Estimation of Location Uncertainty for Scale Invariant Feature Points

Bernhard Zeisl[1]
http://campar.in.tum.de/Main/BernhardZeisl
Pierre Fite Georgel[1]
http://campar.in.tum.de/Main/PierreGeorgel
Florian Schweiger[2]
http://www.lmt.ei.tum.de/team/florian/
Eckehard Steinbach[2]
http://www.lmt.ei.tum.de/team/steinb/
Nassir Navab[1]
http://campar.in.tum.de/Main/NassirNavab

[1] Chair for Computer Aided Medical
Procedures & Augmented Reality
Technische Universität München
Munich, GER

[2] Institute for Media Technology
Technische Universität München,
Munich, GER

Abstract

Image feature points are the basis for numerous computer vision tasks, such as pose estimation or object detection. State of the art algorithms detect features that are invariant to scale and orientation changes. While feature detectors and descriptors have been widely studied in terms of stability and repeatability, their localisation error has often been assumed to be uniform and insignificant.

We argue that this assumption does not hold for scale-invariant feature detectors and demonstrate that the detection of features at different image scales actually has an influence on the localisation accuracy. A general framework to determine the uncertainty of multi-scale image features is introduced. This uncertainty is represented via anisotropic covariances with varying orientation and magnitude. We apply our framework to the well-known SIFT and SURF algorithms, detail its implementation and make it available [1]. Finally, the usefulness of such covariance estimates for bundle adjustment and homography computation is illustrated.

1 Introduction

Robust image feature point detection, matching, and tracking represent basic operations for many computer vision algorithms. Feature points are meaningful and stable points in an image, which are extracted using a mathematical operator. Once extracted, they are described in a distinctive way. The interest points and the attached descriptors define an abstract image representation. Feature based methods have been successfully applied in many fields such as scene modeling [15], 3D tracking [18] and image retrieval [8].
One of the most prominent (corner) detectors was introduced by Harris [5]. For detection it

BMVC 2009 doi:10.5244/C.23.57

[1] Binaniers and code are available from http://campar.in.tum.de/Main/CovarianceEstimator

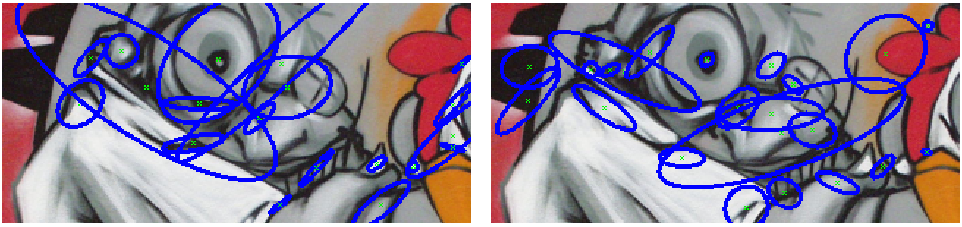


Figure 1: Covariance estimates for interest regions detected by SIFT (left) and SURF (right).

builds on the second order derivative matrix, constructed from intensity values, while matching is performed employing the cross correlation between local image patches. The matching accuracy and robustness heavily depend on the actual transformation between views.
To tackle this problem several scale and rotation invariant interest point detectors have been introduced. They are referred to as blob detectors, because they are not only able to detect points in an image but also interest regions; simply speaking areas which are brighter or darker than the surrounding. Different to basic corner detectors the algorithms search for interest points in scale space allowing us to find similar features at different scales. To address rotation invariance, the descriptor often detects a local primary orientation. Popular region detectors are the SIFT [10], SURF [1] or Hessian-Laplace, Harris-affine, and Hessian-affine detectors. The latter two are additionally invariant to affine transformations. Schmid *et al.* [14] and Mikolajczyk *et al.* [11, 12] provide comprehensive evaluations of different feature point and region detectors. [14] compares corner detectors similar to Harris in terms of repeatability and information content, while [11, 12] cover scale-invariant region detectors. They measure the detector performance based on repeatability, "localisation error" and the number of correspondences in images under different geometric and photometric transformations.
While the "localisation error" of the detector has been evaluated between matching features, an investigation of the detection precision has not been performed to the best of our knowledge. By doing so it is possible to parametrise the localisation error. In this sense our work is complementary to the extensive comparisons in [12, 14] and the goal of this work is to obtain an individual estimate for the localisation uncertainty for each region found. Figure 1 exemplarily displays results we obtained [2]. The covariances can then be used for model fitting where we minimise a weighted least square cost function based on covariances instead of minimising a least square cost function (which assumes a uniform error across the data). Model fitting itself appears in many different computer vision problems including scene modelling with bundle adjustment and stitching with homography estimation. These two applications are used in our work to demonstrate the usefulness of the proposed localisation error estimation.

