Localization Error Measurement in Features Extraction Algorithms

December 21, 2010

Abstract

Image features detection and matching is a fundamental step in many computer vision applications. A lot of methods have been proposed in recent years, with the aim to extract image features invariant to a group of transformations. Even if the state-of-art has not achieved the full invariance, many methods, like SIFT, Harris-affine and Hessian-affine combining a robust descriptor, give sufficient invariance for some practical applications. In contrast to the advance in the invariance of feature detectors, the detection precision has not been paid enough attention even if the repeatability and stability are extensively studied. In this report, we focus on the SIFT method and measures its localization precision by average localization error under a given geometric transform and localization uncertainty by covariance matrix. These measurements can be easily extended to other feature detectors. For those which are scale invariant, it can be shown that the localization error and localization uncertainty both increase with the scale of features.

1 Introduction

Recent years see the blossom of invariant feature points (regions) detector. Even though they are more and more invariant to a group of image transformations and illumination change, the detection error is always ignored and considered isotropic. In fact the isotropy of localization error highly depends on the type of feature detector used and the parameters set in the detection process. It is not appropriate to say a feature detector is good if the detected features has homogeneous isotropic localization error. In fact, different feature detectors extract different types of features and they are complementary. For example, Harris corner-based detectors often select features on the corners, which have homogeneous isotropic localization error. Interestingly, human perception system have the tendency to choose this kind of features [17]. In contrast, DoG-based or Hessian-based detector prefer blob-like regions, which usually do not have isotropic localization error. In particular, for some scale-invariant feature detectors, their localization error increases with the scale [3, 43], for example SIFT [21] and SURF [12]. By reviewing different feature detectors and descriptors, we get the impression that almost all the detectors emphasize on the feature selection or affine adapted neighbor estimation without paying attention to the detection precision. Thus the detection precision is intrinsically decided by the detection process itself and the image noise.

1.1 Feature detection

A typical input of many computer vision problems are a set of corresponding features between two or several images, like camera calibration, image mosaicking, object retrieval and 3D scene modeling. This raises the question how to extract features from image and match them correctly. This problem can be divided into three steps: feature detection, descriptor construction and descriptor matching. The first two steps are closely related because some features detection algorithms also give a region adapted to the image local structure, which is a good candidate to construct a invariant descriptor. The third step is somewhat independent to the first two.

To detect feature points/regions in image, an appropriate mathematic operator is applied on the image and the response is thresheld to find the stable features. Harris corner [11] is one of the first proposed feature point detector. This detector computes auto-correlation matrix on each point in image, which captures the local image structure. A criteria called *cornerness* is proposed to select the points whose auto-correlation matrix has two big eigenvalues. And the feature points are considered as the local maxima of *cornerness*. Harris corner achieves rotation invariant by estimating the principal gradient direction of the corner. But it is not scale invariant since a fixed-size image patch around the extracted corner is used to construct the descriptor. Several variants based on Harris corner has been proposed. Multi-scale Harris [19] is simply to apply Harris corner on images differently blurred by Gaussian convolution. But it is still not scale invariant due to the fixed-size descriptor. Harris-Laplace [18, 19] obtains the scale invariance by an iterative selection of the characteristic scale by LoG (Laplace-of-Gaussian) operator for the feature points extracted by multi-scale Harris. It is shown that multiple feature points for the same image structure will converge to the same point and scale. Harris-Affine [18, 19, 39] is invariant to affine image transformation. It begins with multi-scale Harris corners and iteratively refines the location and estimates the affine neighborhood around the point by some *affine adaptation* process based on the second moment matrix. Hessian-Laplace [18, 19] (and Hessian-Affine) is similar to Harris-Laplace (and Harris-Affine) except that points are localized in space at the maxima of the Hessian determinant. So Hessian-Laplace is scale and rotation invariant and Hessian-Affine is affine invariant. Some other detectors are also designed to be affine invariant, like an edge-based region detector [47, 45], an intensity-based region detector [46, 47], an entropy-based region detector [44], and two independently developed level line-based region detectors MSER ("maximally stable extremal region") [14] and LLD ("level line descriptor") [31, 33, 32]. But all of them are not yet fully affine invariant since *they start with initial feature scales and locations selected in a non-affine invariant manner* [21].

Efforts are also dedicated to design distinctive and robust descriptors to achieve affine invariance, such as distribution-based descriptors [15, 37, 48, 21, 1, 36], descriptors based on spatial-frequency techniques [10], differential descriptors [5, 38] and moment-based descriptors [20]. But the fully affine invariance is not yet obtained according to the extensive comparison [16]. Recently a fully affine invariant region detector (ASIFT) is proposed by Morel and Yu [26]. Their method is based on Lowe's method [21] but simulates two missing parameters left over by Lowe: latitude and longitude (two angles dening the camera axis orientation). They mathematically proved that their method is

fully affine invariant, up to an arbitrary precision. For descriptor construction, ASIFT follows SIFT method, which is almost the best with respect to the other popular descriptors [16]. ASIFT has the assumption that 3D scene is piecewise planar such that any deformation introduced by camera motion can be locally well approximated by an affine transformation. This assumption is true for most of natural scenes, but there still exist some scenes with strong 3D structure for which ASIFT fails. It is worth to mention SIFT method here since it is the first fully similarity invariant region detector combined with a distinctive and robust descriptor. SIFT method can be recapitulated as follows: for each feature selected as the local extrema in 3D scale space approximated by difference of Gaussian, a local image patch around this point along the dominant gradient is extracted to construct the descriptor, which is a 3D histogram of gradient location and orientation weighted by gradient magnitude. The robustness of SIFT descriptor makes it also partially invariant to illumination change and affine transformation.

The last step is to match descriptors. There are usually three types of strategies: distance threshold, nearest neighbor distance threshold, Lowe's nearest neighbor distance ratio [21]. Lowe's nearest neighbor distance ratio performs well in practice when there is no or unique matching descriptor for one descriptor. Recently an *A Contrario* matching criterion [35] is proposed to deal with the situation where there are multiple matching descriptors.

1.2 Localization uncertainty

In this section, some previous work about the localization uncertainty is reviewed. It is necessary to distinguish "localization uncertainty" from "localization error" to make the following context more clear.

Localization error means the absolute error in the detected feature position, while localization uncertainty indicates the variation of error introduced in the feature detection process. Localization uncertainty is a statistical term, which is usually measured by covariance matrix. It depends only on the detection process itself and can be evaluated on one image directly. But localization error is always evaluated on a pair of images in previous work. The evaluation is performed in the common framework that the ground truth transformation between two images is known. In [22], localization error is evaluated at different scales for multi-scale Harris corners and the average error is reported about $1 \sim 3$ pixels. In [19, 16], repeatability is used to evaluate the localization error under different image transformation or illumination change for scale-invariant detectors. In [7], more attention is paid to the localization error of Harris-like detectors in terms of repeatability and information content. These evaluations give an average localization error for a group of matched features. We remark that the evaluation depends not only on the feature detection but also on the feature matching. The distinctiveness and robustness of descriptor against the undergone image transformation or illumination change plays also an important role. Different errors besides detection precision are mixed in the evaluation. Thus the evaluation does not only reflect the detection precision.

The complementary work to localization error is the localization uncertainty measured by covariance matrix for each detected feature individually. In [17], the author used two methods to estimate the covariance matrix for corner feature and concluded that the accuracy of geometric computation is not improved by

incorporating the covariance matrix into optimization since covariance matrix seems to be homogeneous isotropic. This is normal because the detector they tested selects always corner-like features. In contrast, Brooks *et al.* [24] observed accuracy improvement in fundamental matrix computation by incorporating the estimated covariance matrix of Harris corner. Steele and Jaynes [42] 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. Orguner and Gustafsson [29] 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. For scale-invariant region feature, the covariance matrix has different property from corner feature [3]. 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.

Localization uncertainty only depends on the covariance matrix, while localization error is influenced by more factors like the gradient, image blur and feature matching process. For scale-invariant feature detectors with sub-sampling, the localization error and uncertainty both increase through scales. We proposed to cancel the sub-sampling to obtain some improvement. In theory, for continuous infinite resolution images, neither the localization uncertainty nor the localization error will be improved by this modification. But in practice, only digital images can be used and the computation will be more precise if the sub-sampling is canceled. The improvement can be observed for localization error if two images contain the same blur, while localization uncertainty does not change a lot in any case. This phenomenon will be analyzed and explained. In practice, two images do not always contain the same blur, which leads to estimate the quantity of blur of both images and blur again the image which is less blurry. This report begin with a review of SIFT method in section 2, which makes the concept of localization error and localization uncertainty more concrete. Synthetic test of localization error is shown in section 3 for Lowe's SIFT and improved SIFT with some comparison and analysis. A method to estimate and align the blur of two images is also proposed. In section 4, different methods to compute localization uncertainty are compared. It is shown that localization uncertainty increases through scale space if the image is well-sampled, whether the sub-sampling is canceled or not.

2 SIFT method [21]

SIFT method is one of the most widely used feature region detectors. It is a good candidate for our analysis of localization error/uncertainty because of its full scale invariance. SIFT method is a complete algorithm including scale-invariant feature detector, gradient-based descriptor and descriptor matching based on nearest neighbor distance ratio. The scale-invariant feature detector relies on 3D scale-space implemented by difference of Gaussian due to its computational

efficiency:

$$\begin{aligned} D(x,y,\sigma) &= \Big(G(x,y,k\sigma) - G(x,y,\sigma)\Big) \circledast I(x,y) \\ &\approx (k-1)\sigma^2 \Delta\Big(G(x,y,\sigma) \circledast I(x,y)\Big) \end{aligned} \tag{1}$$

This means the Laplacian is approximated by the different of Gaussian. Remark that the Laplacian is normalized by factor $(k-1)\sigma^2$, which in fact gives scale invariance to the Laplacian threshold in SIFT method. The SIFT method gives stable performance when k is smaller than $\sqrt{2}$. To get more efficiency, a sub-sampling by 2 is also added in scale-space. All the images with the same size belong to a common octave in scale space. SIFT scale space consists of several octaves (Fig. 1): one octave contains $Ninter + 3$ Gaussian blurred images with the same resolution which are used to compute $Ninter+2$ difference-of-Gaussian images. The local extrema is only detected on the $Ninter$ in the middle ($Ninter$ is the number of intervals in an octave, the default value is 3). The Gaussian blur is increased with the multiplicative factor $2^{1/Ninter}$ and thus $k = 2^{1/Ninter}$. The 2-subsampling is performed on the image in octave which contains two times the blur of the initial image of the same octave. This convolution and 2-subsampling procedure is repeated until the image is too small for feature detection. It is easy to see that the sampling with respect to the blur is the same for all octaves. So one image has the same nature as its counterparts in the other octaves. This process simulates camera zoom-out and explains why SIFT method is scale invariant.

With $\sigma_2 = 2\sigma_1$ in Eq. (1), the normalization factor has the relationship: $(k-1)\sigma_2^2 = 4(k-1)\sigma_1^2$. This implies that $\Delta\Big(G(x,y,\sigma_2) \circledast I(x,y)\Big) = \frac{1}{4}\Delta\Big(G(x,y,\sigma_1) \circledast I(x,y)\Big)$. Is it true for all $I(x,y)$ and pair (σ_1,σ_2) satisfying $\sigma_2 = 2\sigma_1$?

In 3D scale-space, features are selected as local extrema by comparing with 26 neighbors (see Fig. 1). Once a feature is extracted, its 3D position is refined by a 3D interpolation, where comes from SIFT sub-pixel precision. Each feature is assigned a principal direction by using the gradient direction and magnitude in the neighborhood. A fixed-size (16×16 pixel) region around image feature along its principal direction is extracted to construct the descriptor. This region is divided into 4×4 sub-regions; in each sub-region, an orientation histogram containing 8 directions is created by quantizing the gradient direction of each sample weighted by gradient magnitude (Fig. 2). To make the detected features useful, their 3D coordinate (location and scale) should be propagated back to the original image.

