

Research Article

MOCC: A Fast and Robust Correlation-Based Method for Interest Point Matching under Large Scale Changes

Feng Zhao,^{1,2} Qingming Huang,³ Hao Wang,² and Wen Gao⁴

¹ School of Information Engineering, Beijing University of Posts and Telecommunications, Beijing 100876, China

² Nokia Research Center, Beijing 100176, China

³ School of Information Science and Engineering, Graduate University of Chinese Academy of Sciences, Beijing 100049, China

⁴ School of Electronics Engineering and Computer Science, Peking University, Beijing 100871, China

Correspondence should be addressed to Feng Zhao, zerof321@live.com

Received 22 February 2010; Revised 5 July 2010; Accepted 30 September 2010

Academic Editor: Mei-Juan Chen

Copyright © 2010 Feng Zhao et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Similarity measures based on correlation have been used extensively for matching tasks. However, traditional correlation-based image matching methods are sensitive to rotation and scale changes. This paper presents a fast correlation-based method for matching two images with large rotation and significant scale changes. Multiscale oriented corner correlation (MOCC) is used to evaluate the degree of similarity between the feature points. The method is rotation invariant and capable of matching image pairs with scale changes up to a factor of 7. Moreover, MOCC is much faster in comparison with the state-of-the-art matching methods. Experimental results on real images show the robustness and effectiveness of the proposed method.

1. Introduction

Matching two images of the same scene or object is one of the fundamental problems in computer vision. Image matching plays an important role in many applications such as stereo correspondence, motion analysis, image registration, and image/video retrieval. It has been an extensively studied topic for the last several decades, and a large number of matching algorithms have been proposed in the literature [1–3].

The methods for image matching can be broadly divided into two classes: area-based matching and feature-based matching. Area-based matching directly compares the gray value distribution in image patches, and the similarity is measured by cross-correlation or least-squares techniques. Feature-based matching extracts salient features such as corners in the two images and then establishes reliable feature correspondences by comparing the feature descriptors. There also have been some matching methods that can be regarded as the combination of the two classes [4, 5].

Normalized cross-correlation is widely used as an effective similarity measure for matching applications. Normalized cross-correlation is invariant to linear brightness and contrast variations, and its easy hardware implementation

makes it useful for real-time applications. However, traditional correlation-based image matching methods will fail when there are large rotations or significant scale changes between the two images. This is because the normalized cross-correlation is sensitive to rotation and scale changes. There also exist generalized versions of cross-correlation that calculate the cross-correlation for each assumed geometric transformation of the correlation windows [6, 7]. Although they are able to handle more complicated cases, the computational load grows very fast in the mean time.

In this paper, we propose a fast and robust method for matching two uncalibrated images based on normalized cross-correlation. Our work addresses the problem of matching image pairs with large rotation and significant scale changes, which cannot be efficiently solved by traditional correlation-based methods. We first build a multiscale pyramid for each image and extract corner points as feature points in each level of the pyramid. Compared with other multiscale feature point detectors, our implementation is simple and fast. Only one Gaussian smoothing operation is required for building a multiscale pyramid, and there is no scale-space extrema detection included. Each feature point is assigned with one dominant orientation. Then a multilevel

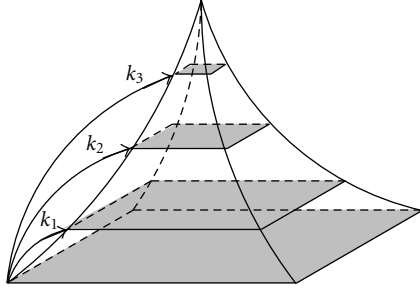


FIGURE 1: Fast multiscale pyramid representation.

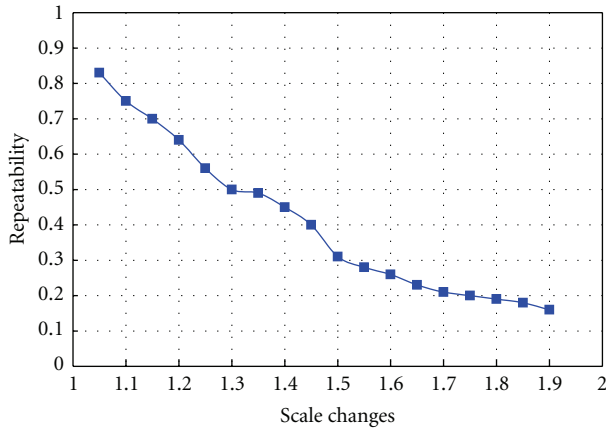


FIGURE 2: Repeatability of Harris points under different scale changes.

matching strategy is used to establish the correspondence of feature points. The multilevel matching strategy makes our method more efficient by removing the redundant computation in the matching procedure.