The paper is structured as follows. Section 2 introduces related work on the topic of covariance estimation. In Section 3 we propose a general framework for uncertainty estimation applicable for scale invariant feature detectors. Section 4 illustrates the application of our framework to SIFT and SURF. Experiments related to these implementations are presented in Section 5, while Section 6 shows results for the incorporation of our estimates into existing algorithms. Section 7 concludes the paper.

[2] We use the Oxford image dataset provided at http://www.robots.ox.ac.uk/~vgg/research/affine/

2 Related Work

Uncertainty estimation for corner-like points as a measure for the localisation precision has been studied before. Common to all approaches is the assumption of a Gaussian error model and hence the characterization using a 2D covariance matrix. The error is assumed to arise either from pixel intensity noise or from the detection algorithm itself.

Brooks *et al.* [3] take into consideration the curvature of the self-matching residual at a Harris-corner point and estimate the covariance from the second moment matrix based on pixel intensities. They demonstrate an error reduction for fundamental matrix estimation.

Kanazawa and Kanatani [7] raise the question about the usefulness of covariance matrices for image features. They provide a more theoretical evaluation and show that covariance estimates based on the Hessian calculated from image intensities reflect the uncertainty of feature localisation. Furthermore, they state that the incorporation of such covariances does not improve homography or fundamental matrix estimation because estimated covariances seem to be isotropic and of similar size. Both works mentioned develop their argumentation independent from the detection operator applied, but associate location uncertainty with the residual error occurring in template matching.

Steele and Jaynes [16] on the other hand focus on the detector and address the problem of feature inaccuracy based on pixel noise. They use different noise models for pixel intensities and propagate the related covariances through the detection process of the Förstner-corner detector to come up with a covariance estimate for each feature point.

Haja *et al.* [4] provide a comparison of region detectors with respect to localisation accuracy. They look at the matching precision of regions; however, they do not parametrize the localisation error of an interest point itself.

Orguner and Gustafsson [13] evaluate the accuracy for Harris corner points. The analysis is built on the probability that pixels are the true corner in the region around the corner estimate. They have found that the accuracy for a corner point can vary depending on the different image color channels (RGB).

Important to note is that [3, 7, 13, 16] base their argumentation on corner detectors which are *not* scale-invariant. Scale-invariant region detectors extract image regions, complementary to the corner-like features, hence we claim two things: First, due to the focus on interest regions, the shape of covariances will be in general anisotropic. Second, the magnitude of covariances will vary significantly due to detection in scale space.

Wu *et al.* [19] also observed a behaviour according to the second statement and introduce less weight for interest points detected on higher scales.

3 Uncertainty Evaluation Framework

Our analysis is based on the assumption that a detection process locates a feature, and that this process generates a measurement error that conforms to a bivariate normal distribution. In this section we will explain the general framework we have developed to estimate the covariance matrix describing this distribution.

Common to all scale invariant feature detectors is a two step approach to find feature points. First, a scale-space representation in form of an image stack D (see Figure 2) is created with the *feature detection operator* at preselected scales $\sigma_i \in \{\sigma_j\}_{j=1\ldots N}$ from the image I. The detection operator $\mathbf{f_{dec}}$ depends on the particular algorithm and is not necessarily a linear function. For the calculation of the operator response $D(\mathbf{x}, \sigma_i)$ at a specific location

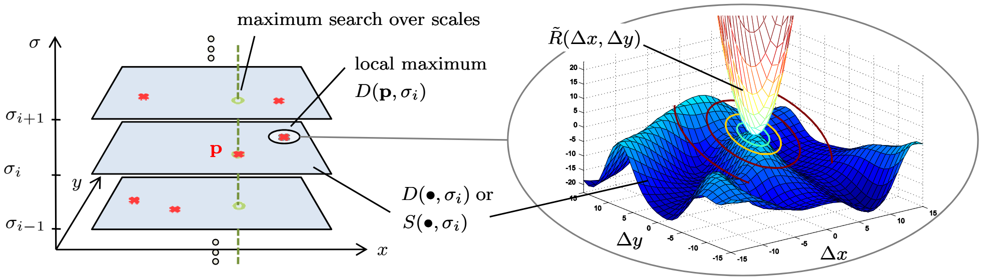


Figure 2: (left) detection stack in scale space; (right) detection operator response and cost function at one detected feature point.

