the descriptor. This is a bit worrying as the parameters of GLOH might have been optimized using the testing dataset. Another significant problem is that we do not know all the values of the optimal parameters of the GLOH descriptor which makes reproduction of the results harder.

Design choices of the SIFT descriptor

We have shown that for the DTU dataset, the SIFT descriptor is unnecessarily complex on two points. The thresholded normalization yields no performance improvements, in fact, it seems to affect performance slightly in the opposite direction. This result conflicts with [25, 52]. An explanation could be that thresholded normalization improves only robustness towards differences in camera exposure. Exposure perturbations are not part of the DTU dataset and therefore, thresholded normalization has no positive effect in our experiments. It has previously been speculated that thresholded normalization also helps in case of changes in illumination and occlusions where parts of a descriptor

changes causing the normalization to distort the remaining parts [52]. In these situations, our results show that thresholded normalization does not work.

Secondly, the additional Gaussian window applied to the gradient magnitudes does not seem to have any effect on the performance. A noteworthy property of the additional Gaussian window is that it is computationally very expensive to use when SIFT descriptors are sampled densely at a fixed scale σ from an image $I(\boldsymbol{r})$. This is because the gradient orientation magnitudes must be weighted differently for each descriptor. Without the Gaussian window, the gradient orientation magnitudes for all fixed-scale descriptors could be computed from just $L_x(\boldsymbol{r};\sigma)$ and $L_y(\boldsymbol{r};\sigma)$ [49]. Our results show that these extra computations can be omitted without any discriminative degradation of the descriptor.

Multi-local vs. multi-scale vs. higher-order description

One of the main goals of this project has been to analyze multi-local histogramming for local description. Following the success of SIFT, this approach has become dominant for local feature description [8,31,44,52]. We have also challenged the approach by investigating a different description strategy where multi-locality is substituted/supplemented by multi-scale and higher-order features.

With our Jet descriptors we have shown that gradient orientation histograms can be replaced quite successfully using higher-order differentials in form of the local k-jet. Moreover, we have shown that higher-order image structure is able to reduce multi-locality to some degree since our 2×2 grid of local 4-jets achieves very competitive performance compared to other state-of-the-art descriptors. This is a novel discovery as research on higher-order differential descriptors has effectively been abandoned after the popularization of multi-local histogramming. Another noteworthy advantage of the Jet descriptors is that they are simple to implement and their parameters are easily optimized compared to histogram-based descriptors with more configuration possibilities (notably, the inner scale, the outer scale and the tonal scale parameters).

Our attempt at supplementing gradient orientation histograms with shape index histograms has shown some potential as we are able to improve performance, but not to a degree where it can replace multi-locality as convincingly as the Jet descriptor. Likewise, our multi-scale feature descriptors have shown limited success as they are able to achieve performance similar to a multi-local descriptor (LOI-GO_4SI_4 vs. LOI-GO-grid2). However, we do not recommend this approach as the local k-jet yields significantly better performance with a lower dimensionality.

Performance vs. descriptor dimensionality

As we have seen with the PCA-SIFT algorithm [17], the dimensionality of e.g. SIFT is needlessly high. Thus, we have constrained our descriptor designs by keeping the descriptor dimensionality low to explore the performance implications. As it turns out, the use of higher-order features is able to

reduce the descriptor dimensionality significantly with little or no reduction in performance. We have even seen that the 20-dimensional Jet-5 is very competitive with state-of-the-art descriptors SIFT on DoG interest points. This result is remarkable considering the conclusions in [30] where single-local descriptors are considered less robust than multi-local descriptors.

Our approach to descriptor design is quite different from that of Winder et al. [51, 52] as they push the feature descriptor dimensionality as far as possible to achieve the best performance. They show, for example, that a feature descriptor of dimensionality 400 achieves the best performance [51]. We speculate that they might be overfitting the descriptors to their dataset, as it is somewhat specialized containing only three different scenes. In [52], they use PCA to reduce the descriptor dimensionality and end up with a 15-dimensional descriptor yielding better performance than SIFT. It should be remarked that that they do not measure performance on different interest point types, but on perturbed DoG interest points. This is problematic as we have seen that descriptor performance is highly dependent on the interest point types. We would have liked to include a comparison with the descriptors from Winder et al. Unfortunately, neither their code nor their dataset is available online.