2.1 Blur

Blur issue is important because the scale invariance in SIFT method is in fact blur invariance. SIFT scale space is a representation of image with increasingly Gaussian blur if it is viewed under the same resolution. SIFT method is based on the assumption that Gaussian convolution can well approximate the blur introduced by camera system and gives an aliasing-free image sub-sampling. In [27], it shows that a well-sampled image contains Gaussian blur about $\beta = 0.8$ and an aliasing-free t-subsampling should be preceded by a Gaussian blur about $\beta \times \sqrt{t^2 - 1}$.

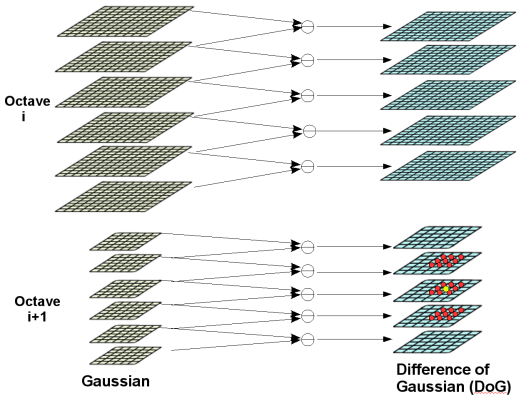

Figure 1: Pyramid-like SIFT scale space: feature is selected as local extrema (yellow point) by comparing 26 neighboring samples (red point).

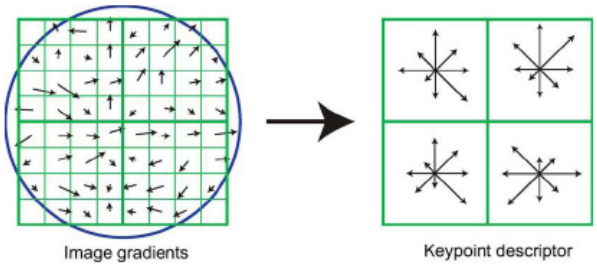

Figure 2: Descriptor is constructed on a square region around feature whose whose side direction is given by the principal gradient direction. Example of a 2×2 descriptor array of orientation histograms (right) computed from an 8×8 set of samples (left). The orientation histograms are quantized into 8 directions and the length of each arrow corresponds to the magnitude of the histogram entry.

The above discussion concerns about the aliasing-free sub-sampling. However, the blur condition becomes different for up-sampling case. This argument is based on the following simple equations:

$$\Delta\left(u(\frac{x}{2},\frac{y}{2})\right) = \frac{1}{4}(\Delta u)\left(\frac{x}{2},\frac{y}{2}\right) \tag{2}$$

$$\frac{\partial\left(u\left(\frac{x}{2},\frac{y}{2}\right)\right)}{\partial\,\bullet} = \frac{1}{2}\frac{\partial u}{\partial\,\bullet}\left(\frac{x}{2},\frac{y}{2}\right) \tag{3}$$

which means the Laplacian is 4 times smaller and gradient is 2 times smaller if an image is up-sampled by 2. Remark that the Laplacian is computed by finite difference schema, but not by Eq. (1), which compensates the factor 1/4. This relationship is only valid when image u is smooth enough. Fig. 3 shows a test for a natural image. The image is first convolved by a Gaussian blur, then it is up-sampled by factor 2. The Laplacian and gradient module are computed on the original image and the up-sampled image respectively. The average and standard deviation of the ratio of Laplacian m and that of the ratio of gradient module n are computed:

$$m = \frac{(\Delta u)\left(\frac{x}{2},\frac{y}{2}\right)}{\Delta\left(u(\frac{x}{2},\frac{y}{2})\right)} \tag{4}$$

$$n = \frac{\sqrt{\left(\frac{\partial u}{\partial x}(\frac{x}{2},\frac{y}{2})\right)^2 + \left(\frac{\partial u}{\partial y}(\frac{x}{2},\frac{y}{2})\right)^2}}{\sqrt{\left(\frac{\partial u(\frac{x}{2},\frac{y}{2})}{\partial x}\right)^2 + \left(\frac{\partial u(\frac{x}{2},\frac{y}{2})}{\partial y}\right)^2}} \tag{5}$$

It is shown that Eq. (2) and (3) satisfy only if the added Gaussian blur is at least about $\beta = 1.6$. This make the image blur become $\sqrt{1.6^2 + 0.8^2} \approx 1.8$. This experiment is complementary to the one dealing with aliasing-free sub-sampling in [27]. In Lowe's SIFT, to increase the number of features, a pre-zoom by 2 is used. For an image containing Gaussian blur 0.8, a 2-upsampling increases the blur to be $\beta = 1.6 = 0.8 \times 2$, which is close to 1.8.

2.2 3D location refinement

Once the local extrema are extracted in 3D scale space, their position can be refined under the assumption that image can be locally approximated by 2-order Taylor expansion. Given a local extrema located at $\mathbf{x} = (x, y, \sigma)$, the DoG function $D(\mathbf{x})$ is expanded at $\mathbf{x}$ by:

$$D(\mathbf{x} + \mathbf{\Delta x}) = D(\mathbf{x}) + \mathbf{\Delta x}^T \frac{\partial D}{\partial \mathbf{x}} + \mathbf{\Delta x}^T \frac{\partial^2 D}{\partial \mathbf{x}^2}\mathbf{\Delta x} \tag{6}$$

The peak of this function is attained when its derivative is set to be zero, which gives the offset $\mathbf{\Delta x} = (\Delta x, \Delta y, \Delta\sigma)^T$:

$$\mathbf{\Delta x} = (\Delta x, \Delta y, \Delta\sigma)^T = \left(\frac{\partial^2 D}{\partial \mathbf{x}^2}\right)^{-1} \frac{\partial D}{\partial \mathbf{x}} \tag{7}$$

Sub-pixel precision is obtained with this interpolation and the final position is $\mathbf{x} + \mathbf{\Delta x}$. The refined scale with respect to the original image resolution is

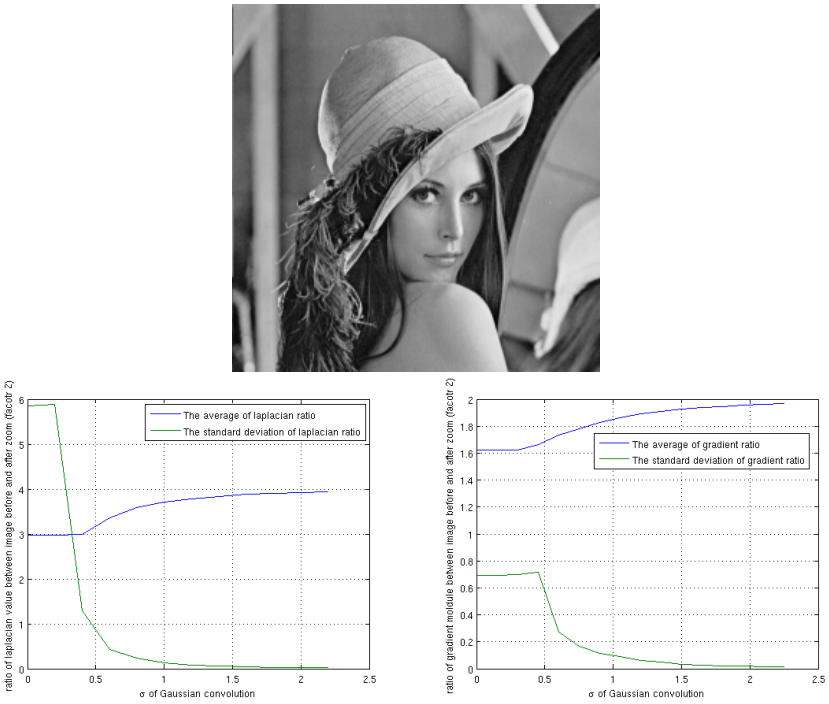

Figure 3: The image at top is convolved by a Gaussian function with standard deviation σ before it is up-sampled by factor 2. The Laplacian value and gradient module before and after the up-sampling are compared. Bottom left: the average and standard deviation of ratio of the Laplacian value before and after 2-upsampling. Bottom right: the average and standard deviation of ratio of gradient module before and after 2-upsampling.

$(\sigma + \Delta\sigma) \cdot 2^{oct}$ with $\sigma = \sigma_0 \cdot 2^{inter/Ninter}$ (σ_0 is the blur of the first image in one octave, *oct* is the octave index, *inter* is the interval index and $Ninter$ is the total number of intervals in one octave). This refinement can also be extended to other feature detectors because the method is relatively independent on the detector. The gradient and Hessian matrix in Eq. (6) is computed by finite difference schema from neighboring sample points. This method can be imprecise if the image is not locally 2-order. A more robust method is to estimate gradient and Hessian by least square minimization [34].

2.3 Improvement

We remark that the biggest error source in SIFT method is that the detected features in scale space are projected back to the original image. Assume a feature located at $\mathbf{x}$ with ideal position $\mathbf{x_0}$ disturbed by the absolute error ε: $\mathbf{x} = \mathbf{x_0} + \varepsilon$. If this feature is detected in the i-th octave in SIFT scale space, then its final position is $2^i\mathbf{x} = 2^i\mathbf{x_0} + 2^i\varepsilon$. The error is increased by the factor 2^i. This factor has no influence only if $\varepsilon = 0$. This gives the inspiration to cancel the sub-sampling between octaves. The new schema is shown in Fig. 4. Although this seems to be an one-step modification to SIFT method, there are some details to look over.

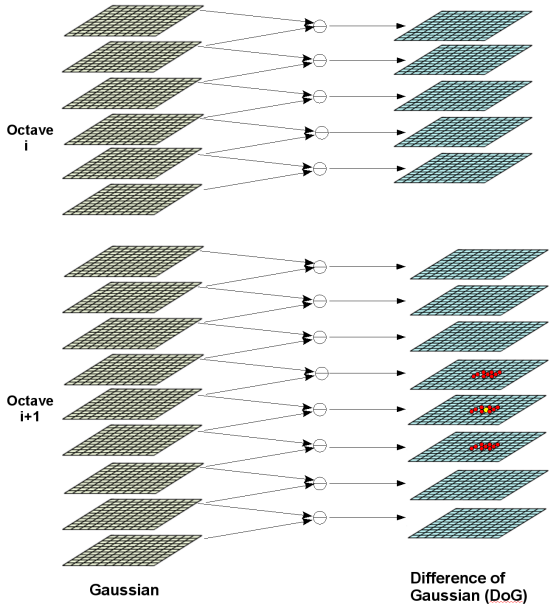

Figure 4: Improved SIFT scale space with sub-sampling canceled. The number of intervals in octave is increased through octaves.