in scale space $(\mathbf{x}, \sigma_i)$ the image neighbourhood $\mathscr{N}_\mathbf{x}$ is taken into consideration. For each layer $D(\bullet, \sigma_i)$ of the stack local maxima are then detected at position $\mathbf{p}$ via a non-maximum suppression approach, leading to a first set of feature points $\mathbb{P}_1$:

$$D(\mathbf{x}, \sigma_i) = \mathbf{f_{dec}}\left(I(\mathscr{N}_\mathbf{x}), \sigma_i\right) \tag{1}$$

$$\mathbb{P}_1 := \bigcup_{i=1}^{N} \left\{ \langle \mathbf{p}, \sigma_i \rangle \,\middle|\, \mathbf{p} = \arg\max_{\mathbf{x} \in \mathscr{N}_\mathbf{p}} \left(D(\mathbf{x}, \sigma_i)\right) \right\} \tag{2}$$

Second, the algorithm selects those features from $\mathbb{P}_1$ for which the response S to the *scale-selection operator* $\mathbf{f_{sel}}$ attains a local maximum over scale (Figure 2 left). Points for which the scale-selection operator attains no extremum or the response is below a threshold τ are rejected:

$$S(\mathbf{p}, \sigma_i) = \mathbf{f_{sel}}\left(I(\mathscr{N}_\mathbf{p}), \sigma_i\right) \tag{3}$$

$$\mathbb{P}_2 := \left\{ \langle \mathbf{p}, \sigma \rangle \,\middle|\, \langle \mathbf{p}, \sigma \rangle \in \mathbb{P}_1,\ \sigma = \arg\max_{\forall \sigma_i} (S(\mathbf{p}, \sigma_i)),\ S(\mathbf{p}, \sigma) > \tau \right\} \tag{4}$$

The selected scale indicates the scale at which a maximum detector response to the local image structure is observed. It is relatively independent of the image resolution and is related to the structure and not to the resolution at which the structure is represented.

One can see that for interest point localisation only the feature detection operator and by this means the created scale-space stack D is of importance. The detection process is accomplished by a maximum search for the characteristic scale. Thus, for the evaluation of a detection error the particular layer $D(\bullet, \sigma)$ of the detection pyramid is the determining factor. Maximising the operator output is equal to minimising the cost function $R(\Delta\mathbf{p})$ in (5). Within a small neighbourhood $\Delta\mathbf{p} = (\Delta x, \Delta y) \in \mathscr{N}_\mathbf{p}$ we can approximate $R(\Delta\mathbf{p})$ via a Taylor expansion up to second order for feature point $\langle \mathbf{p}, \sigma \rangle$ (see also Figure 2 right):

$$R(\Delta\mathbf{p}) = |D(\mathbf{p}, \sigma) - D(\mathbf{p} + \Delta\mathbf{p}, \sigma)| \approx \tilde{R}(\Delta\mathbf{p}) = \frac{1}{2} \Delta\mathbf{p}^\top \mathbf{H} \Delta\mathbf{p}. \tag{5}$$

Model point and first derivative vanish, while the Hessian $\mathbf{H}$ characterizes the curvature at the interest point $\mathbf{p}$. Simply speaking, for a low curvature the detection process will imply an error due to the missing discriminative behaviour of $D(\bullet, \sigma_p)$ in the neighbourhood $\mathscr{N}_\mathbf{p}$, whereas for a high curvature, the spacial detection process will be more accurate. Following

the argumentation in [7], it makes sense to regard the inverse of the Hessian $\mathbf{H}$ as a measure for feature localisation uncertainty.
Therefore, we decided on taking the inverse of the Hessian as our covariance estimate. The estimation process then happens in two steps:

1. Estimate the covariance for each interest point $\langle \mathbf{p}, \sigma \rangle$ according to

$$\Sigma = \mathbf{H}^{-1} = \begin{bmatrix} R_{xx}(\mathbf{p},\sigma) & R_{xy}(\mathbf{p},\sigma) \\ R_{xy}(\mathbf{p},\sigma) & R_{yy}(\mathbf{p},\sigma) \end{bmatrix}^{-1} = \mp \begin{bmatrix} D_{xx}(\mathbf{p},\sigma) & D_{xy}(\mathbf{p},\sigma) \\ D_{xy}(\mathbf{p},\sigma) & D_{yy}(\mathbf{p},\sigma) \end{bmatrix}^{-1}, \tag{6}$$

   where R_{xx}, R_{xy}, R_{yy} and D_{xx}, D_{xy}, D_{yy} are the second order derivatives at the point $\mathbf{p}$, respectively. The last term has a negative sign if the interest point relates to a maximum in the operator response and a positive sign if it is a minimum.