Design choices for different feature matching scenarios

As we have seen, the robustness of a descriptor design depends to a great extent on the situation. Some of our results have been shown before in e.g. [7, 30], however, we are able to provide a more nuanced view.

We have seen that the choice of interest point detector greatly influence the ordering of descriptor performances. For the region-based detector, MSER, multi-local histogramming descriptors like SIFT, PCA-SIFT and GLOH seem to perform best for view angle and scale changes. For the blob-based detector, DoG, our Jet approach seems to have a clear advantage in all test cases. For the corner-based detectors, HarLap and HarAff, PCA dimensionality reduction of multi-local histogramming descriptors seems more robust to changes in view angle and scale.

Regarding the perturbation factors, we have shown the following general trends. For view angle changes, multi-local descriptors like GLOH, SIFT, PCA-SIFT and our Jet-4-grid2 descriptor yield the best results. For illumination changes, the Jet approach yields the best results followed by PCA-SIFT. Furthermore, we have shown that the performance of multi-local histogramming descriptors drops significantly compared to our Jet descriptors when used on non-planar surfaces. We speculate that this could explain part of the success of multi-local histogramming descriptors, since the majority of datasets for evaluating local descriptor performance contain only planar structures.

8.2 Challenges of descriptor evaluation

It should be noted that the Jet descriptors are not as convincing when evaluated on the Oxford dataset as on the DTU dataset. The reason for this is simply that the Oxford dataset does not include the perturbations that favor

the Jet descriptor such as non-planar surfaces and changes in light source positioning. This reveals a shortcoming of the Oxford dataset; it is not varied enough to cover descriptor performance in all evaluation scenarios. Note that same can be said about the DTU dataset for perturbations with regards to camera exposure and rotations around the optical axis.

Thus, we have on several occasions shown results that conflict with results from previous papers. Our findings reveal a recurring problem with the research of local feature description. Descriptors are often evaluated on small and unrepresentative datasets with a relatively narrow set of evaluation scenarios that fail to reveal the general performance of the descriptors. For example, in the SURF paper [5], SURF is evaluated only on interest points generated by a custom-made detector (*Fast Hessian*) using selected scenes from the Oxford dataset. In these cases, SURF is shown to outperform both SIFT and GLOH by a visible margin. The problem is that the evaluation is not fair to SIFT and GLOH. SIFT is intended to be used on DoG interest points [25], and therefore, the evaluation should have included performance graphs using these as well. Thus, the SURF paper only reveals that SURF achieves a local maximum for the given feature matching setup which is difficult to compare with other feature matching setups with different descriptors/detectors. Note that with the same argumentation, we are not fair in our descriptor evaluation either as we do not include the Fast Hessian detector, which might have been an advantage to the SURF descriptor.

We admit that local image descriptors are very difficult to evaluate in a thorough and representative manner. Our results in this project are a testament to that as we have seen the individual descriptor performances vary significantly over different evaluation scenarios. Moreover, we have found it difficult to come up with an alternative evaluation criterion that reflects the robustness and sensitivity of a descriptor when exposed to an isolated perturbation factor.

The main challenge of descriptor evaluation is the complexity caused by the many configuration possibilities of the feature matching setup combined with the various perturbation factors. Therefore, it is important that the evaluation criteria reflects the actual end usage/application of the descriptor. For example, one should consider whether the recall vs. fallout or the recall vs. $1-$ precision criteria is more suitable for a specific application. In situations where the goal is to match as many features correctly as possible with little importance of mispredictions, the recall vs. fallout criteria might be more representative. For applications more sensitive to mispredictions, the recall vs. $1-$ precision measure is better suited as it penalizes mispredictions to a larger degree.

With the complex state of contradicting descriptor evaluations, we understand the popularity gained by the SIFT descriptor as it was the first to achieve consistent results across all usage scenarios. In some case, SIFT might not be the optimal choice, but for the general computer vision system it is *good enough*. Furthermore, it has good and relatively fast implementations available online [25, 49].