First the Laplacian threshold, the most important threshold in SIFT to select stable features. It is kept constant in SIFT method because it is scale invariant. But with sub-sampling canceled, this threshold should be decreased through octaves since images becomes more and more blurred. Compared to original

SIFT scale space structure, the modified scale space have images equivalently zoomed by 1, 2, 4, $\cdots$ through octaves. Thus according to Eq. (2), if the images contain Gaussian blur bigger than 1.6, the Laplacian threshold should be divided by 1, 4, 16, $\cdots$. But the Laplacian in Eq. (2) is computed by finite difference schema, which is not scale invariant. In fact, if the finite difference schema is used in SIFT, it will be difficult to determine Laplacian threshold even in one octave because Laplacian decreases with the increase of blur in image. Scale normalized Laplacian in Eq. (1) enables an uniform Laplacian threshold through octaves when the 2-subsampling is canceled. This can be shown by upsampling a DoG function $D(x, y)$ at a certain scale σ by factor 2:

$$\begin{aligned} D(\tfrac{x}{2}, \tfrac{y}{2}, \sigma) &= \Big(G(\frac{x}{2}, \frac{y}{2}, k\sigma) - G(\frac{x}{2}, \frac{y}{2}, \sigma)\Big) \circledast I(\frac{x}{2}, \frac{y}{2}) \\ &= \Big(G(x, y, 2k\sigma) - G(x, y, 2\sigma)\Big) \circledast I(\frac{x}{2}, \frac{y}{2}) \\ &\approx (k-1)(2\sigma)^2 \Delta\Big(G(x, y, 2\sigma) \circledast I(\frac{x}{2}, \frac{y}{2})\Big) \\ &= (k-1)(2\sigma)^2 \Delta\Big(I(\frac{x}{2}, \frac{y}{2})\Big) \circledast G(x, y, 2\sigma) \\ &= (k-1)(2\sigma)^2 \frac{1}{\alpha}(\Delta I)(\frac{x}{2}, \frac{y}{2}) \circledast G(x, y, 2\sigma) \\ &= (k-1)\frac{4}{\alpha}\sigma^2(\Delta I)(\frac{x}{2}, \frac{y}{2}) \circledast G(x, y, 2\sigma) \\ &= (k-1)\frac{4}{\alpha}\sigma^2 \Delta\Big(G(\frac{x}{2}, \frac{y}{2}, \sigma) \circledast I(\frac{x}{2}, \frac{y}{2})\Big). \end{aligned} \tag{8}$$

According to Eq. (2), $\alpha = 4$ when a blur bigger than 1.6 is added to image. Then, $D(\frac{x}{2}, \frac{y}{2}, \sigma) = (k-1)\sigma^2\Delta\Big(G(\frac{x}{2}, \frac{y}{2}, \sigma) \circledast I(\frac{x}{2}, \frac{y}{2})\Big)$, which means the Laplacian value does not change through octaves.

Second, Lowe's SIFT descriptor is constructed from a region with fixed size 16×16 around extracted features. There is no problem if the scale change between two compared images is 2^i $(i \in \mathcal{N})$ because one feature will find its correspondence at the same interval up to i octaves shift. So any fixed-size descriptor capturing enough distinctive local information works in such situation. But if the scale change between two image is $2^{i+\delta}$ $(i \in \mathcal{N}, 0 < \delta < 1)$, then one feature will not find its correspondence at the same interval. In such case, the fixed-size descriptor will not cover the same image region and thus introduces some false matchings. By considering this default, we propose that the descriptor region has the size proportional to the Gaussian blur where the feature is detected. This increases the overlap of covered region between correspondences when scale change is $2^{i+\delta}$ (Fig. 5). In the other hand, the size of descriptor becomes bigger and bigger through octaves if the sub-sampling is canceled. So the sampling step becomes smaller and smaller with respect to blur. This is not consistent with scale invariance. To keep SIFT scale invariance, the new descriptor is sub-sampled again to make it as similar as possible to the original SIFT(see Fig. 6). Thus the new SIFT framework is still scale-invariant.

Third, with increased blur through octaves, the step between two adjacent intervals increases also by factor 2, 4, 16, $\cdots$. Then the scale space is sampled more and more sparsely through octaves. This makes it more difficult the 3D interpolation refinement. In addition, SIFT descriptor is constructed approximately on these sparse intervals without really interpolating a new interval.

Figure 5: The right image is $2^{2/3}$-subsampling of the left image. A feature in the right image with blur σ corresponds to a feature with $\sigma \times 2^{2/3}$ in the left image. The fixed-size descriptor gives evidently different patch for a common feature in two images (yellow patch in the left image via green patch in the right image). The two patches (in green) look almost the same by using descriptor with size proportional to blur.

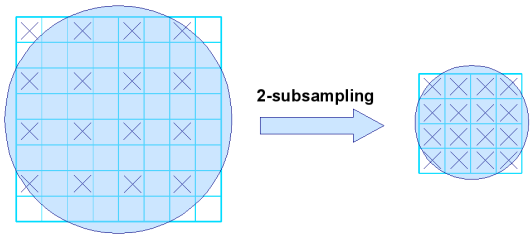

Figure 6: SIFT descriptor is constructed by summarizing the gradient information of a region around the detected feature. This region has fixed size in Lowe's SIFT. The gradient information is weighted by Gaussian weighting function indicated by the overlaid circle. In improved SIFT, the region has size proportional to blur and needs to be sub-sampled to maintain the scale invariance since images are up-sampled. A 2-subsampling is shown on the right.

This introduces also error. To compensate this effect, the number of intervals is increased with the same factor through octaves (see Fig. 4). This means that the up-sampling is performed also in the scale direction, just as that in x and y directions in image. Remark that this up-sampling in the scale still keep the scale invariance of Laplacian value.

For sub-pixel refinement, the sub-sampling removal has three-fold effect. On the one hand, the assumption that image is locally 2-order is more valid with the increase of Gaussian blur. This makes the 3D interpolation more precise. On the other hand, the increase of Gaussian blur also makes it more difficult to localize feature. The third point has already been mentioned: the back-projection is already removed. Are the three factors are completely compensated by each other? Theoretically the answer is yes. Given a local extrema $\mathbf{x} = (x_0, y_0, \sigma_0)$, its offset is given by Eq. (7): $\mathbf{\Delta x} = \left(\frac{\partial^2 D}{\partial \mathbf{x}^2}\right)^{-1} \frac{\partial D}{\partial \mathbf{x}}$. If the factor to back project to the original resolution is 2, the final offset is $2\mathbf{\Delta x}$. If the error is directly computed on 2-upsampled image, by Eq. (2), (3) (7) and (8), it gives: $\left(\frac{4}{m^2}\frac{\partial^2 D}{\partial \mathbf{x}^2}\right)^{-1} \frac{4}{mn}\frac{\partial D}{\partial \mathbf{x}} = \frac{m}{n}\left(\frac{\partial^2 D}{\partial \mathbf{x}^2}\right)^{-1} \frac{\partial D}{\partial \mathbf{x}} = \frac{m\mathbf{\Delta x}}{n}$. According to Fig. 3, by adding Gaussian blur bigger than 1.6, $\frac{m}{n}$ is about 2 and the variation is ignorable. Then the offset is about $2\mathbf{\Delta x}$, which means no improvement in precision is obtained by canceling the sub-sampling between octaves. But in practice, we do gain something due to the fact that the used image is digital. First, the local extrema at integer pixel position is more precise in up-sampled image, which means 3D refinement has a good departure point. Second, the gradient and the Hessian computation is more precise on up-sampled image. Even if the uncertainty of feature position is increased, a Ransac algorithm can be used to post-process SIFT matchings to only keep the most precise ones. Remark that if no blur is added to the image, the average of $\frac{m}{n}$ is also about 2, but the variance is rather high. Thus the computed offset of features is not reliable.

3 Absolute localization error evaluation

Absolute localization error of Lowe's SIFT method is evaluated by a pair of images in this section. We first review different precision evaluation procedures and point out their drawbacks. We test the localization error more directly under different geometric transformation.

3.1 Evaluation method

The most popular evaluation criteria is *repeatability* introduced in [19, 7], which is defined as the ratio between the number of correspondences and the minimum number of points detected in two images. Feature x_a and x_b correspond if:

- the error in relative point location is less than ϵ_p pixels: $x_a - H \cdot x_b < \epsilon_p$, where ϵ_p is typically 1.5 pixels and H is the ground truth homography between two images;
- $1 - \frac{R_{\mu_a} \cap R_{H^T \mu_b H}}{R_{\mu_a} \cup R_{H^T \mu_b H}} < \epsilon_o$ where R_μ the detected region determined by the shape matrix μ given by affine invariant interest point detector [19]; $H^T \mu_b H$

is shape matrix projected to the other image; $R_{\mu_a} \cap R_{H^T \mu_b H}$ is the intersection of regions and $R_{\mu_a} \cup R_{H^T \mu_b H}$ is their union.

We argue that "repeatability" is not well adapted for localization error due to the following reasons:

1. It is not sure that x_a and x_b is a good correspondence if the two above criteria are satisfied. In fact the above two measures depend on the detected scale. So it is not easy to find a universal threshold.

2. It is assumed that the scene is planar and the ground truth homography between two images is estimated from some control points. But in practice, we cannot have a completely planar scene. Even if it is, the camera lens introduces some non-linear distortion. Thus a pair of real images are related by a homography up to some error. This means the ground truth itself is problematic.

3. The features detected in different scales do not have the same precision. So it is better to evaluate in different octaves separately.

4. More precise matchings can be obtained by using a Ransac-based algorithm.

There exists also other methods to evaluate the precision of interest points. The most direct methods are based on visual inspections [2, 23], which depend directly on human perception. Some methods suppose that the ground-truth is known, which is not always true for many applications. Ground-truth is in general created by human and relies on his symbolic interpretation of the image and is therefore subjective. In [6], the edge detector performance is evaluated based on receiver operating characteristic (ROC) curves. Edge detector output is matched against ground truth to count true positive and false positive edge pixels.

Instead of evaluating directly location precision of interest points, some other criteria do not require the exact position of interest points. In [8, 13], authors used projective invariants to evaluate interest points precision. But this kind of methods needs scene composed of simple geometric objects, like polygons or polyhedrals since the reference values are computed from scene measurements. In [30], authors used four different criteria: alignment of the extracted points, accuracy of the 3D reconstruction, accuracy of the epipolar geometry and stability of the cross-ratio. In [4], another four global criteria are proposed: collinearity, intersection at a single point, parallelism and localization on an ellipse.

None of the above methods consider lens distortion or other optic system aberrations in their precision evaluation procedure. For the methods requiring scene composed of simple geometric objects, they are designed for some specific model-based interest points detectors and not very adequate for SIFT points. Moreover, since the reference measures are from the scene, the objects should be constructed with high precision, which is very difficult. The repeatability evaluation looks a good criteria for SIFT points. But the problem is that this criteria is designed to be generic for different interest points detectors. Lens distortion or "outliers" can easily affect the evaluation performance.

Due to the above problems, we evaluate the absolute localization error in a more direct way. Synthetic images are used in the test to avoid any masturbation like lens distortion. A transformation is applied on a reference image to obtain the second image. SIFT is applied on two images to extract feature points. The associated SIFT descriptors are matched by nearest neighbor distance ratio. A Ransac-like parameter-free algorithm [25] is applied to eliminate false matchings. A global homography is computed from the Ransac-verified matchings by using a least-square method. The precision is evaluated as the residual to this homography.

3.2 Test

The test is performed on five kinds of transformation: translation, rotation, zoom, tilt and affine transformation. The pre-zoom in SIFT method make the blur of the initial image about 1.6. Features are detected and matched by SIFT method. The number of octaves is fixed to be four. The proposed evaluation procedure is used and the error is computed respectively on different octaves. Remark that the number of features decreases fast through octaves. There can be not enough matchings on the third or fourth octave that the evaluation is not very reliable, in particular in the case of tilt and affine transformation, which is a hard task for SIFT.