For similarity measure between two feature points, we adopt the rotation invariant normalized cross-correlation. The orientation of the correlation window is determined by the dominant orientation of the feature point to achieve rotation invariance. Moreover, both the shape and the size of the correlation window are fixed, which contributes to the simplicity of our method. The epipolar geometry constraint is imposed to reject the false matches. We also provide an effective method to further improve the quality of matching results based on the average differences of dominant orientations. Experimental results on real images of various contents demonstrate that our method can deal with large rotation and significant scale changes efficiently. The method can also tolerate weak affine deformations and is robust to illumination changes, occlusion, and other content changes.

The remainder of this paper is organized as follows. In Section 2, we briefly review the related work. Section 3 describes the multiscale feature point detection algorithm. Section 4 presents the definition and calculation of similarity measure in detail. Section 5 focuses on the multilevel matching strategy. Section 6 describes false match rejection by imposing epipolar geometry constraint and dominant

orientation constraint. Experimental results are presented in Section 7, and conclusions are given in Section 8.

2. Related Work

The scale space representation of a digital image is a set of images represented at various levels of resolutions [8]. It can be built by sequential smoothing of the original image with kernels of different scales. The problem of finding the *characteristic* scale of a local image structure has been studied in depth by Lindeberg [9]. The concept of automatic scale selection was used by Lowe [10] and Mikolajczyk and Schmid [11] for scale invariant feature detection. The multiscale representation of image can also be built as an image pyramid, which is a collection of copies of the original image at different sizes. The typical method is to smooth the higher resolution image with Gaussian kernels and then downsampled it by the corresponding scale factor to create the lower resolution level. Using image pyramid to obtain computational efficiency in image processing can be traced back to the early eighties of the last century [12]. The extensive study of the pyramid representation and its property can be found in the studies by Burt and Adelson [13] and Crowley and Parker [14].

Feature points (also called interest points) are characteristic points in an image. They are stable and distinctive image locations, and have high information content. Feature points are useful in many research areas such as content-based image retrieval [15]. Numerous approaches for feature point detection exist in the literature. Corners are highly informative image locations, and they are considered as good candidates for feature points. The early work of using corners for image matching is the study by Moravec [16] on stereo matching. Among the most popular corner detectors, the Harris corner detector [17] is known to be robust against camera noise, image rotation, and illumination changes [18]. Using Harris corners as feature points has been proved to be effective for image matching applications [5, 19, 20].

However, the Harris corner detector is very sensitive to changes in image scale. Its repeatability rate significantly decreases when the scale change between two images is large. In order to deal with the problem of scale changes, multiscale versions of Harris detector are proposed. Dufournaud et al. [21] proposed the scale-adapted Harris detector for matching images with different resolutions. Zhao et al. [22] introduced multiscale Harris detector that detected Harris corners in pyramid representation. Mikolajczyk and Schmid [11] proposed a method for detecting scale invariant Harris corners by finding local maxima in scale space.

There are several other scale invariant feature point detectors, such as the salient region detector proposed by Kadir and Brady [23] and the edge-based region detector proposed by Jurie and Schmid [24]. Focusing on the speed, Lowe [10] proposed a method based on Difference of Gaussian (DoG) for scale invariant feature detection. Recently, Bay et al. [25] proposed a fast feature detector, SURF. SURF is based on Hessian matrix, and it utilizes