8.3 Future directions

As described above, the descriptor evaluation performed in this project is lacking some deformation scenarios, most notably rotations around the optical axis. It could also be interesting to dissect descriptor performance even further by studying the descriptor performance of more specialized scenarios. For example, we could compare the descriptor behavior on different scene types (e.g. plush toys vs. beer cans). Moreover, our sensitivity analysis should be expanded to include further details. At its current state, we are not completely sure what causes the results that differ from the ordinary matching experiments. That being said, it is not our goal to make the two evaluation criteria identical. It would, however, be relevant to know the effect of e.g. discarding occluded interest points to see which descriptors are most sensitive towards occlusions.

Another angle that we have ignored in this project is the use of color information to describe image patches. Typically, the image description is performed in an alternative color space (e.g. opponent RGB) on the color channels separately [46]. It would be interesting to see which descriptors work best under these conditions.

Regarding the descriptors proposed in this project, we could extend them by performing PCA dimensionality reduction similar to PCA-SIFT and GLOH. This would most likely allow us to turn up the descriptor complexity (e.g. in terms of multi-locality), however, it would break with our simplified approach to designing local feature descriptors. Furthermore, we could experiment with quantization of our descriptor vectors (similar to SIFT) to lower their storage footprint. Finally, it could also be interesting to try our Jet descriptors in a suitable computer vision system to test their performance in a more realistic setting.

9 Conclusion

In this project, we have performed an in-depth study of local image descriptors. We have investigated different description algorithms to see what makes a good descriptor and we have evaluated their performance in a comparative study with other state-of-the art descriptors.

For the performance evaluation we have used the recently released DTU Robot dataset that offers unique evaluation possibilities compared to other datasets available, mainly wrt. to changes to view angle, scale and lighting.

We have proposed new descriptors based on the locally orderless image representation and on higher-order differential structure. This approach is somewhat different from the popular image descriptors in the literature that are usually based on multi-local gradient orientation histograms. By using higher-order image structure in form of the local k-jet, we have shown that we are able to reduce both the multi-locality and the descriptor dimensionality with little or no loss in the discriminative ability of the descriptor.

In our comparative evaluation, we have shown that SIFT, GLOH, PCA-SIFT and our Jet-4-grid2 perform well across most scenarios in the dataset. However, the individual descriptor performance is dependent on many influencing factors as we have seen numerous times in our different evaluation scenarios. For example, the Jet-4-grid-2 has an advantage over the three other descriptors when using DOG interest points (and vice versa for MSER, HarLap and HarAff). Furthermore, Jet-4-grid-2 is always the top performer for lighting variations. Thus, when selecting a local image descriptor for a computer vision system, one should take the different advantages and disadvantages of each descriptor into consideration.

Bibliography

[1] H. Aanæs, A. Dahl, and K. Pedersen. On recall rate of interest point detectors. In *3DPVT 2010: Fifth International Symposium on 3D Data Processing, Visualization and Transmission*, 2010.

[2] H. Aanæs, A. Dahl, and K. Steenstrup Pedersen. Interesting interest points. *International Journal of Computer Vision*, pages 1–18, 2011.

[3] S. Agarwal, N. Snavely, I. Simon, S. Seitz, and R. Szeliski. Building rome in a day. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 72–79. Ieee, 2009.

[4] E. Balmashnova and L. Florack. Novel similarity measures for differential invariant descriptors for generic object retrieval. *Journal of Mathematical Imaging and Vision*, 31:121–132, 2008. 10.1007/s10851-008-0079-0.

[5] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool. Speeded-up robust features (surf). *Computer Vision and Image Understanding*, 110(3):346 – 359, 2008. Similarity Matching in Computer Vision and Multimedia.

[6] G. Carneiro and A. Jepson. Phase-based local features. In A. Heyden, G. Sparr, M. Nielsen, and P. Johansen, editors, *Computer Vision — ECCV 2002*, volume 2350 of *Lecture Notes in Computer Science*, pages 282–296. Springer Berlin / Heidelberg, 2002.