Translation

Translation is the simplest case to test. For integer pixel translation, the evaluation always gives zero error. This is because two images are exactly the same up to an integer pixel translation. Error is observed when the translation is non-integer pixel. The reason can be two-fold: first, if the reference image itself is a little aliased, the second image is not exact and some artifact can be introduced; second, the feature position refinement is performed by a 3D interpolation in SIFT, which is based on the assumption that image can be locally approximated by 2-order Taylor expansion. Thus the performance of 3D refinement depends on the image regularity around local extrema. In addition, the implementation of sub-pixel translation can also pose a problem: FFT interpolation is exact but will introduces "ringing" artifact at the contours and image borders, while spline interpolation is just an approximation to FFT interpolation. Here a 7-order spline interpolation is used (see images in Fig. 7). Even this test is very simple, it gives us the idea about the best precision that SIFT method can achieve. The average and standard deviation of the residual error is recapitulated in Table 3.2. It shows that the error increases through octave. Remark that the integer translation $(45, 32)$ gives zero error on the first octave, but on the other octaves, error is observed because images are sub-sampled and translation becomes non-integer. In fact, the error is smaller when the translation is close to integer or semi-integer.

Rotation

Rotation is another basic geometric transform in image processing. An exact implementation by FFT is feasible by decomposing rotation into three shear transforms. But it suffers also from "ringing" artifact as the translation case.

	octave −1	octave 0	octave 1	octave 2
$(45, 32)$	6.8e−5/5.4e−4	5.4e−5/1.4e−4	0.1202/0.09516	0.1475/0.09603
$(45.1, 32.1)$	0.02727/0.02277	0.02513/0.02253	0.1195/0.09369	0.1529/0.09976
$(45.3, 32.3)$	0.04100/0.03296	0.06049/0.04516	0.1219/0.09963	0.2403/0.2079
$(45.5, 32.5)$	0.01188/0.01516	0.07425/0.05799	0.1196/0.09577	0.2596/0.2162
$(45.7, 32.7)$	0.04466/0.03993	0.06434/0.05063	0.1230/0.09569	0.2851/0.2176
$(45.9, 32.9)$	0.02578/0.02106	0.02729/0.02631	0.1393/0.1131	0.2729/0.2114

Table 1: The average/standard deviation of residual error in translation case for Lowe's SIFT. The translation in x and y direction is $(45, 32)$, $(45.1, 32.1)$, $(45.3, 32.3)$, $(45.5, 32.5)$, $(45.7, 32.7)$ and $(45.9, 32.9)$.

Thus a 7-order spline interpolation is used like in translation test (see images in Fig. 7). As the evaluation shown in Table 3.2, more close to 0° or 90° is the rotation, more small is the error. This is due to the error in orientation estimation of feature in SIFT method. This leads to incorrect descriptor and thus to false matchings. The same phenomenon as the translation case is that the error increases through octaves.

	octave −1	octave 0	octave 1	octave 2
15°	0.03917/0.03254	0.06778/0.04967	0.1353/0.09935	0.3101/0.2485
25°	0.04381/0.03431	0.07788/0.05160	0.1544/0.09405	0.3289/0.2262
35°	0.04924/0.03434	0.08954/0.06176	0.1725/0.1056	0.3866/0.2547
45°	0.05201/0.03729	0.08964/0.06092	0.1772/0.1241	0.4791/0.4057
55°	0.05311/0.03959	0.08459/0.05609	0.1864/0.1362	0.4375/0.3382
65°	0.04785/0.03788	0.07989/0.05578	0.1393/0.08710	0.3472/0.2120
75°	0.04101/0.03350	0.06352/0.04420	0.1352/0.09963	0.2717/0.1656
85°	0.03262/0.02661	0.05160/0.03899	0.1206/0.1100	0.2076/0.1510

Table 2: The average/standard deviation of residual error in rotation case for Lowe's SIFT. The rotation angle varies from 15° to 85° with the step of 10°.

Zoom

In case of zoom, the right image is generated by blurring the left image with $\sqrt{t^2 - 1} \times 0.8$ followed by t-subsampling with t the scale change between two images. Another approach to obtain the right image is to zoom in the left image by t. The first approach is used here because the second approach will increase an image blur to $t \times 0.8$. This is not consistent with the assumption that a well-sampled image has Gaussian blur 0.8.

It is more difficult to deal with zoom than rotation and translation. Even if theoretically SIFT is scale invariant, in practice the scale invariance is disturbed a lot by scale quantization and blur issue. SIFT scale invariance is in fact blur invariance. That is, for two matched features, their associated descriptors are identical if they are compared at the same resolution. To be more clear, the most important lemma in [27] is cited here:

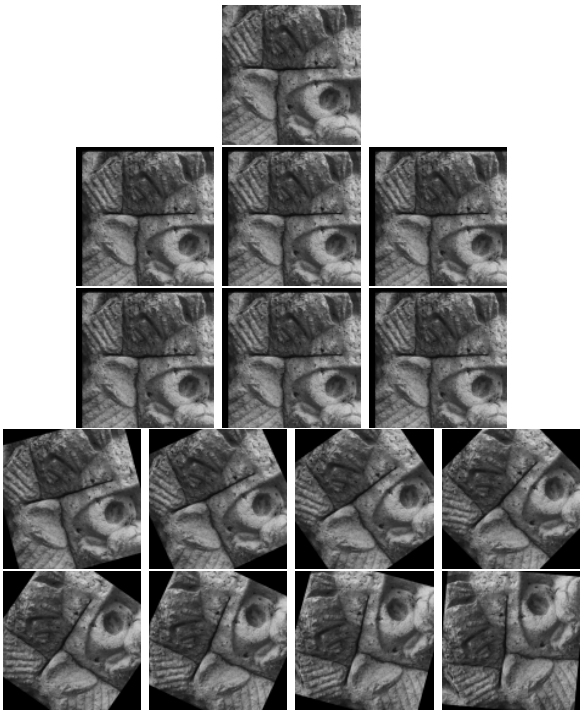

Figure 7: First row: original image. Second and third row: image translated by $(45, 32)$, $(45.1, 32.1)$, $(45.3, 32.3)$, $(45.5, 32.5)$, $(45.7, 32.7)$, $(45.9, 32.9)$. Fourth and fifth row: image rotated by 15°, 25°, 35°, 45°, 55°, 65°, 75°, 85°.

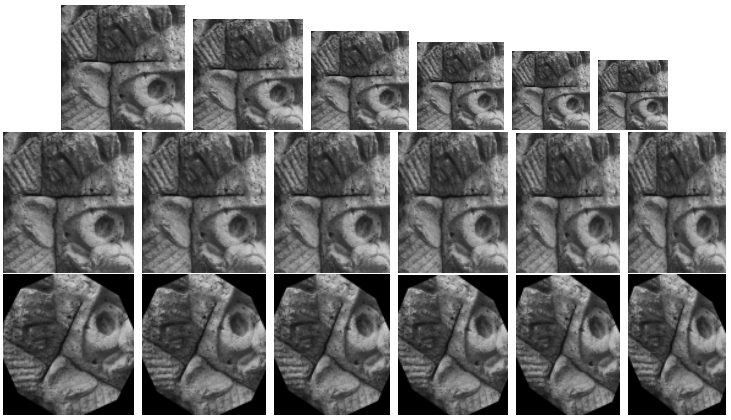

Figure 8: First row: image sub-sampled by factor $2^{1/6}$, $2^{2/6}$, $2^{3/6}$, $2^{4/6}$, $2^{5/6}$, 2. Second row: image transformed by tilt with t equal to $2^{1/12}$, $2^{2/12}$, $2^{3/12}$, $2^{4/12}$, $2^{5/12}$, $2^{6/12}$. Third row: image transformed by affine transformation with $\phi = 37°$, $\psi = 24°$ and t equal to $2^{1/12}$, $2^{2/12}$, $2^{3/12}$, $2^{4/12}$, $2^{5/12}$, $2^{6/12}$.

Lemma 1 *Let u and v be two digital images that are frontal snapshots of the same continuous flat image $\mathbf{u}_0$, $u := \mathbf{S_1}\mathbf{G}_\beta\mathbf{H}_\lambda\mathbf{u}_0$ and $v =: \mathbf{S_1}\mathbf{G}_\delta\mathbf{H}_\mu\mathbf{u}_0$, taken at different distances, with different Gaussian blurs and possibly different sampling rates. Let $\mathbf{w}(\sigma, \mathbf{x}) = (\mathbf{G}_\sigma\mathbf{u}_0)(\mathbf{x})$ denote the scale space of $\mathbf{u}_0$. Then the scale spaces of u and v are*

$$\mathbf{u}(\sigma, \mathbf{x}) = \mathbf{w}(\lambda\sqrt{\sigma^2 + \beta^2}, \lambda\mathbf{x}) \text{ and } \mathbf{v}(\sigma, \mathbf{x}) = \mathbf{w}(\mu\sqrt{\sigma^2 + \delta^2}, \mu\mathbf{x})$$

If $(s_0, \mathbf{x_0})$ is a key point of $\mathbf{w}$ satisfying $s_0 \geq max(\lambda\beta, \mu\delta)$, then it corresponds to a key point of $\mathbf{u}$ at the scale σ_1 such that $s_0 = \lambda\sqrt{\sigma_1^2 + \beta^2}$, whose SIFT descriptor is sampled with mesh $\sqrt{\sigma_1^2 + \beta^2}$. In the same way (s_0, x_0) corresponds to a key point of $\mathbf{v}$ at scale σ_2 such that $s_0 = \mu\sqrt{\sigma_2^2 + \delta^2}$, whose SIFT descriptor is sampled with mesh $\sqrt{\sigma_2^2 + \delta^2}$.

$\mathbf{S_1}$ is the 1-sampling operation applied on continuous image to obtain a digital image; $\mathbf{H}_\lambda$ is sub-sampling of factor λ; $\mathbf{G}_\beta$ is the Gaussian convolution with standard deviation β. $\mathbf{G}_\beta$ and $\mathbf{G}_\delta$ are camera blur applied on the infinite resolution image (blur free) before 1-sampling to avoid aliasing. The above lemma is proved in the continuous setting under the assumption that Gaussian blur performed by SIFT method approximates well the camera blur and gives aliasing-free images.

This lemma is easier to understand by an example with $\mu/\lambda = 2$. Then $s_0 = \sqrt{\sigma_1^2 + \beta^2} = 2\sqrt{\sigma_2^2 + \delta^2}$. Assume a pair of correspondence: $f_\mathbf{u}$ in $\mathbf{u}$ and $f_\mathbf{v}$ in $\mathbf{v}$. $f_\mathbf{u}$ lies on the scale two times coarser than $f_\mathbf{v}$, and $f_\mathbf{u}$'s descriptor has sampling step two times bigger than $f_\mathbf{v}$'s descriptor. SIFT's dyadic scale space structure is adaptive to this case: $\mathbf{u}$ and $\mathbf{v}$ ideally superposes by one octave shift (if the pre-zoom in SIFT is not considered here). This is true for all cases with $\mu/\lambda = 2^i, i \in \mathcal{N}$. So the localization error is zero up to machine precision. But the case becomes more complicated when $\mu/\lambda \neq 2^i$. Assume $\mu/\lambda = 2^{i+\epsilon}, i \in \mathcal{N}, 0 < \epsilon < 1$, then $\mathbf{u}$ and $\mathbf{v}$ never superpose in scale space due to the contradiction between SIFT scale space dyadic structure and $\mu/\lambda = 2^{i+\epsilon}$.

It is special when $\epsilon = s/Ninter, s = 0, 1, \cdots, Ninter - 1$ ($Ninter$ is the number of intervals in one octave). Now assume $Ninter = 3$ and $s = 1$, then $\mu/\lambda = 2^{1/3}$, then $f_\mathbf{u}$ lies on the scale $2^{1/3}$ times blurred than $f_\mathbf{v}$. This coincides with the fact that one SIFT octave is divided into $Ninter = 3$ intervals (Fig. 1). For $f_\mathbf{u}$ on interval s of octave o, $f_\mathbf{v}$ on interval $s+1$ of octave o containing blur $2^{1/3}$ bigger. But the problem rises when 3D refinement (in space and scale) is performed on local extrema. 3D refinement is sensible to blur so the refinement in space localization and scale of $f_\mathbf{u}$ and $f_\mathbf{v}$ will be different. And the region used for descriptor has the size proportional to the refined scale, which can also introduce error in descriptor and possibly leads to false matchings: if there exists a feature $f'_\mathbf{u}$ very close to $f_\mathbf{u}$, then little error in descriptor can make $f_\mathbf{v}$ matched by $f'_\mathbf{u}$, instead of $f_\mathbf{u}$. But the introduced error is particularly small, Ransac-like algorithm does not guarantee to remove this kind of false matchings.