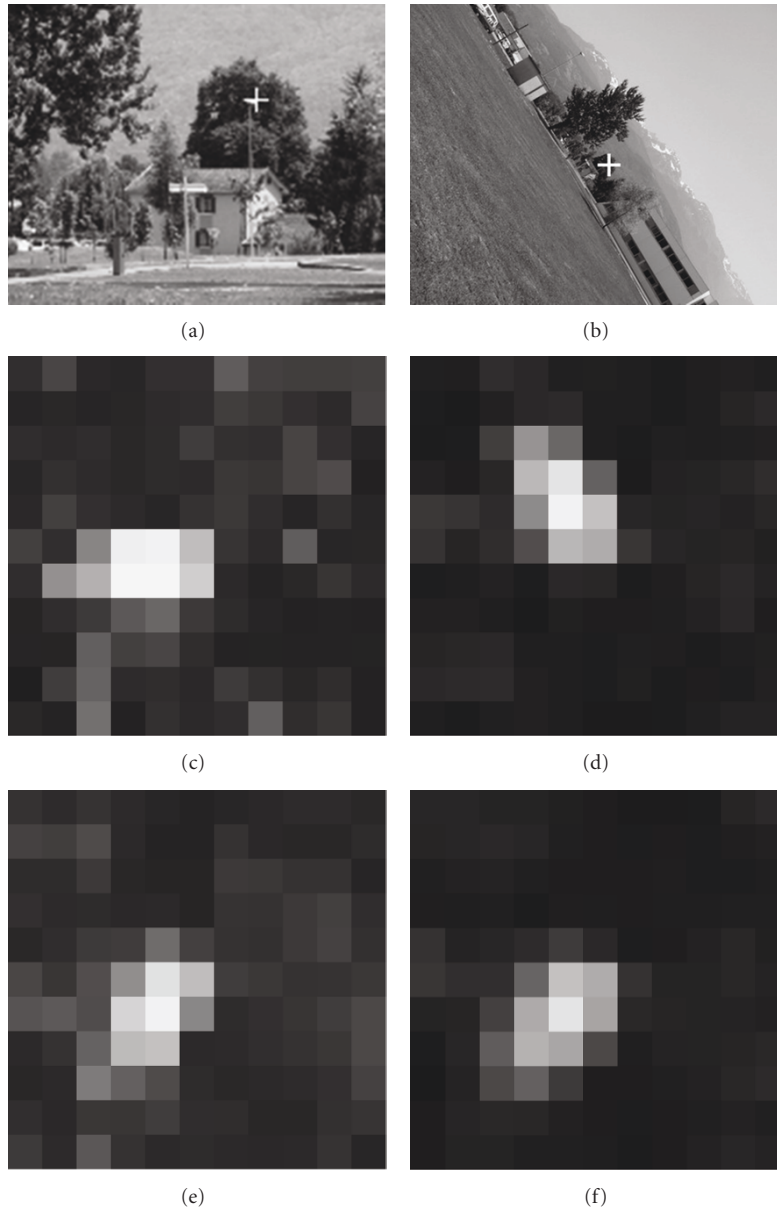


FIGURE 3: Matching feature points under remarkable scale changes. (a) The feature point on fourth pyramid level for the first image. (b) The feature point on first pyramid level for the second image. (c) and (d) 11×11 correlation window of the two feature points before orientation normalization. (e) and (f) 11×11 correlation window of the two feature points after orientation normalization.

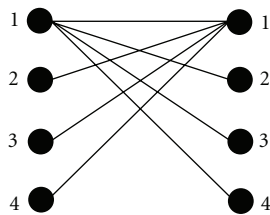


FIGURE 4: Matching between feature point groups.

integral images to reduce the computation time. With the development of local features, there has recently been an impressive body of work on matching images taken from very

different viewpoints with affine covariant features [20, 26–29].

Our work is closely related to that of Dufournaud et al. [21] that attempted to match two images with scale changes. Their method is able to deal with very different resolutions but requires prior knowledge on the scale change between the two images. To compute the Mahalanobis distance of the invariant descriptors, the covariance matrix should be estimated over a large set of image samples. Note that Brown et al. [30] also introduced a matching method based on a multiscale Harris detector, but they concentrated on the panoramic image stitching application, where the scale variation between images is expected to be fairly small.