2. Depending on the particular creation process of the detector stack D, it may be required to propagate the covariance matrix back to the initial scale σ_0 (according to the initial image). By doing so it is ensured that covariances retain their proportional relationship. Rescaling is particularly important, if layer $D(\bullet, \sigma_i)$ does not have the same resolution as $D(\bullet, \sigma_0)$; this often is the case for computational reasons. A back projection then is done via

$$\Sigma^{(0)} = \Sigma \cdot \left(\frac{res(D(\bullet, \sigma_0))}{res(D(\bullet, \sigma_i))} \right)^2. \tag{7}$$

$\Sigma^{(0)}$ here refers to the covariance associated with a feature point $\langle \mathbf{p}, \sigma \rangle$ at position $\mathbf{p}$ in the initial image, describing its localisation precision. The proposed method is applicable to feature detection algorithms detecting points in scale space. In the following we will demonstrate how to implement the framework for SIFT and SURF.

4 Uncertainty Estimation for SIFT and SURF Features

We chose to apply our framework to SIFT and SURF as these two feature detection algorithms are widely used.

4.1 Scale Invariant Feature Transform (SIFT)

SIFT [10] is one of the most popular region detectors possessing scale-invariance. It uses the Laplacian operator for spatial feature detection *and* scale selection. To lower its complexity, the operator is approximated by a difference of Gaussians (DoG) operator. This approach allows creating the detection stack D from the difference of neighbouring layers of a Gaussian pyramid:

$$\underbrace{D(\mathbf{x}, \sigma_i)}_{:=S(\mathbf{x},\sigma_i)} = \underbrace{(G(\mathbf{x}, \sigma_{i+1}) - G(\mathbf{x}, \sigma_i))}_{\approx \nabla^2 G(\mathbf{x},\sigma_i)} * I(\mathbf{x}) \tag{8}$$

$$= G(\mathbf{x}, \sigma_{i+1}) * I(\mathbf{x}) - G(\mathbf{x}, \sigma_i) * I(\mathbf{x}) \tag{9}$$

To achieve lower memory usage the image stack introduced before is now represented by an image pyramid grouped in octaves. Between subsequent octaves, down sampling by a factor

2 is performed, which retains the same information as smoothing the image with doubled standard deviation. While an octave contains images of equal resolution, it is divided into intervals created by increasing detector size. This allows us to compute the relation from detected feature scale to the original image scale:

$$\sigma_i = \sigma_0 \cdot 2^{octave+interval/N_{intervals}}, \tag{10}$$

where $\sigma_0 = 1.6$ is defined as the smoothing strength of the very bottom pyramid layer and $N_{intervals}$ is the predefined number of intervals per octave.

Feature regions $\langle \mathbf{p}, \sigma \rangle$ are located spatially and in scale via non-maximum suppression in a $3 \times 3 \times 3$ neighbourhood for each pyramid location $(\mathbf{x}, \sigma_i)$ according to Equations (2) and (4). For a more accurate interest point localisation compared to the one obtained from the sampled scales σ_i, detected feature points are interpolated in scale space leading to a second order estimate

$$\hat{\mathbf{u}} = \begin{pmatrix} \hat{\mathbf{p}} \\ \hat{\sigma} \end{pmatrix} = \arg\max_{\hat{\mathbf{u}}} \left(D(\mathbf{u}) + \frac{\partial D^\top}{\partial \hat{\mathbf{u}}} (\hat{\mathbf{u}} - \mathbf{u}) + \frac{1}{2} (\hat{\mathbf{u}} - \mathbf{u})^\top \frac{\partial^2 D}{\partial \hat{\mathbf{u}}^2} (\hat{\mathbf{u}} - \mathbf{u}) \right) \tag{11}$$

The Laplacian operator performs well for scale selection, yet detects less meaningful points or regions (e.g. on edges) which need to be post processed. For more information the reader is referred to [10].

4.2 Speeded Up Robust Features (SURF)

SURF [1] adopts the idea of SIFT and improves the process in order to obtain a faster detection and matching. We will briefly explain the detection process, which is necessary for covariance estimation.