The most general case occurs when μ/λ is any value. In such case, no features extracted in scale space is good candidate for matching because the blur can never be equal: $\lambda\sqrt{\sigma_1^2 + \beta^2} \neq \mu\sqrt{\sigma_2^2 + \delta^2}$. Depending on the performance of 3D refinement, more accurate blur information can be estimated. But even with the correct blur ($\lambda\sqrt{\sigma_1^2 + \beta^2} = \mu\sqrt{\sigma_2^2 + \delta^2}$), the descriptor is always constructed from the region extracted in the closest interval (instead of

	octave −1	octave 0	octave 1	octave 2
$2^{1/6}$	0.2333/0.1193	0.2399/0.1414	0.3136/0.2050	0.4557/0.3383
$2^{2/6}$	0.2259/0.1249	0.2240/0.1307	0.2054/0.1479	0.2834/0.1700
$2^{3/6}$	0.2274/0.1265	0.1753/0.1103	0.1690/0.1175	0.3664/0.2801
$2^{4/6}$	0.2639/0.1852	0.1474/0.08682	0.1425/0.09033	0.1737/0.1230
$2^{5/6}$	0.1705/0.1015	0.1571/0.08692	0.1731/0.1114	0.1993/0.1158
$2^{6/6}$	0.1113/0.07107	0.03163/0.02510	0.03320/0.03017	0.02933/0.02641

Table 3: The average/standard deviation of residual error in zoom case for Lowe's SIFT. The zoom factor is $2^{1/6}$, $2^{2/6}$, $2^{3/6}$, $2^{4/6}$, $2^{5/6}$ and $2^{6/6}$.

interpolating a region). In addition, the blur is only used to determine the size of region for descriptor. But the region is not sampled on the mesh mentioned in Lemma 1. This means the descriptor is just approximative and can lead to false matchings. Due to all of the above reasons, the error in the case of zoom is bigger than the case of translation and rotation (Table 3.2).

The above problem implies that in SIFT method the initial blur in image affects SIFT matching performance. The Gaussian blur performed by SIFT in scale space is based on the initial blur estimation. With incorrect estimation, the resulted images in scale space are too or not enough blurred. This leads to inaccurate feature position and scale refinement, thus to inaccurate descriptor.

The evaluation is shown in Table 3.2. In the first octave, the left image has no matchings with the right image when the scale change is 2. But in the other octaves, the precision is higher when the scale change between two images is close to 2. This can be explained by two limit scale changes: 2 and $2^{1/6}$. For example, if the scale change is 2, octave 2 of the left image's scale space almost superposes octave 1 of the right image's scale space. Thus only the features in the left image are reprojected to original image resolution; while for scale change of $2^{1/6}$, almost all the matchings features are found on the same octave and both are reprojected and thus introduce more error.

Affine transformation

Affine transformation is still more difficult. Affine transformation A can be decomposed by SVD (Singular Value Decomposition):

$$A = H_\lambda R_1(\psi) T_t R_2(\phi) = \lambda \begin{pmatrix} \cos\psi & -\sin\psi \\ \sin\psi & \cos\psi \end{pmatrix} \begin{pmatrix} t & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \cos\phi & -\sin\phi \\ \sin\phi & \cos\phi \end{pmatrix} \tag{9}$$

with $R_1(\psi)$ and $R_2(\phi)$ rotatin matrix, T_t a tilt and H_λ an expansion of λ. Rotation and zoom have been already analyzed before, the only new element here is the tilt. Without loss of generality, assume $t < 1$, the tilt T_t sub-samples the image by factor t in x-direction without changing the resolution in y-direction. The tilt is not consistent with Gaussian blur performed in SIFT scale space because Gaussian blur is isotropic but tilt is not. More precisely, to attain blur (scale) invariance, on one hand, in y-direction, $f_{\mathbf{u}}$ should be on scale $\frac{1}{t}$ times coarser than $f_{\mathbf{v}}$; at the other hand, in x-direction, $f_{\mathbf{u}}$ should be on the same scale as $f_{\mathbf{v}}$. This inconsistency is an extra error source besides the previously mentioned errors. Affine transformation or tilt is a challenge for SIFT

	octave −1	octave 0	octave 1	octave 2
$2^{1/12}$	0.1795/0.1149	0.1672/0.1025	0.2324/0.1485	0.3481/0.1906
$2^{2/12}$	0.1626/0.1108	0.1991/0.1340	0.3305/0.1997	0.6024/0.4363
$2^{3/12}$	0.1721/0.1037	0.2512/0.1642	0.4125/0.2843	0.7197/0.4508
$2^{4/12}$	0.1877/0.1231	0.2743/0.1783	0.4941/0.3351	0.8789/0.5487
$2^{5/12}$	0.2370/0.1405	0.2915/0.1882	0.5619/0.3951	1.0779/0.5896
$2^{6/12}$	0.2391/0.1305	0.3358/0.2118	0.6287/0.3868	1.0522/0.5904

Table 4: The average/standard deviation of residual error in tilt case for Lowe's SIFT. $\phi = 0°$, $\psi = 0°$ and t is $2^{1/12}$, $2^{2/12}$, $2^{3/12}$, $2^{4/12}$, $2^{5/12}$ and $2^{6/12}$.

	octave −1	octave 0	octave 1	octave 2
$2^{1/12}$	0.2024/0.1192	0.1989/0.1161	0.3231/0.2552	0.6415/0.4839
$2^{2/12}$	0.1722/0.1069	0.2055/0.1335	0.3915/0.2671	0.7809/0.5571
$2^{3/12}$	0.1708/0.1101	0.2573/0.1724	0.4605/0.3552	0.7238/0.4215
$2^{4/12}$	0.1926/0.1302	0.2538/0.1439	0.5094/0.3415	0.8436/0.5008
$2^{5/12}$	0.2475/0.1559	0.2999/0.1834	0.6134/0.4374	0.7514/0.4538
$2^{6/12}$	0.2517/0.1414	0.3341/0.1921	0.5596/0.3058	0.8972/0.7274

Table 5: The average and standard deviation of error in affine transformation case for Lowe's SIFT. $\phi = 37°$, $\psi = 24°$ and t is $2^{1/12}$, $2^{2/12}$, $2^{3/12}$, $2^{4/12}$, $2^{5/12}$ and $2^{6/12}$.

because it is experimentally shown that SIFT works only with *transition tilt* smaller than 2.5 [26]. Thus an affine transformation or a tilt is more error-prone than zoom and rotation. In Table 3.2, a pure tilt transformation is evaluated with t varying from $2^{1/12}$ to $2^{6/12}$. The error increase with t because it is difficult for SIFT to match correctly the features with big t. In Table 3.2, an affine transformation is tested with $\phi = 37°$, $\psi = 24°$ and t varying from $2^{1/12}$ to $2^{6/12}$. Remark that in case of tilt and affine transformation, fewer matchings are found in coarse octaves. So the error evaluation can be less reliable.

Homography

Homography can describe any transformation between two images of a plane scene viewed by an ideal pinhole camera. Any homography $H : (x, y) \rightarrow (X, Y) = (F_1(x, y), F_2(x, y))$ can be locally approximated by an affine transformation around each point $(x_0, y_0) \rightarrow (X_0, Y_0)$ with 1-order Taylor expansion:

$$\begin{pmatrix} X - X_0 \\ Y - Y_0 \end{pmatrix} = \begin{pmatrix} \frac{\partial F_1}{\partial x}(x_0, y_0) & \frac{\partial F_1}{\partial y}(x_0, y_0) \\ \frac{\partial F_2}{\partial x}(x_0, y_0) & \frac{\partial F_2}{\partial y}(x_0, y_0) \end{pmatrix} \begin{pmatrix} x - x_0 \\ y - y_0 \end{pmatrix} + O \begin{pmatrix} (x - x_0)^2 + (y - y_0)^2 \\ (x - x_0)^2 + (y - y_0)^2 \end{pmatrix}$$

At point (i, j), the local zoom factor in two orthogonal directions $t_1(i, j)$, $t_2(i, j)$ can be computed by decomposing the corresponding affine transformation $A(i, j)$ like Eq.(9). If $t_1(i, j)$ and $t_2(i, j)$ are both bigger than 1 for all i, j, image is compressed nowhere and H can be directly applied on image. If $t_1(i, j)$ (or $t_2(i, j)$) is smaller than 1, then there is a sub-sampling around i, j along the corresponding direction. In such case, a pre-zoom of factor $\frac{1}{t_1}$ (or $\frac{1}{t_2}$) is needed around (i, j)

along the corresponding direction before applying H to avoid aliasing. But this point-wise pre-zoom is not feasible since $t_1(i,j)$ (or $t_2(i,j)$) differs from point to point. It is neither feasible to pre-zoom image by factor $\frac{1}{\min_{(i,j)}(t_1(i,j))}$ and $\frac{1}{\min_{(i,j)}(t_2(i,j))}$ separately in two directions since the direction $\phi(i,j)$ differs from point to point and a rotation commutes only with an isotropic zoom. So the only solution is first to compute $t = \min_{(i,j)}(t_1(i,j), t_2(i,j))$. If $t < 1$, a global pre-zoom $\frac{1}{t}$ is applied on image. With this adapted anti-aliasing pre-zoom, H can be safely applied on an image. Afterwards, to cancel the pre-zoom, we should again do a zoom-out with the same factor t. As always, a Gaussian blur $0.8 \times \sqrt{\frac{1}{t^2} - 1}$ is required before sub-sampling. The algorithm is recapitulated in Algorithm 1. It can be proven that $t = \min_{(i,j)}(t_1(i,j), t_2(i,j))$ can be computed just on four corners of image (**PROOF NEEDED**). The resulted image will be a little blur since the Gaussian blur applied is adapted to the biggest local zoom-out.

Algorithm 1 (Anti-aliasing homography)

Input*: image* $\mathbf{I}$*, homography* $\mathbf{H}$
Output*: image* $g(\mathbf{I}, \mathbf{H}) = \mathbf{I} \circ \mathbf{H}^{-1}$
At each corner of $\mathbf{I}$ *compute the Jacobian* $\mathbf{J}$ *of* $\mathbf{H}$ *and the SVD of* $\mathbf{J}$*. Take* t *the smallest among these 8 singular values.*
Let $\mathbf{S} = \begin{pmatrix} s & 0 & 0 \\ 0 & s & 0 \\ 0 & 0 & 1 \end{pmatrix}$ *with* $s = max(\frac{1}{t}, 1)$*.*
if $t(\mathbf{H}) < 1$ **then**
  $\mathbf{I} = g(\mathbf{I}, \mathbf{SH})$*;*
  Convolve $\mathbf{I}$ *with Gaussian kernel of standard deviation* $0.8 \times \sqrt{\frac{1}{t^2} - 1}$*;*
  Replace $\mathbf{H}$ *by the zoom-out matrix* $\mathbf{S}^{-1} = \begin{pmatrix} t & 0 & 0 \\ 0 & t & 0 \\ 0 & 0 & 1 \end{pmatrix}$
end
Return image $\mathbf{I} \circ \mathbf{H}^{-1}$*, computed by Fourier interpolation or other high-order interpolation.*

Conclusion

Different tests have been performed to evaluate SIFT localization error. But as we have seen, this evaluation depends not only on localization error of SIFT detector, but also a lot on SIFT descriptor and matching performance. SIFT detector precision is mainly determined by local Laplacian extrema extraction and 3D interpolation. SIFT descriptor is constructed by summarizing the gradient information of a region with size proportional to blur. The blur value is correct only when the initial image blur is well estimated and the interpolation in scale direction is well performed. For computational efficiency, the interpolated blur is only used to decide the size of region for descriptor. No new interval is interpolated between two intervals. The region for descriptor is extracted from the closest existing interval. This can introduce error in descriptor and lead to false matchings.