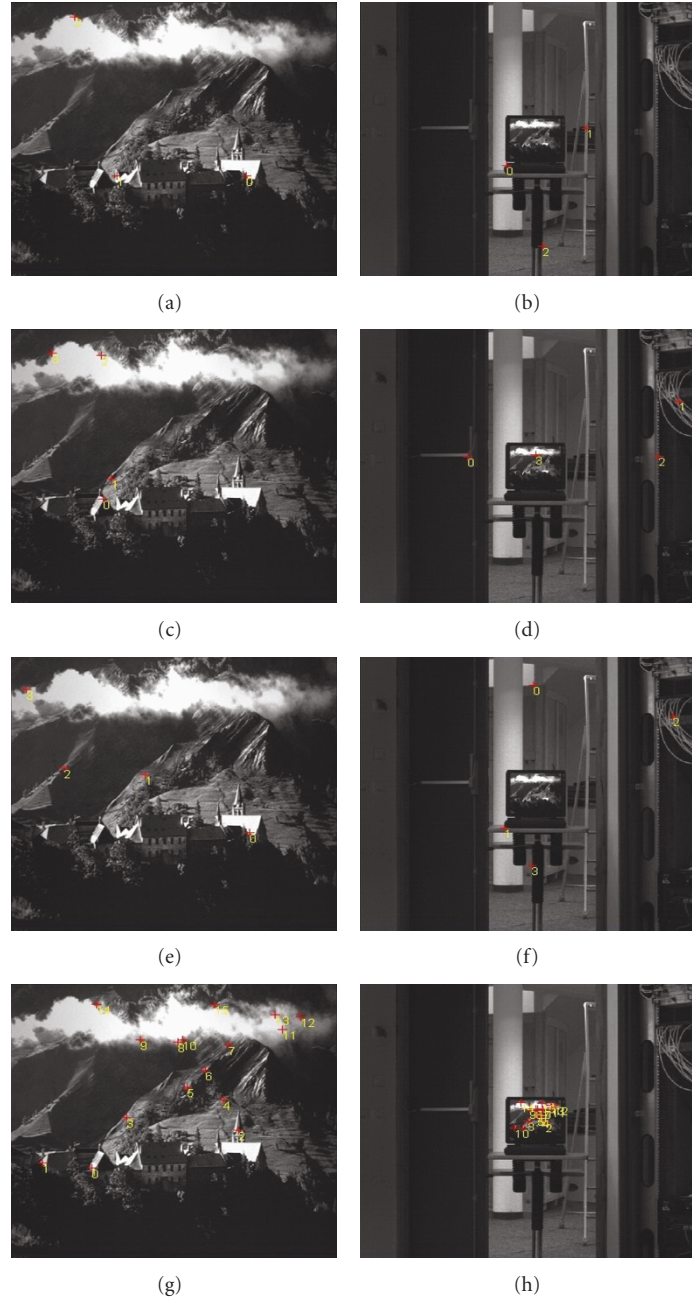


FIGURE 5: Group-to-group matching results for image pair Laptop with significant scale changes. (a), (c), (e), and (g) Feature points in matching results on pyramid levels 1, 2, 3, and 4 for the first image. (b), (d), (f), and (h) Corresponding feature points in matching results on pyramid level 1 for the second image.

3. Multiscale Feature Point Detection

We propose a multiscale feature detection strategy in our method. A fast multiscale corner detector is used to extract feature points in different scale levels.

3.1. Multiscale Pyramid Representation. We first build a multiscale pyramid representation for the image. The pyramid consists of four levels. The first level of the pyramid is the image itself. Other levels of the pyramid are created by

sampling the image with a set of scale factors k_n ($n = 1, 2, 3$). The original image is smoothed by a Gaussian function with $\sigma_{\text{init}} = 1$ before downsampling. The scale factor k_n should be chosen carefully since it greatly affects the matching result. Figure 1 shows the pyramid scale space representation.

The standard Harris detector will give a low repeatability rate when the scale change between two images is beyond 1.5 [18]. Hence, the scale changes that can be handled by the traditional matching methods based on standard Harris detector are in the range of $[1/\sqrt{2}, \sqrt{2}]$. This means that