[7] A. L. Dahl, H. Aanæs, and K. S. Pedersen. Finding the best feature detector-descriptor combination. In *The First Joint Conference of 3D Imaging, Modeling, Processing, Visualization and Transmission*, May 2011.

[8] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005*. IEEE Computer Society, 2005.

[9] R. Datta, D. Joshi, J. Li, and J. Z. Wang. Image retrieval: Ideas, influences, and trends of the new age. *ACM Comput. Surv.*, 40:5:1–5:60, May 2008.

[10] J. Davis and M. Goadrich. The relationship between Precision-Recall and ROC curves. In *Proceedings of the 23rd international conference on Machine learning*, ICML '06, pages 233–240, New York, NY, USA, 2006. ACM.

[11] D. Endres and J. Schindelin. A new metric for probability distributions. *Information Theory, IEEE Transactions on*, 49(7):1858–1860, 2003.

[12] T. Fawcett. An introduction to ROC analysis. *Pattern Recognition Letters*, 27(8):861 – 874, 2006. ROC Analysis in Pattern Recognition.

[13] L. Florack, B. M. ter Haar Romeny, M. Viergever, and J. Koenderink. The gaussian scale-space paradigm and the multiscale local jet. *International Journal of Computer Vision*, 18:61–75, 1996.

[14] L. M. Florack, B. M. ter Haar Romeny, J. J. Koenderink, and M. A. Viergever. Scale and the differential structure of images. *Image and Vision Computing*, 10(6):376 – 388, 1992. Information Processing in Medical Imaging.

[15] L. Juan and O. Gwun. A comparison of SIFT, PCA-SIFT and SURF. *International Journal of Image Processing (IJIP)*, 3(4):143–152, 2009.

[16] B. Kaneva, A. Torralba, and W. T. Freeman. Evaluating image feaures using a photorealistic virtual world. In *IEEE International Conference on Computer Vision*, 2011.

[17] Y. Ke and R. Sukthankar. PCA-SIFT: A more distinctive representation for local image descriptors. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 2:506–513, 2004.

[18] J. Koenderink. The structure of images. *Biological Cybernetics*, 50:363–370, 1984. 10.1007/BF00336961.

[19] J. J. Koenderink and A. J. van Doorn. Surface shape and curvature scales. *Image and Vision Computing*, 10(8):557 – 564, 1992.

[20] J. J. Koenderink and A. J. Van Doorn. The structure of locally orderless images. *International Journal of Computer Vision*, 31:159–168, 1999.

[21] S. Kullback and R. Leibler. On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86, 1951.

[22] J. Lin. Divergence measures based on the shannon entropy. *Information Theory, IEEE Transactions on*, 37(1):145–151, 1991.

[23] T. Lindeberg. *Scale-space theory in computer vision*. Kluwer international series in engineering and computer science. Kluwer Academic, 1994.

[24] T. Lindeberg. Scale-space: A framework for handling image structures at multiple scales. In *Proceedings of CERN School of Computing*, pages 27–38, 1996.

[25] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60:91–110, 2004.

[26] B. Markussen, K. Pedersen, and M. Loog. A scale invariant covariance structure on jet space. In O. Fogh Olsen, L. Florack, and A. Kuijper, editors, *Deep Structure, Singularities, and Computer Vision*, volume 3753 of *Lecture Notes in Computer Science*, pages 12–23. Springer Berlin / Heidelberg, 2005.

[27] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide-baseline stereo from maximally stable extremal regions. *Image and Vision Computing*, 22(10):761 – 767, 2004.

[28] K. Mikolajczyk, B. Leibe, and B. Schiele. Local features for object class recognition. *Computer Vision, IEEE International Conference on*, 2:1792–1799, 2005.

[29] K. Mikolajczyk and C. Schmid. Scale & affine invariant interest point detectors. *International Journal of Computer Vision*, 60:63–86, 2004. 10.1023/B:VISI.0000027790.02288.f2.

[30] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27:1615–1630, 2005.

[31] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. V. Gool. A comparison of affine region detectors. *International Journal of Computer Vision*, 65(1/2):43–72, 2005.