The tests can be divided into two groups. One group with scale change: zoom, tilt and affine transformation. The other without scale change: translation and rotation. Both groups suffer from the above error factors. But the group without scale change suffers less from the blur issue than the group with scale change because scale change leads to different performance in 3D interpolation for a feature and its correspondence. For all transformations, the error increases with octaves. This is normal because all the features are finally projected back to the original image. For translation and rotation, inaccurate descriptor does not affect a lot the evaluation result because this descriptor inaccuracy caused by blur issue is quite similar for the left and right image when there is no scale change between them. So the matching can still be reliable. Rotation has an error bigger than translation because the rotation case suffers also from the imprecise estimation of the principal orientation of features. This can be seen from precision recapitulation for rotation case in Table 3.2: better precision is obtained when the rotation angle is close to $0°$ or $90°$. We think that in translation and rotation case, the error reflects the SIFT detection accuracy itself without disturbed a lot by matching error. For tilt and affine transformation, SIFT even does not find enough matchings in octave 3 and 4 because SIFT is only partially invariant against affine transformation and the number of features decrease through octaves. In such case, Ransac-like algorithm does not work well and some curves are stepped, which are not very reliable. Remark that this kind of evaluation depends not only on detector's localization capacity, but also on the blur issue, descriptor and matching error.

3.3 Improvement

In this section, the improvement in section 2.3 is evaluated. The pre-zoom in SIFT method makes the initial blur of image about 1.6. The evaluation is still disturbed by previously mentioned error factors if there is scale change between two images: blur estimate error, inaccurate descriptor and false matchings. This kind of error is amplified because the removal of sub-sampling makes images more blurred and can lead to more error in blur. In the other hand, the more blurred images can also make 3D refinement more precise because the assumption that image can be locally approximated by 2-order Taylor expansion is more valid. For translation and rotation, they benefit from the precision improvement of 3D interpolation. So the precision is kept and even improved through octaves (see Table 3.3 and 3.3). But for the zoom, tilt and affine transformation which contain a scale change, the error incurred by blur issue plays a more important role and prevails over the gain in precision of 3D interpolation. So the improvement in precision in not evident compared with Lowe's SIFT. (see Table 3.3, 3.3, and 3.3).

4 Localization uncertainty

Localization uncertainty is more adapted to evaluate the performance of detector since it only depends on detection process. One image is enough to evaluate localization uncertainty. Thus the error factors like blur inconsistency and inaccurate descriptor do not have influence because they only introduce errors when features are matched. As indicated in [17, 3], localization uncertainty is

	octave −1	octave 0	octave 1	octave 2
$(45, 32)$	6.8e−5/5.4e−4	2.3e−4/1.4e−3	9.1e−4/3.0e−3	3.9e−3/7.4e−3
$(45.1, 32.1)$	0.03043/0.02880	0.02005/0.01794	0.02073/0.02067	0.01879/0.01830
$(45.3, 32.3)$	0.04140/0.03352	0.02918/0.02393	0.02863/0.02510	0.03208/0.02951
$(45.5, 32.5)$	0.01164/0.01493	2.8e−4/1.1e−3	7.7e−4/2.4e−3	4.2e−3/7.6e−3
$(45.7, 32.7)$	0.04383/0.03824	0.02901/0.02381	0.03209/0.03125	0.02735/0.02250
$(45.9, 32.9)$	0.02609/0.02158	0.02154/0.02003	0.01856/0.01568	0.01686/0.01984

Table 6: The average/standard deviation of residual error in translation case for improved SIFT. The translation in x and y direction is $(45, 32)$, $(45.1, 32.1)$, $(45.3, 32.3)$, $(45.5, 32.5)$, $(45.7, 32.7)$ and $(45.9, 32.9)$.

	octave −1	octave 0	octave 1	octave 2
15°	0.03885/0.03174	0.02426/0.01923	0.02023/0.01699	0.01671/0.01309
25°	0.04279/0.03197	0.02938/0.02256	0.02689/0.02276	0.01659/0.01470
35°	0.05023/0.03590	0.03131/0.02303	0.02446/0.01936	0.02787/0.02257
45°	0.05084/0.03535	0.03176/0.02172	0.02642/0.02090	0.02807/0.02685
55°	0.05337/0.03985	0.03149/0.02399	0.02445/0.01894	0.02257/0.02201
65°	0.04915/0.03965	0.02924/0.02250	0.02510/0.02360	0.01987/0.01771
75°	0.04017/0.03175	0.02472/0.01969	0.01911/0.01517	0.01496/0.01095
85°	0.03277/0.02665	0.02297/0.01903	0.01908/0.01603	0.02305/0.01893

Table 7: The average/standard deviation of residual error in rotation case for improved SIFT. The rotation angle varies from 15° to 85° with the step of 10°.

	octave −1	octave 0	octave 1	octave 2
$2^{1/6}$	0.2336/0.1205	0.2403/0.1403	0.2877/0.1632	0.3947/0.2697
$2^{2/6}$	0.2303/0.1304	0.2234/0.1231	0.1800/0.1340	0.2012/0.1168
$2^{3/6}$	0.2324/0.1340	0.1644/0.09814	0.1281/0.0841	0.2094/0.1471
$2^{4/6}$	0.2609/0.1847	0.1516/0.09282	0.1222/0.07958	0.1200/0.09484
$2^{5/6}$	0.1697/0.1017	0.1571/0.08651	0.1497/0.09481	0.1953/0.1264
$2^{6/6}$	0.1242/0.07303	0.03078/0.02087	0.01993/0.01506	0.02534/0.02320

Table 8: The average/standard deviation of residual error in zoom case for improved SIFT. The zoom factor is $2^{1/6}$, $2^{2/6}$, $2^{3/6}$, $2^{4/6}$, $2^{5/6}$ and $2^{6/6}$.

	octave −1	octave 0	octave 1	octave 2
$2^{1/12}$	0.1800/0.1153	0.1795/0.1122	0.2373/0.1737	0.3202/0.2012
$2^{2/12}$	0.1601/0.1077	0.2086/0.1427	0.3604/0.2620	0.4791/0.3165
$2^{3/12}$	0.1657/0.09487	0.2528/0.1621	0.3953/0.3079	0.7354/0.4808
$2^{4/12}$	0.1873/0.1214	0.2713/0.1681	0.4825/0.3372	0.7222/0.3998
$2^{5/12}$	0.2357/0.1389	0.2946/0.1813	0.5678/0.4029	0.9433/0.5110
$2^{6/12}$	0.2400/0.1301	0.3223/0.2003	0.6022/0.4352	1.1887/0.6242

Table 9: The average/standard deviation of residual error in tilt case for improved SIFT. $\phi = 0°$, $\psi = 0°$ and t is $2^{1/12}$, $2^{2/12}$, $2^{3/12}$, $2^{4/12}$, $2^{5/12}$ and $2^{6/12}$.

	octave −1	octave 0	octave 1	octave 2
$2^{1/12}$	0.1999/0.1170	0.1880/0.1186	0.2283/0.1892	0.3613/0.2570
$2^{2/12}$	0.1749/0.1090	0.2007/0.1414	0.2979/0.1829	0.5280/0.3677
$2^{3/12}$	0.1687/0.1073	0.2384/0.1573	0.3800/0.2362	0.6316/0.4205
$2^{4/12}$	0.1954/0.1353	0.2697/0.1692	0.4350/0.2547	0.6980/0.3401
$2^{5/12}$	0.2431/0.1485	0.3007/0.1812	0.5334/0.3965	0.7264/0.3358
$2^{6/12}$	0.2557/0.1430	0.3314/0.2006	0.6456/0.4070	0.8707/0.4386

Table 10: The average/standard deviation of residual error in affine transformation case for improved SIFT. $\phi = 37°$, $\psi = 24°$ and t is $2^{1/12}$, $2^{2/12}$, $2^{3/12}$, $2^{4/12}$, $2^{5/12}$ and $2^{6/12}$.

usually measured as the inverse of Hessian matrix, which can be computed in different manners: residual-based approach [28, 41], derivative-based approach [9, 40] and direct approach [3].

The residual-based approach evaluates the residual of self-matching, which is similar to that used in Harris corner. Let $D(i, j)$ be the DoG (Difference of Gaussian) at a certain scale in SIFT scale space ($D(i, j)$ is replaced by gray-level $I(i, j)$ if the evaluation is on gray-level image). The residual of self-matching for DoG at point (i, j):

$$J(x, y) = \frac{1}{2} \sum_{(p,q)} w_{p,q} \Big(D(i + p + x, j + q + y) - D(i + p, j + q) \Big)^2 \tag{10}$$

where (p, q) is the displacement in a neighborhood around (i, j), and $w_{p,q}$ is an appropriate (Gaussian) weight. $J(x, y)$ is a continuous function in x and y if $I(i, j)$ is appropriately interpolated. $J(x, y)$ can be approximated by a quadratic function $g(x, y)$ over a neighborhood of $(0, 0)$ in the form

$$\begin{aligned} g(x, y) &= \frac{1}{2}(n_1 x^2 + 2n_2 xy + n_3 y^2) \\ &= \frac{1}{2} \begin{pmatrix} x & y \end{pmatrix} H \begin{pmatrix} x \\ y \end{pmatrix} \end{aligned} \tag{11}$$

with the Hessian matrix H in the form:

$$H = \begin{pmatrix} h_{11} & h_{12} \\ h_{12} & h_{22} \end{pmatrix} \tag{12}$$

h_{11}, h_{12} and h_{22} can be computed by a weighted least square problem:

$$\mathbf{h}^T = (h_{11}, h_{12}, h_{22}) = \underset{(h_{11},h_{12},h_{22})}{\operatorname{argmin}} \int\int_{\mathcal{X}} w(x,y) \left(J(x,y) - g(x,y)\right)^2 dxdy \quad (13)$$

with $w(x,y)$ an appropriate weight. A linear system is obtained by differentiating the above weighted sum with respect to h_{11}, h_{12} and h_{22} respectively and letting the result be 0:

$$\frac{1}{2}\mathbf{A}\mathbf{h} = \mathbf{b} \quad (14)$$

with $\mathbf{A}$, $\mathbf{h}$ and $\mathbf{b}$ defined as:

$$\begin{aligned}
\mathbf{A} &= \int\int_{\mathcal{X}} w(x,y)\mathbf{m}(x,y)\mathbf{m}(x,y)^T dxdy \\
\mathbf{h} &= \begin{pmatrix} h_{11} & h_{12} & h_{22} \end{pmatrix}^T \\
\mathbf{b} &= \int\int_{\mathcal{X}} w(x,y)J(x,y)\mathbf{m}(x,y)dxdy
\end{aligned}$$

and $\mathbf{m}(x,y) = (x^2, 2xy, y^2)^T$. H is then obtained by Eq. (12) and covariance matrix is defined as the inverse of Hessian matrix H:

$$\Sigma = H^{-1} \quad (15)$$

A default of residual-based approach is that H (and Σ) is not necessarily positive-definite. This contradicts with the definition of covariance matrix.