The algorithm relies on the usage of integral images, which accounts for most of the reduction in computation time. It employs the determinant of the Hessian matrix as spatial feature detection *and* scale selection operator, similar to the Harris-corner detector but adapted for scale-invariance. The entries of the Hessian are calculated by convolving the appropriate Gaussian second order derivatives with the image at the analysed position. SURF approximates derivatives with box filters of different sizes according to the scale. The Hessian can then be evaluated at constant, low computational cost using integral images for arbitrary filter size:

$$\underbrace{D(\mathbf{x}, \sigma_i)}_{:=S(\mathbf{x}, \sigma_i)} = \det \begin{bmatrix} L_{xx}(\mathbf{x}, \sigma_i) & L_{xy}(\mathbf{x}, \sigma_i) \\ L_{xy}(\mathbf{x}, \sigma_i) & L_{yy}(\mathbf{x}, \sigma_i) \end{bmatrix}, \tag{12}$$

where L_{xx}, L_{xy}, L_{yy} are the responses of the image convolved with the according box filter.

The scale space is analysed by up-scaling the filter size rather than iteratively reducing the image size. The smallest box filter has size 9×9 and the output is considered as the initial scale layer with scale $\sigma_0 = 1.2$. Following layers are obtained by filtering the image with gradually bigger masks. Sampled scales thus directly relate to the filter size s via

$$\sigma_i = \sigma_0 \cdot \frac{s}{9}. \tag{13}$$

The scale space is grouped into octaves as well. An octave includes a series of filter responses of equal size. In total an octave encompasses a scaling factor of 2, thus filter responses in the following octave are subsampled and are half the size. In order to localise interest points $\langle \mathbf{p}, \sigma \rangle$ in the image and across scales, non-maximum suppression in a $3 \times 3 \times 3$ neighbourhood followed by an interpolation step is applied similar to SIFT.

4.3 Covariance Estimation for SIFT and SURF

Our covariance estimation framework is easily applied to SIFT and SURF. The covariance is calculated according to Equation (6) as the inverse of the Hessian. Derivatives are calculated by taking differences of neighbouring sample points. To get a more robust estimate it is useful to increase the influence region from 3×3 to a 5×5 neighbourhood and calculate the Hessian as a Gaussian weighted sum:

$$\Sigma = \left(\sum_{i,j \in \mathcal{N}_{\mathbf{p}}} w(i,j) \cdot \begin{bmatrix} D_{xx}(i,j,\sigma_p) & D_{xy}(i,j,\sigma_p) \\ D_{xy}(i,j,\sigma_p) & D_{yy}(i,j,\sigma_p) \end{bmatrix} \right)^{-1} . \tag{14}$$

Note that the interpolation step shown in Equation (11) will lead to a detection scale $\hat{\sigma}$ which is not represented by pyramid scales σ_i. Interpolation between pyramid layers leads to $D(\hat{\mathbf{p}}, \hat{\sigma})$. Covariance estimation is performed at this characteristic scale; so at a given octave and (sub)interval, requiring back propagation of covariances to the original size according to

$$\Sigma^{(0)} = \Sigma \cdot (2^{octave})^2. \tag{15}$$

To lower the complexity, covariances can be estimated at the detection scale σ rather than at $\hat{\sigma}$ without degrading the result significantly. Using $D(\bullet, \sigma)$ as the reference layer for the Hessian calculation requires a back projection of $\Sigma^{(0)} = \Sigma \cdot (2^{octave} + (\hat{\sigma} - \sigma))^2$.

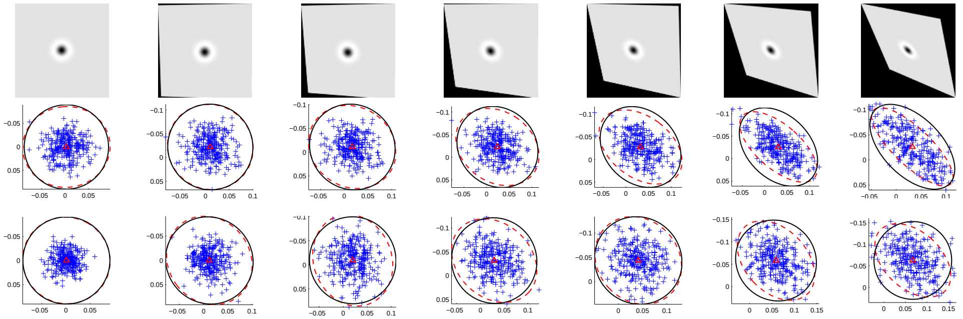
Covariances estimated in this manner can only be determined up to scale. Normalisation as suggested by [6, 7] is not reasonable in our case, as we want to preserve the proportions between covariances. Therefore, we scale all covariance matrices such that a circular feature detected in the very bottom pyramid layer will approximately have Frobenius norm 1. This constant factor has been determined experimentally. Note, that scaling is only performed for numerical reasons and does not change the influence of covariance in any way.