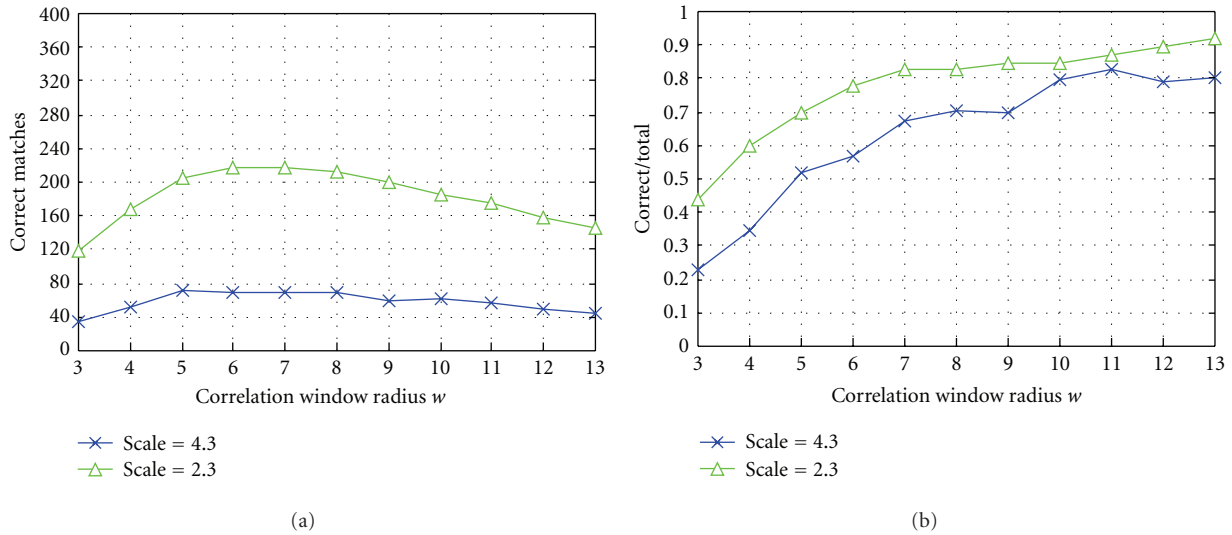


FIGURE 6: The evaluation for different correlation window radii. (a) This graph shows the number of correct matches as a function of radius of the correlation window under different scale changes. (b) This graph shows the ratio of correct matches to total matches as a function of radius of the correlation window under different scale changes. The results are obtained from images in “Boat” sequence (frame 0, 5, and 9). The scale change between frames 0 and 5 is 2.3, and the rotation angle is about 9 degrees. The scale change between frames 0 and 9 is 4.3, and the rotation angle is about 45 degrees.

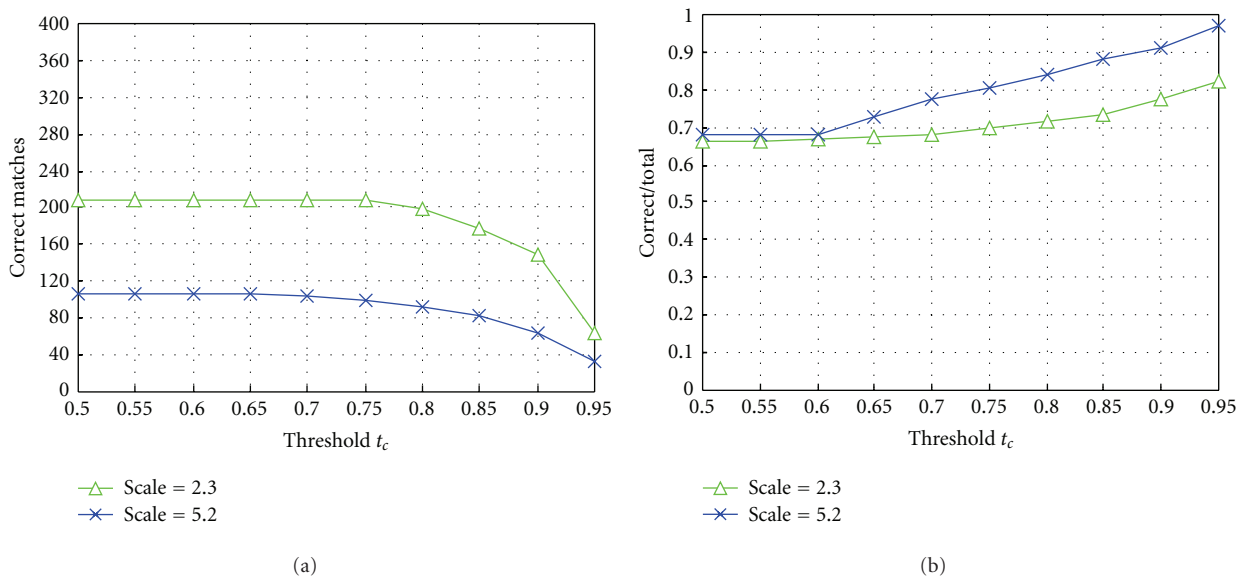


FIGURE 7: Evaluation for different threshold t_c . (a) This graph shows the number of correct matches with respect to the threshold t_c under different scale changes. (b) This graph shows the ratio of correct matches to total matches with respect to the threshold t_c under different scale changes. The results are obtained from images in “Boat” sequence (frames 0 and 5) and images in “East_south” sequence (frames 0, and 9). The scale change between images in “Boat” sequence is 2.3, and the rotation angle is about 9 degrees. The scale change between images in “East_south” sequence is 5.2, and the rotation angle is about 59 degrees.

the scale change between consecutive scale levels should not exceed $2(\sqrt{2} * \sqrt{2})$. Considering the above points and after experimentation with different sets of scale factors, we choose the set of scale factors $\{1/2, 1/4, 1/5\}$ in our method as it provides the most stable results. Experimental results on repeatability of Harris points under different scale changes are illustrated in Figure 2 to explain how to determine the scale factors.