The derivative-based approach approximates Eq. (10) by developing $D(i + p + x, j + q + y) - D(i + p, j + q)$ via 1-order Taylor expansion:

$$\begin{aligned}
J(x,y) &= \frac{1}{2}\sum_{(p,q)} w_{p,q} \left(xD_x(i+p) + yD_y(j+q)\right)^2 \quad (16) \\
&= \frac{1}{2}\begin{pmatrix} x & y \end{pmatrix} H \begin{pmatrix} x \\ y \end{pmatrix}
\end{aligned}$$

with D_x, D_y 1-order partial derivative of D. This is a strong assumption that $D(i,j)$ can be locally approximated by 1-order Taylor expansion. The Hessian matrix has the explicit form:

$$H = \begin{pmatrix} \sum_{p,q} w_{p,q} D_x^2 & \sum_{p,q} w_{p,q} D_x D_y \\ \sum_{p,q} w_{p,q} D_x D_y & \sum_{p,q} w_{p,q} D_y^2 \end{pmatrix} \quad (17)$$

The covariance matrix is again defined as the inverse of the Hessian: $\Sigma = H^{-1}$.

A more direct approach for covariance matrix is to compute the Hessian by finite difference schema:

$$H = \begin{pmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{pmatrix} \quad (18)$$

with D_{xx}, D_{xy} and D_{yy} the second-order derivative of D. At a local extrema (i,j), approximate its DoG by 2-order Taylor expansion:

$$\begin{aligned}
D(i + \Delta x, j + \Delta y) &= D(i,j) + \begin{pmatrix} D_x(i,j) & D_y(i,j) \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix} \quad (19) \\
&\quad + \frac{1}{2}\begin{pmatrix} \Delta x & \Delta y \end{pmatrix} H \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix}
\end{aligned}$$

Since SIFT detector selects local extrema, the gradient $\begin{pmatrix} D_x(i,j) & D_y(i,j) \end{pmatrix}$ is close to zero, while the Hessian H captures the curvature at the feature (i,j). When point (i,j) is at a local minima, H is definite-positive; while H is definite-negative when (i,j) is at local maxima. Since the covariance matrix is always positive-definitive, the covariance matrix is finally in the form:

$$\Sigma = \pm H^{-1} \tag{20}$$

with positive sign when point (i,j) is at local minima and negative sign when (i,j) is at local maxima. To make the estimation of more robust, H is obtained by weighting Hessian matrix in a neighborhood around (i,j).

$$\Sigma = \left(\sum_{(i,j)\in\mathcal{N}_p} w(i,j) \begin{pmatrix} D_{xx}(i,j) & D_{xy}(i,j) \\ D_{xy}(i,j) & D_{yy}(i,j) \end{pmatrix} \right)^{-1} \tag{21}$$

The above three methods to compute the covariance matrix can give different result. We try to interpret them in the same framework and find which one is the most appropriate for SIFT features. Eq. (10) can be interpreted more properly by developing $D(i+p+x, j+q+y) - D(i+p, j+q)$ via 2-order Taylor expansion:

$$\begin{aligned} J(x,y) &= \frac{1}{2} \sum_{(p,q)} w_{p,q} \Big(D(i+p+x, j+q+y) - D(i+p, j+q) \Big)^2 \\ &= \frac{1}{2} \sum_{(p,q)} w_{p,q} \Big(\begin{pmatrix} x & y \end{pmatrix} \begin{pmatrix} D_x(i+p,j+q) \\ D_y(i+p,j+q) \end{pmatrix} + \\ &\quad \frac{1}{2} \begin{pmatrix} x & y \end{pmatrix} \begin{pmatrix} D_{xx}(i,j) & D_{xy}(i,j) \\ D_{xy}(i,j) & D_{yy}(i,j) \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \Big)^2 \\ &= \frac{1}{2} \sum_{(p,q)} \frac{1}{4} w_{p,q} \left(\begin{pmatrix} x & y \end{pmatrix} \begin{pmatrix} D_{xx}(i+p,j+q) & D_{xy}(i+p,j+q) \\ D_{xy}(i+p,j+q) & D_{yy}(i+p,j+q) \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \right)^2 \end{aligned} \tag{22}$$

$\begin{pmatrix} D_x(i+p,j+q) & D_y(i+p,j+q) \end{pmatrix}^T$ is set to zero under the assumption that the gradient of points in a small neighborhood of a local extrema is almost zero. This assumption is true when the image is enough blurred and a local extrema pixel looks similar to its neighboring pixels. The above expansion implies that Hessian matrix satisfies:

$$\begin{aligned} &\begin{pmatrix} x & y \end{pmatrix} H \begin{pmatrix} x \\ y \end{pmatrix} \\ &= \sum_{(p,q)} \frac{1}{4} w_{p,q} \left(\begin{pmatrix} x & y \end{pmatrix} \begin{pmatrix} D_{xx}(i+p,j+q) & D_{xy}(i+p,j+q) \\ D_{xy}(i+p,j+q) & D_{yy}(i+p,j+q) \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \right)^2 \end{aligned} \tag{23}$$

This is contradictory because the left side is 2-order while the right side is 4-order. Thus the residual-based approach is not appropriate to SIFT features. In fact, residual-based approach becomes exactly the same as the direct approach by removing the square in Eq. (10).

For the derivative-based approach, Eq. (16) is close to zero if the gradient $(i+p, j+q)$ is close to zero. This is valid under the same assumption as before.

In Lowe's SIFT, images are blurred and sub-sampled by 2 through octaves. All the images are well sampled and each image has the same nature in blur

as its counterparts on the same interval in the other octaves. These images are not enough blurred and the above assumption is thus not valid. So the above three approaches for Hessian matrix give the similarly reasonable result (see Fig. 9)). On the other hand, for the improved SIFT, images become more and more blurred through octaves due to the removal of sub-sampling. Then the assumption is also more valid in coarse octaves. The consequence is that residual-based approach and derivative-based approach give Hessian matrix with too small eigenvalues in coarse octaves (see Fig. 9). So these two approaches are not very adaptive for improved SIFT schema. Only the direct approach gives a reasonable estimation of Hessian matrix for both Lowe's SIFT and improved SIFT.

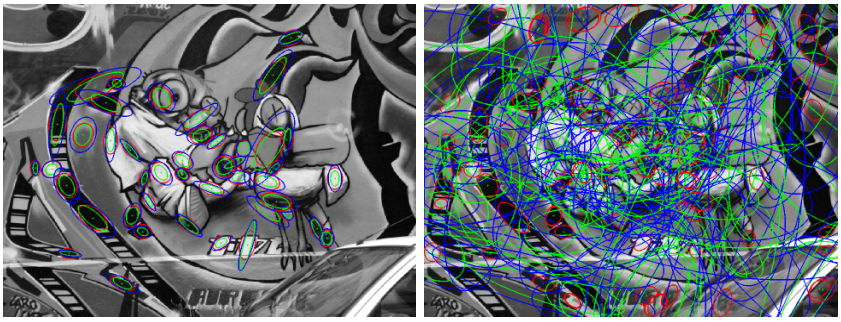

Figure 9: Comparaison of three approaches of Hessian matrix computation on the third octave. Left: Lowe's SIFT. Right: improved SIFT without sub-sampling between octaves. The pre-zoom is used in both SIFT schema. The covariance matrix is plot as ellipse on the image. The uncertainty is proportional to the size of ellipse. Red indicates the direct approach, green for residual based approach and blue for derivative based approach. Remark that three approaches give the similar Hessian matrix for Lowe's SIFT, while only the direct approach gives the reasonable Hessian matrix for improved SIFT.

In localization error evaluation, the improvement in precision has been observed in translation and rotation case while it is not very evident in case of zoom, tilt or affine transformation due to blur inconsistency and inaccurate descriptor. The localization uncertainty has the advantage that it depends only on the detection process itself and does not suffer from other errors. In Lowe's SIFT, the Hessian matrix H can be computed by any of the three approaches on the scale σ where the local extrema $\mathbf{x} = (x, y, \sigma)$ is detected. More precisely, the Hessian matrix should be computed on an interpolated sub-interval $\hat{\sigma} = \sigma + \Delta\sigma$, which is determined by the 3D interpolation in Eq. (7). This H is computed on the local scale $\hat{\sigma}$ and should be propagated to the initial image resolution for practical applications. The equivalent Hessian matrix H_0 on initial resolution can be obtained by multiplying H by a factor:

$$H_0 = \frac{1}{(2^{oct})^2} H \tag{24}$$

To reduce the computation complexity, the Hessian matrix is computed directly on the scale σ without degrading a lot the performance. This changes a little the multiplicative factor: $H_0 = \frac{1}{(2^{oct}+\Delta\sigma)^2} H$. This rescaling implies

that features detected in coarse scales have larger uncertainty. Without this back-propagation, SIFT features detected in different octaves have the similar covariance since images have the same nature in blur. This is the main difference between the scale invariant feature detectors and non scale-invariant scale detectors. For non scale-invariant detectors, like Harris corner, the detected corners are almost homogeneous without scale change while SIFT features are inhomogeneous due to the above multiplicative factor in back-propagation. Another difference is that Harris corners have isotropic covariance matrix while it is not the case for SIFT features. The isotropy of SIFT features depends on the threshold of edge response in SIFT, which is the ratio between two eigenvalues of Hessian matrix. To eliminate the features along straight edges, this threshold is set to be no bigger than 10, which keeps features on corners, contours and junctions. These features do not all have isotropic covariance matrix (see Fig. 10 for different types of covariance matrix).

Another interesting question is whether the Hessian matrix is better estimated in improved SIFT schema than in Lowe's SIFT. In Lowe's SIFT, the Hessian matrix is multiplied by the factor $\frac{1}{(2^{oct})^2}$. In improved SIFT, images are zoomed by factor *oct* compared to their counterparts in Lowe's SIFT. By Eq. (2), if these images all contain Gaussian blur bigger than 1.6, the same factor $\frac{1}{(2^{oct})^2}$ will be introduced in the Hessian matrix. In Fig. 11, the Hessian matrix of local extrema is compared octave by octave for Lowe's SIFT and improved SIFT. Note that not all of the feature points in Lowe's SIFT are found by improved SIFT and improved SIFT also finds new features points. This means a well-sampled image can be approriately interpolated to recover a high resolution image. Former local extrema are not necessarily still local extrema, while new local extrema can appear. Yet, most of features are repeatedly detected by two SIFT schema. Their covariance matrix seems to be very similar. This means that no big improvement in localization uncertainty by canceling the subsampling in SIFT scale space. Even if the Hessian matrix does not change a lot in two SIFT schema, the small change is sufficient to estimate a better offset for refinement in Eq. (7). In addition, more precise integer pixel position of local extrema and gradient estimation also contribute to precise feature position.

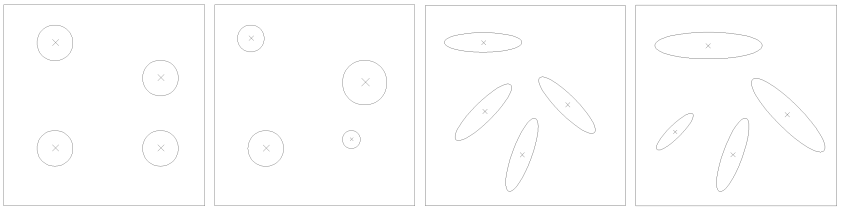

Figure 10: Illustration of covariance matrix with different homogeneity or isotropy. From left to right: homogeneous isotropic, nonhomogeneous isotropic, homogeneous anisotropic and nonhomogeneous anisotropic.