5 Experiments

The following contains a description of the experiments we carried out to ensure our uncertainty estimates are related to the real underlying location error distribution. First we generate samples of the localisation error in order to obtain a quantitative measurement for the detector accuracy. Second, we compare the proposed covariance estimates to the created error distribution and evaluate the accuracy of the estimates.

The idea behind the statistical error sampling is to create synthetic images with which it is possible to control the ground truth location of feature points. For detected feature points in these images we are then able to measure the localisation error. SIFT uses a difference of Gaussians (DoG) as detection operator. For controlling the feature point location this means, that the output of the image and operator convolution has to be maximum at the specified ground truth location. It is achieved via a matched filter approach by placing a DoG itself at the desired feature location. The detection scale is influenced by appropriate choice of the DoG in the image according to Equation (10). For SURF a matched filter approach is not feasible, because the determinant of the Hessian is a nonlinear detection operator. Yet, it will generate maximum response at the centre of a bivariate Gaussian.

To build up a localisation error distribution, repeating the detection several times does not give the desired result, since the detection process is deterministic. By adding pixel noise in the original image we expect the localisation error to change as well. In order to test



(a) 0° (b) 10° (c) 20° (d) 30° (e) 40° (f) 50° (g) 60°

Figure 3: Distribution of localisation error (plus sign) and comparison of maximum likelihood estimate (dashed line) to our Hessian based covariance estimate (solid line) for different viewpoints (a) - (g) and detectors SIFT (top) and SURF (bottom).

the influence of viewpoint changes we additionally warp the initial synthetic image with a perspective transformation.

Viewpoint change	0°	10°	20°	30°	40°	50°	60°
	Bhattacharyya distance($\cdot 10^3$)						
SIFT	0.181	0.850	0.955	2.72	7.94	32.9	50.2
SURF	0.391	0.411	0.449	1.17	3.57	16.9	28.4

Table 1: Covariance comparison between ML and our estimate

Results for the error modelling are shown in Figure 3. The maximum likelihood estimate of the sampled error distribution and our covariance estimates are compared to each other via the Bhattacharyya distance [2]. A normalization before comparison is necessary as the estimated covariances can only be determined up to scale. Table 1 lists the results for varying viewpoint changes. Note that the error distribution for SURF does not depend that much on the feature shape compared to SIFT. Calculation of our covariance estimate as a weighted sum in the interest point neighbourhood results in a more circular shape, but guarantees more stable estimates.

From our evaluation we conclude that the covariance estimate does represent the underlying localisation error distribution. Figure 4 displays the change of the covariance norm over the related detection scale. The curves show that features detected at higher scales are less accurate compared to features detected at lower scales. This is intuitive as layers in the detection pyramid corresponding to higher scales are built from more blurred or sub sampled versions of the original image. This loss of information is the reason for the increasing localisation error.

6 Results for Model Fitting

While we have investigated the correctness of our covariance estimates in the previous section, now the goal is to verify that usage of those really improves the performance of current algorithms. Bundle adjustment and homography estimation are the two applications which we will discuss in the following. The concept of covariance incorporation is alike for both

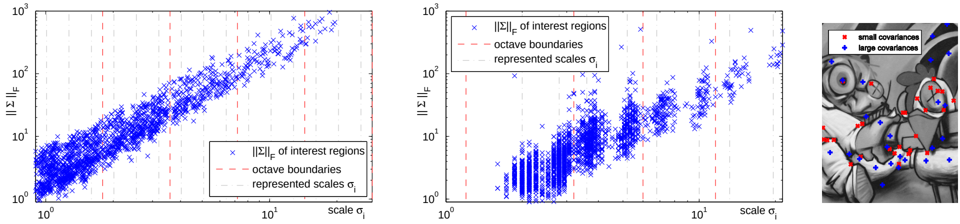


Figure 4: Frobenius norm of estimated covariances for interest regions detected in real images; SIFT (left) and SURF (middle). (right) illustrates the relation between the distinctiveness of a feature and its covariance.