[32] F. Mindru, T. Tuytelaars, L. V. Gool, and T. Moons. Moment invariants for recognition under changing viewpoint and illumination. *Computer Vision and Image Understanding*, 94(1–3):3 – 27, 2004. Special Issue: Colour for Image Indexing and Retrieval.

[33] P. Moreno, A. Bernardino, and J. Santos-Victor. Improving the sift descriptor with smooth derivative filters. *Pattern Recognition Letters*, 30(1):18 – 26, 2009.

[34] K. Murphy, A. Torralba, D. Eaton, and W. Freeman. Object detection and localization using local and global features. In J. Ponce, M. Hebert, C. Schmid, and A. Zisserman, editors, *Toward Category-Level Object Recognition*, volume 4170 of *Lecture Notes in Computer Science*, pages 382–400. Springer Berlin / Heidelberg, 2006.

[35] E. Nowak, F. Jurie, and B. Triggs. Sampling strategies for bag-of-features image classification. In A. Leonardis, H. Bischof, and A. Pinz, editors, *Computer Vision – ECCV 2006*, volume 3954 of *Lecture Notes in Computer Science*, pages 490–503. Springer Berlin / Heidelberg, 2006.

[36] K. S. Pedersen. *Statistics of natural image geometry*. PhD thesis, University of Copenhagen, Department of Computer Science, 2003.

[37] K. S. Pedersen, A. L. Dahl, and H. Aanæs. personal communication, 2012.

[38] O. Pele and M. Werman. The quadratic-chi histogram distance family. In K. Daniilidis, P. Maragos, and N. Paragios, editors, *Computer Vision – ECCV 2010*, volume 6312 of *Lecture Notes in Computer Science*, pages 749–762. Springer Berlin / Heidelberg, 2010.

[39] C. Schmid and R. Mohr. Local grayvalue invariants for image retrieval. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 19(5):530–535, 1997.

[40] R. Szeliski. Image alignment and stitching: a tutorial. *Found. Trends. Comput. Graph. Vis.*, 2:1–104, January 2006.

[41] F. Tang, S. Lim, N. Chang, and H. Tao. A novel feature descriptor invariant to complex brightness changes. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 2631–2638. IEEE, 2009.

[42] B. ter Haar Romeny. *Front-End Vision and Multi-Scale Image Analysis*, volume 27 of *Computational Imaging and Vision*. Springer, 2003.

[43] B. M. ter Haar Romeny, L. M. Florack, A. H. Salden, and M. A. Viergever. Higher order differential structure of images. *Image and Vision Computing*, 12(6):317 – 325, 1994. Information processing in medical imaging.

[44] E. Tola, V. Lepetit, and P. Fua. Daisy: An efficient dense descriptor applied to wide-baseline stereo. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(5):815–830, 2010.

[45] T. Tuytelaars and K. Mikolajczyk. Local invariant feature detectors: a survey. *Foundations and Trends® in Computer Graphics and Vision*, 3:177–280, July 2008.

[46] K. Van De Sande, T. Gevers, and C. Snoek. Evaluating color descriptors for object and scene recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(9):1582–1596, 2010.

[47] B. van Ginneken and B. ter Haar Romeny. Applications of locally orderless images. In M. Nielsen, P. Johansen, O. Olsen, and J. Weickert, editors, *Scale-Space Theories in Computer Vision*, volume 1682 of *Lecture Notes in Computer Science*, pages 10–21. Springer Berlin / Heidelberg, 1999.

[48] L. Van Gool, T. Moons, and D. Ungureanu. Affine / photometric invariants for planar intensity patterns. In B. Buxton and R. Cipolla, editors, *Computer Vision — ECCV '96*, volume 1064 of *Lecture Notes in Computer Science*, pages 642–651. Springer Berlin / Heidelberg, 1996. 10.1007/BFb0015574.

[49] A. Vedaldi and B. Fulkerson. VLFeat: An open and portable library of computer vision algorithms, 2008.