The test image set contains images with scale changes from 1.05 to 1.9. At each scale change level, we get 5 pairs of images in order to calculate the average repeatability under this scale change level. Given two images, Harris points are detected in them separately. The repeatability is defined as the ratio between the number of potential point matches and the smaller of the numbers of detected Harris points in the pair of images. Homography between image pairs is

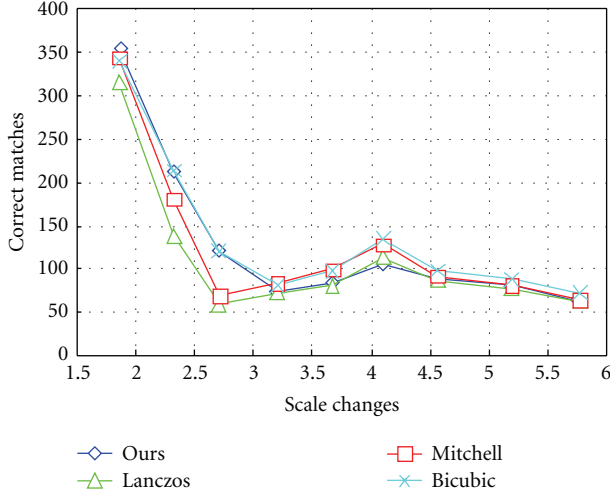


FIGURE 8: Performance evaluation of different downsampling methods.

used to compute the number of potential point matches, which indicate how many point pairs correspond to the same location of the scene. If the repeatability is too small, the following matching process will probably fail because there are not enough matched points.

From Figure 2, we can see that the Harris detector will give a low repeatability rate when the scale change between two images is beyond 1.5. Further experiments show that the matching results will be unstable when the scale change between two images is beyond 1.5. Therefore, the maximal scale change covered by standard Harris detector is around $1.5 \approx \sqrt{2}$. In order to cover all the scales between consecutive scale levels, the scale factor should not exceed $\sqrt{2} * \sqrt{2} = 2$.

Compared with the traditional Gaussian pyramid representation, only one Gaussian smoothing operation is required for building the multiscale pyramid and each level is downsampled from the smoothed original image instead of its lower level.

There are five points to be addressed here. Firstly, the selection of sampling factors k_n determines the scale factors between consecutive levels. If the scale factor between consecutive levels is too large, the matching procedure will fail for image pairs with certain scale change. For example, if we set k_1 to 0.4, it means that the scale factor between the first two levels is 2.5. Then the method will fail when the scale factor between two images falls into the range of [1.5, 1.7].

Secondly, the order of smoothing and downsampling operations also affects the performance of the method. Traditional Gaussian pyramid is built by smoothing current level and downsampling by a constant scale factor to obtain the higher level. We find that the matching results will be very poor with such pyramid when the scale change between two images is large. As for the downsampling operation, we simply smooth the original image with a Gaussian function and then downsample the smoothed one using bilinear interpolation. We have tried more complex approaches such as Bicubic, Lanczos, and Mitchell. But these approaches give hardly distinguishable matching results. We will provide

more discussions on this issue with experimental results in Section 7.2.

Thirdly, using pyramid representation introduces coordinate transformation. The coordinates of the feature points detected in different levels are relative to their levels. They should be mapped to the original image after matching procedure if they are not detected in the first level. Therefore, the error analysis should be performed for the matching results with the transformed coordinates.

Fourthly, the third scale factor is set to 1/5 instead of 1/8. We find that using 1/8 as the third scale factor results in too few features in the fourth level when the image resolution is under $1024 * 768$. Therefore, we increase the third scale factor for more stable results at the cost of less scale changes that can be handled. However, the proposed method can match image pairs with scale changes up to a factor of about 7 ($5 * \sqrt{2}$) when the third scale factor is set to 1/5. We believe that it is acceptable for most applications.

The last point we want to address is the number of the scale images in pyramid's generation. For each image, we generate 3 scale images besides the original image, which are directly downsampled by the scale factors from the smoothed original image. The original image is treated as the first scale level. Therefore, the scale representation in our method consists of 4 scale levels for each image as shown in Figure 1. There is a tradeoff between the speed and the capability of handling large scale changes. In current implementation, we utilize 4 scale images to cover the scale changes under 7. However, if the approximate scale changes between images are known, we could generate less scale images to achieve further speeding up. For example, if the scale changes between images are always less than 2.8 ($2 * \sqrt{2}$), we could generate only 1 extra scale image for matching purpose.