model fitting approaches. They state optimization methods, which try to minimise a least square problem in the form of:

$$\arg\min_{\mathbf{x}} \mathbf{y}(\mathbf{x})^\top \mathbf{y}(\mathbf{x}) \tag{16}$$

In computer vision the vector $\mathbf{y}$ is composed of single observations $\mathbf{y}_{ij}$ for point j in image i and is calculated as the difference between a known image point $\mathbf{p}_{ij}$ and a mapped model point $w(\mathbf{T}_i\mathbf{M}_j)$, where $\mathbf{T}$ defines the mapping, $\mathbf{M}$ refers to the model and w is a warping function. For the Euclidian norm $\mathbf{y}_{ij}$ becomes

$$\mathbf{y}_{ij} = \mathbf{p}_{ij} - w(\mathbf{T}_i\mathbf{M}_j). \tag{17}$$

Considering covariances, minimisation of the Euclidian distance results in minimising the Mahalanobis distance. Thereby terms with large covariances maintain less influence on the overall cost reduction, resulting in a weighted least square optimization with

$$\mathbf{y}_{ij} = \Sigma_{ij}^{-\frac{1}{2}} \left(\mathbf{p}_{ij} - w(\mathbf{T}_i\mathbf{M}_j)\right). \tag{18}$$

Bundle adjustment simultaneously refines the 3D coordinates describing the scene geometry as well as camera poses and intrinsic camera parameters. A comprehensive introduction can be found in [17]. Given a set of images representing a scene from different viewpoints and their corresponding image feature points, bundle adjustment tries to minimize the reprojection error of 3D points in all images. The projection function from Equations (17) and (18) is $\mathbf{T} = \mathbf{K}[\mathbf{R}\ \mathbf{t}]$, while the model parameters $\mathbf{M}$ refer to the 3D points.

The scene we use for bundle adjustment is created synthetically, so its geometry is known beforehand. It consists of four parallel quadratic image patches located at different depths from the camera centre (Figure 5 left). The scene is captured from varying viewpoints with known camera matrix $\mathbf{K}$ and poses $[\mathbf{R},\mathbf{t}]$ and feature points including their covariance estimates are detected in each of the images. An initial estimate of the 3D structure and camera poses is created from 2 images. Finally, we compute the target reprojection error between the known corner points $\bar{\mathbf{c}}$ and projected 3D corner points $\bar{\mathbf{C}}$ by means of the estimated mapping parameters:

$$e = \frac{1}{n_c}\sum_{i=1}^{n_c} \left\|\bar{\mathbf{c}} - w(\hat{\mathbf{T}}\bar{\mathbf{C}})\right\|. \tag{19}$$

For our simulations we use the sparse bundle adjustment framework provided by Lourakis [9]. Table 2 summarises the performance improvement of bundle adjustment with covariances employed. Note that the reprojection error is smaller for smaller patches, due to more distinctive feature points with smaller covariances detected at smaller patches.

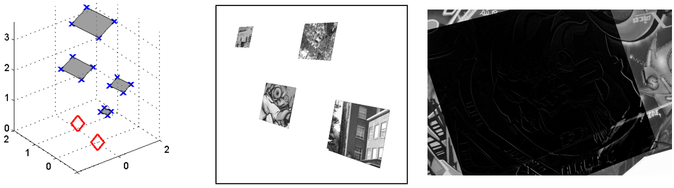


Figure 5: Bundle adjustment: (left) artificial setup for bundle adjustment, (middle) example image for one camera position; Homography estimation: (right) overlay of images using the homography estimated with covariances.

	mean all patches		smallest patch		largest patch	
covariance usage	no	yes	no	yes	no	yes
SIFT	2.031	1.759	1.941	1.672	2.088	1.828
SURF	2.554	2.363	2.518	2.292	2.631	2.464

Table 2: Reprojection error for bundle adjustment with and without covariance estimates used. Values indicate the mean performance as pixel offset for about 100 different image pairs. Smallest and largest patch refer to the patch size seen in the images.

We also applied the estimated covariances for homography computation. We compute a homography between two images based on their feature point correspondences. In this case the mapping function reads as the homography itself: $\mathbf{T} = \mathbf{H}$. The model parameters $\mathbf{M}$ are the feature points in the images associated with their covariances, which stay unchanged during the optimization. Figure 5 (right) presents some qualitative results using SIFT and the Oxford dataset.
Evaluation of the mean difference of pixel intensity values indicates that using covariance information does not result in a more accurate estimate compared to not using covariances; estimated homographies are rather equally good. Still our covariance estimates prove to be correct, otherwise we would observe worse an probably unstable results.