5 Conclusion

SIFT and its improved version are evaluated by average localization error and element-wise localization uncertainty. The first criteria suffers from blur incon-

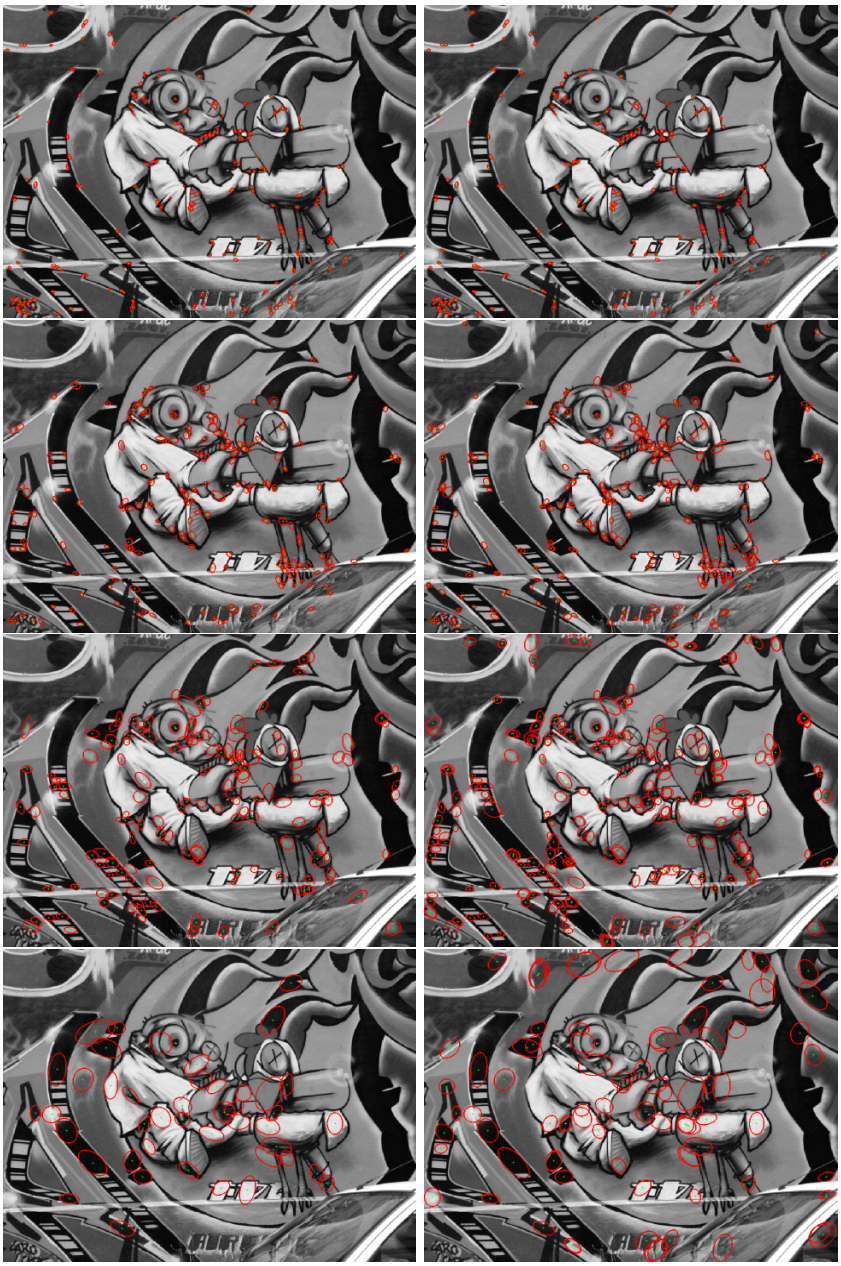

Figure 11: Covariance matrix is displayed by ellipse. Big ellipse means large uncertainty. The first column: Lowe's SIFT from octave 0 to octave 3. The second column: improved SIFT from octave 0 to octave 3.

sistency, descriptor inaccuracy and matching error, while the second criteria requires only one image to evaluate the uncertainty introduced in detection process. The uncertainty of feature points does not change a lot for both SIFT schema if images contain enough Gaussian blur. On the other hand, enough blur is also needed in improved SIFT to better compute gradient and Hessian matrix, which contributes to decrease localization error. To really make the gain in localization error useful, a more precise descriptor and matching procedure should be considered to avoid the blur inconsistency.

Bibliography

[1] P. Rockett A. Ashbrook, N. Thacker and C. Brown. Robust recognition of scaled shapes using pairwise geometric histograms. *British Machine Vision Conference*, pages 503–512, 1995.

[2] Joan Serrat Juan J. Villanueva Antonio M. Lopez, Felipe Lumbreras. Evaluation of methods for ridge and valley detection. *IEEE Transactions on Pattern Analysis and Machine*, 21(4):327–335, 1999.

[3] .F Schweiger E. Steinbach N. Navab B. Zeisl, P.F. Georgel. Estimation of location uncertainty for scale invariant feature points. *BMVC*, 2009.

[4] S. Baker and S.K. Nayar. Global measures of coherence for edge detector evaluation. *IEEE Conference on Computer Vision and Pattern Recognition*, 2:373–379, 1999.

[5] A. Baumberg. Reliable feature matching across widely separated views. *Computer Vision and Pattern Recognition*, pages 774–781, 2000.

[6] Dougherty S. Bowyer K., Kranenburg C. Edge detector evaluation using empirical roc curves. *Computer Vision and Image Understanding*, 84(1):77–103, 2001.

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

[8] Joseph L. Mundy David A. Forsyth Andrew Zisserman Christopher Coelho, Aaron Heller. An experimental evaluation of projective invariants. *Mit Press Series Of Artificial Intelligence Series*, pages 87–104, 1992.

[9] W. Forstner. Reliability analysis of parameter estimation in linear models with applications to mensuration problems in computer vision. *Computer Vision Graphics Image Process*, 40:273310, 1987.

[10] D. Gabor. Theory of communication. *Journal IEE*, 3(93):429–457, 1946.

[11] C. Harris and M. Stephens. A combined corner and edge detector. *Alvey Vision Conference*, 15:147–151, 1988.

[12] L. Van Gool Herbert Bay, Tinne Tuytelaars. Surf: Speeded up robust features. *Computer Vision and Image Understanding*, 110(3), 2008.

[13] K. Heyden, A. Rohr. Evaluation of corner extraction schemes using invariance methods. *International Conference on Pattern Recognization*, 13:895–899, 1996.

[14] M. Urban J. Matas, O. Chum and T. Pajdla. Robust wide-baseline stereo from maximally stable extremal regions. *Image and Vision Computing*, 22(10):761767, 2004.

[15] A. Johnson and M. Hebert. Object recognition by matching oriented points. *Computer Vision and Pattern Recognition*, pages 684–689, 1997.

[16] C Schmid K Mikolajczyk. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine*, 27(10):1615–1630, 2005.

[17] Y. Kanazawa and K. Kanatani. Do we really have to consider covariance matrices for image features? *IEEE International Conference on Computer Vision*, 2, 2001.

[18] Cordelia Schmid Krystian Mikolajczyk. An affine invariant interest point detector. *ECCV*, pages 128–142, 2002.

[19] Cordelia Schmid Krystian Mikolajczyk. Scale and affine invariant interest point detectors. *IJCV*, 60(1):63–86, 2004.

[20] T. Moons L. Van Gool and D. Ungureanu. Affine/photometric invariants for planar intensity patterns. *European Conference Computer Vision*, pages 642–651, 1996.

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

[22] S. Winder M. Brown, R. Szeliski. Multi-image matching using multi-scale oriented patches. *CVPR*, 2005.

[23] Thomas Sanocki Kevin W. Bowyer Michael D. Heath, Sudeep Sarkar. Robust visual method for assessing the relative performance of edge-detection algorithms. *IEEE Transactions on Pattern Analysis and Machine*, 19(12):1338–1359, 1997.

[24] D. Gawley M.J. Brooks, W. Chojnacki and A. van den Henge. What value covariance information in estimating vision parameters? *IEEE International Conference on Computer Vision*, 1, 2001.

[25] L. Moisan and B. Stival. A probabilistic criterion to detect rigid point matches between two images and estimate the fundamental matrix. *International Journal of Computer Vision*, 57(3):201218, 2004.

[26] J.M. Morel and G.Yu. Asift: A new framework for fully affine invariant image comparison. *SIAM Journal on Imaging Sciences*, 2(2):438–469, 2009.

[27] J.M. Morel and G.Yu. On the consistency of the sift method. *Preprint, CMLA, ENS-Cachan*, 2009.

[28] D. D. Morris and T. Kanade. A unied factorization algorithm for points, line segments and planes with uncertainty models. *International Conference Computer Vision*, page 696702, 1998.

[29] U. Orguner and F. Gustafsson. Statistical characteristics of harris corner detector. *IEEE Workshop on Statistical Signal Processing (SSP)*, page 571575, 2007.

[30] Brand P. and Mohr R. Accuracy in image measure. *Proceedings of the SPIE Conference on Videometrics III*, 2350:218–228, 1994.

[31] F. Cao P. Musé, F. Sur and Y. Gousseau. Unsupervised thresholds for shape matching. *International Conference on Image Processing*, 2003.

[32] F. Cao J.L. Lisani P. Musé, F. Sur and J.M. Morel. *Three-Dimensional Computer Vision: A Geometric Viewpoint*. Mit Press, 2007.

[33] F. Cao Y. Gousseau P. Musé, F. Sur and J.M. Morel. An a contrario decision method for shape element recognition. *International Journal of Computer Vision*, 69(3):295315, 2006.

[34] Javier Ruiz-del-solar Patricio Loncomilla. Improving sift-based object recognition for robot applications. *Lecture Notes in Computer Science*, pages 1084–1092, 2005.

[35] J. Delon Rabin and Y. Gousseau. A statistical approach to the matching of local features. *SIAM Journal Imaging Science*, 2(3):931–958, 2009.

[36] J. Malik S. Belongie and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Transaction Pattern Analysis and Machine Intelligence*, 2:509–522, 2002.

[37] C. Schmid S. Lazebnik and J. Ponce. Sparse texture representation using affine-invariant neighborhoods. *Computer Vision and Pattern Recognition*, pages 319–324, 2003.

[38] F. Schaffalitzky and A. Zisserman. Multi-view matching for unordered image sets. *European Conference Computer Vision*, pages 414–431, 2002.

[39] F. Schaffalitzky and A. Zisserman. Multi-view matching for unordered image sets, or "how do i organize my holiday snaps?". *ECCV*, 1:414–431, 2002.

[40] J. Shi and C. Tomasi. Good features to track. *IEEE Conference Computer Vision Pattern Recognition*, page 1994, 593600.

[41] A. Singh. An estimation-theoretic framework for image-ow computation. *International Conference Computer Vision*, page 168177, 1990.

[42] R.M. Steele and C. Jaynes. Feature uncertainty arising from covariant image noise. *IEEE Conference on Computer Vision and Pattern Recognition*, 1, 2005.

[43] J.M. Morel T. Buades, Y. Lou and Z. Tang. A note on multi-image denoising. *Proceeding of the International Workshop on Local and Non-Local Approximation (LNLA) in Image Processing*, 2009.

[44] A. Zisserman T. Kadir and M. Brady. An ane invariant salient region detector. *European Conference on Computer Vision*, page 228241, 2004.

[45] et al T. Tuytelaars, L. Van Gool. Content-based image retrieval based on local anely invariant regions. *Int. Conf. on Visual Information Systems*, page 493500, 1999.

[46] T. Tuytelaars and L. Van Gool. Wide baseline stereo matching based on local, anely invariant regions. *British Machine Vision Conference*, page 412425, 2000.

[47] T. Tuytelaars and L. Van Gool. Matching widely separated views based on ane invariant regions. *International Journal of Computer Vision*, 59(1):61–85, 2004.

[48] R. Zabih and J. Woodfill. Non-parametric local transforms for computing visual correspondance. *European Conference Computer Vision*, pages 151–158, 1994.