3.2. Feature Point Extraction. Feature points are then extracted using a standard Harris corner detector in each level of the multiscale pyramid with a constant parameter set. The Harris corner detector is based on the autocorrelation matrix, which is built as follows:

$$M = g(\sigma_h) * \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}, \quad (1)$$

where I_x and I_y indicate the x and y directional derivatives, respectively. The autocorrelation matrix performs a smoothing operation on the products of the first derivatives by convolving with a Gaussian window function. Since we use $\sigma_h = 1$ in each pyramid level, two 1D Gaussian convolutions in the x -direction and y -direction are applied to perform the Gaussian smoothing. It takes 8 multiplications and 12 additions per pixel, while the fastest Gaussian recursive filter [31] requires 14 multiplications and 6 additions per pixel.

The Harris corner strength measure is then calculated from the determinant and the trace of this matrix:

$$C_H = \det(M) - \alpha \text{trace}^2(M), \quad (2)$$

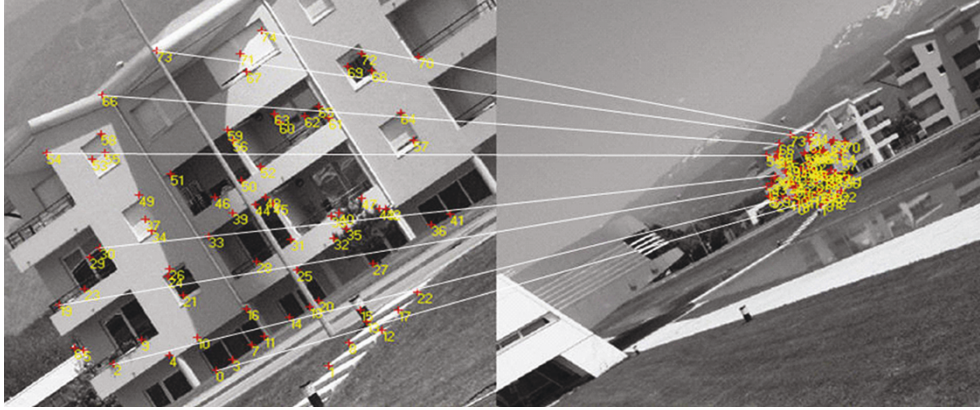


FIGURE 9: Matching result for image pair Residence (frames 0 and 9 of “Resid” sequence). The scale factor is 4.7, and the rotation angle is 5 degrees.



FIGURE 10: Matching result for image pair Boat (frames 0 and 9 of “Boat” sequence). The scale factor is 4.3, and the rotation angle is 45 degrees.

where α is a constant. A threshold t_h is used to select corner points. A point is identified as a corner if $C_H > t_h$ and C_H is the local maximum in its 8 neighborhoods. In our implementation, I_x and I_y are computed by convolution with the mask $[-1 \ 0 \ 1]$ for computation efficiency. The parameter α is set to be 0.04. In order to select corners with high significance, threshold t_h is set to be 15000. To improve the accuracy of the localization, a parabola is fit to the 3 values of C_H closest to each corner to interpolate the corner position. The interpolations are processed in x - and y -directions separately.

Each feature point is assigned one dominant direction to achieve invariance to rotation. We adopt the histogram-based approach for dominant orientation assignment [10]. Some modifications are made in order to adapt this approach for our implementation. An orientation histogram with 36 bins covering the range of 360 degrees is used to accumulate the local gradient orientations within a square region centered on a feature point. The size of the region equals the size of the correlation window used in the following matching procedure, which is set to be 11×11 pixels. The pixel differences for computing the gradient magnitude and orientation are calculated on the pyramid level at which

the feature point is detected. The pixel value is obtained by smoothing with a Gaussian window function with $\sigma_p = 1$. The gradient orientation of each sample in the region is weighted by its gradient magnitude and by a Gaussian window function with $\sigma_r = 1.7$ before being added to the histogram. After building the orientation histogram, we perform a smoothing operation on the histogram by iterative local averaging of every 3 consecutive bins in a cyclical fashion. The orientation corresponding to the largest bin in the smoothed histogram is selected to be the dominant orientation of the feature point. We also have tried to assign multiple dominant orientations for one feature point as done in [10]. But performance improvement is very limited, and the number of features increases about 30%, which slows down the whole matching procedure observably.