7 Conclusion and Future Work

In this paper, we have presented a novel framework for estimating location uncertainty for scale invariant feature points. We have shown that the covariance of the localisation error can be calculated from the detector response map in the neighbourhood of a feature point without significant computational overhead. Consequently, covariances differ according to the particular detection scale and interest region shape. We have implemented the proposed framework for SIFT and SURF and verified that our covariance estimates relate to the real underlying error distribution. Furthermore, we used this covariance information in model fitting and have shown a performance improvement for bundle adjustment.
Future work should include the application of our framework to other multi-scale feature detectors and investigate if the covariance information could also be used to construct more robust descriptors.

Acknowledgment We would like to thank Marc Pollefeys for the inspiring discussion about the normalization behaviour of our covariances for differently sized images.

References

[1] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool. Speeded-up robust features (SURF). *Computer Vision and Image Understanding*, 110(3):346–359, 2008.

[2] A. Bhattacharyya. On a measure of divergence between two statistical populations defined by probability distributions, vol. 35. *Bulletin of the Calcutta Mathematical Society*, 1943.

[3] M.J. Brooks, W. Chojnacki, D. Gawley, and A. van den Hengel. What value covariance information in estimating vision parameters? In *Proceedings IEEE International Conference on Computer Vision (ICCV) 2001*, volume 1, Vancouver, British Columbia, Canada, July 7-14 2001.

[4] A. Haja, B. Jähne, and S. Abraham. Localization accuracy of region detectors. In *Proceedings IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR) 2008*, pages 1–8, Anchorage, AK, USA, June 23-28 2008.

[5] C. Harris and M. Stephens. A combined corner and edge detector. In *Alvey Vision Conference*, volume 15, page 50, Manchester, UK, August 31 - September 2 1988.

[6] K. Kanatani. Uncertainty modeling and model selection for geometric inference. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 26(10):1307–1319, 2004.

[7] Y. Kanazawa and K. Kanatani. Do we really have to consider covariance matrices for image features? In *Proceedings IEEE International Conference on Computer Vision (ICCV) 2001*, volume 2, Vancouver, British Columbia, Canada, July 7-14 2001.

[8] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Proceedings IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR) 2006*, New York, NY, USA, June 17-22 2006.

[9] M.I.A. Lourakis and A.A. Argyros. The design and implementation of a generic Sparse Bundle Adjustment software package based on the Levenberg-Marquardt algorithm. Technical Report 340, Institute of Computer Science - FORTH, Heraklion, Greece, Aug. 2004. Available from `http://www.ics.forth.gr/~lourakis/sba`.

[10] D.G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision (IJCV)*, 60(2):91–110, 2004.

[11] K. Mikolajczyk and C. Schmid. Scale & affine invariant interest point detectors. *International Journal of Computer Vision (IJCV)*, 60(1):63–86, 2004.

[12] 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 (IJCV)*, 65(1):43–72, 2005.

[13] U. Orguner and F. Gustafsson. Statistical characteristics of Harris corner detector. In *Proceedings IEEE Workshop on Statistical Signal Processing (SSP), 2007*, pages 571–575, Madison, WI, USA, August 26-29 2007.

[14] C. Schmid, R. Mohr, and C. Bauckhage. Evaluation of interest point detectors. *International Journal of Computer Vision (IJCV)*, 37(2):151–172, 2000.

[15] N. Snavely, S. M. Seitz, and R. Szeliski. Modeling the world from Internet photo collections. *International Journal of Computer Vision (IJCV)*, 80(2):189–210, 2008.

[16] R.M. Steele and C. Jaynes. Feature uncertainty arising from covariant image noise. In *Proceedings IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR) 2005*, volume 1, San Diego, CA, USA, June 20-15 2005.

[17] B. Triggs, P.F. McLauchlan, R.I. Hartley, and A.W. Fitzgibbon. Bundle adjustment - A modern synthesis. *Lecture Notes in Computer Science*, pages 298–372, 1999.

[18] L. Vacchetti, V. Lepetit, and P. Fua. Stable real-time 3d tracking using online and offline information. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 26(10):1385–1391, 2004.

[19] C. Wu, B. Clipp, X. Li, J.-M. Frahm, and M. Pollefeys. 3D model matching with viewpoint invariant patches (VIPs). In *Proceedings IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR) 2008*, Anchorage, AK, USA, June 23-28 2008.