[50] A. Vedaldi, H. Ling, and S. Soatto. Knowing a good feature when you see it: Ground truth and methodology to evaluate local features for recognition. In R. Cipolla, S. Battiato, and G. Farinella, editors, *Computer Vision*, volume 285 of *Studies in Computational Intelligence*, pages 27–49. Springer, 2010.

[51] S. Winder and M. Brown. Learning local image descriptors. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE, 2007.

[52] S. Winder, G. Hua, and M. Brown. Picking the best daisy. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 178–185. IEEE, 2009.

[53] A. P. Witkin. Scale-space filtering. In *Proceedings of the Eighth international joint conference on Artificial intelligence - Volume 2*, pages 1019–1022, San Francisco, CA, USA, 1983. Morgan Kaufmann Publishers Inc.

[54] J. Zhang, M. Marszalek, S. Lazebnik, and C. Schmid. Local features and kernels for classification of texture and object categories: A comprehensive study. In *Computer Vision and Pattern Recognition Workshop, 2006. CVPRW'06. Conference on*, pages 13–13. Ieee, 2006.

[55] H. Zhou, Y. Yuan, and C. Shi. Object tracking using sift features and mean shift. *Computer Vision and Image Understanding*, 113(3):345 – 352, 2009. Special Issue on Video Analysis.

A Artificial scene relighting

In Section 6, we have assumed that the scene lighting possibilities consisted of diffuse lighting and two paths along the x and z-axis respectively. However, this does not reflect the original dataset that offers more complex lighting possibilities. In this appendix we briefly describe the scene illumination of the DTU Robot dataset and how to generate the two artificial light paths that are used in this project. We follow the scene relighting method described in [2].

In the DTU Robot dataset, the scene objects are illuminated exclusively by 19 LEDs placed over the scene. See Figure A.1 for a top-down view of the lighting setup. For each camera position, 19 images I_i , $i = 1, \ldots, 19$ are taken to capture the illumination caused by each LED. The diffuse lighting image I_{diffuse} is generated by averaging the 19 images $I_{\text{diffuse}} = \frac{1}{19} \sum_{i=1}^{19} I_i$. The image I_x resulting from a scene illumination by an artificial camera placed at location $\boldsymbol{x}$ is generated by placing a Gaussian window with $\sigma = 20$ at $\boldsymbol{x}$. I_x is then calculated from the linear combination $I_{\boldsymbol{x}} = \sum_{i=1}^{19} w_i I_i$, where w_i denotes the weight as calculated from the Gaussian: $w_i = c \exp\left(-\frac{(\boldsymbol{x}-\boldsymbol{x}_i)^2}{2\sigma^2}\right)$. $\boldsymbol{x}_i$ denotes the position of I_i and c is chosen such that $\sum_{i=1}^{19} w_i = 1$.

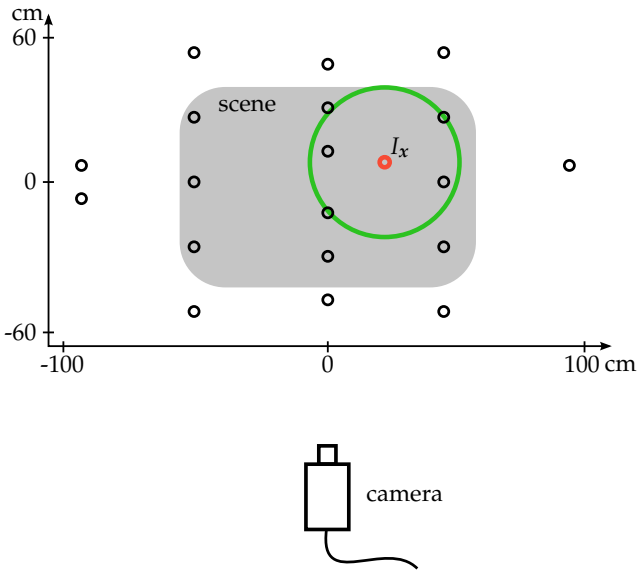


Figure A.1: Top-down view of the light setup of the DTU Robot dataset. The black circles indicate LED placements. The red circle indicates the position of an artificial light source and the green circle represents the Gaussian window used to weight the contributions of the surrounding LEDs.