4. Similarity Measure Based on Rotation Invariant Correlation

Traditional similarity measure based on correlation is not invariant to image rotation. In our method, rotation invariant normalized cross-correlation is used to evaluate

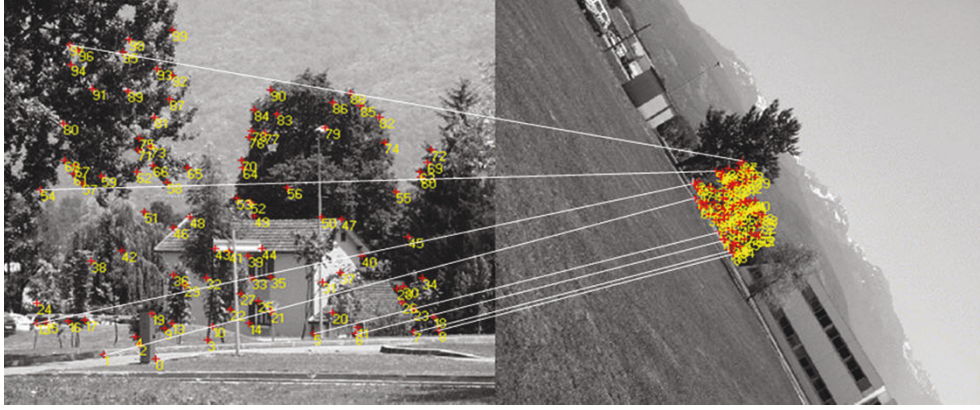


FIGURE 11: Matching result for image pair East_south (frames 0 and 9 of “East_south” sequence). The scale factor is 5.2, and the rotation angle is 59 degrees.

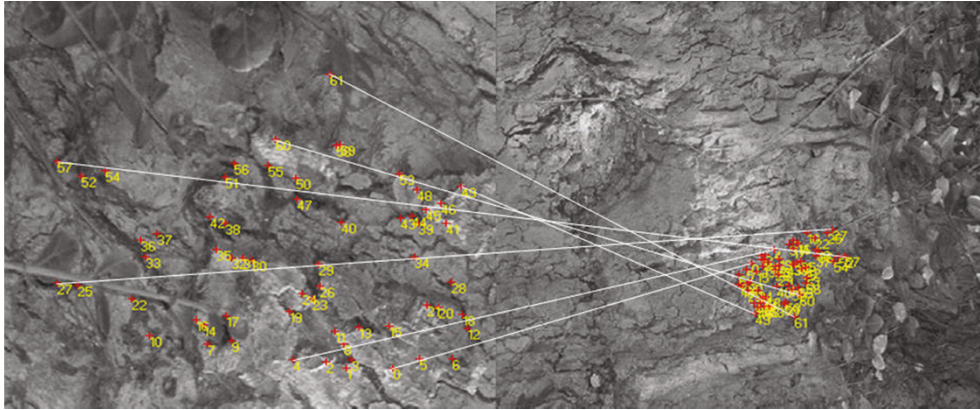


FIGURE 12: Matching result for image pair Bark (frames 1 and 6 of “Bark” sequence). The scale factor is 4.0, and the rotation angle is 154 degrees.

the difference between feature points. The orientation of the correlation window is determined by the dominant orientation of the feature point. Therefore, the similarity measure between feature points is invariant to rotation. Since our proposed method leverages both feature detection strategy and correlation-based similarity measure, it can be treated as a combination of feature-based matching and area-based matching.

4.1. Definition of the Similarity Measure. The definition of similarity measure is described as follows. Let $p = L_m(x, y)$ be a feature point at the m th pyramid level in the first image with dominant orientation θ_1 , and let $q = L_n(x', y')$ be a feature point at the n th pyramid level in the second image with dominant orientation θ_2 . W_1 and W_2 are two correlation windows of size $(2w + 1) \times (2w + 1)$ centered on each feature point. W'_1 is the correlation window generated by rotating W_1 clockwise by θ_1 around p , and W'_2 is the correlation window generated by rotating W_2 clockwise by θ_2 around q . Then W'_1 and W'_2 can be represented as two

$(2w + 1) \times (2w + 1)$ arrays of pixel intensities A and B :

$$\begin{aligned} A_{uv} &= L_m(x + u \cos \theta_1 - v \sin \theta_1, y + v \cos \theta_1 + u \sin \theta_1), \\ B_{uv} &= L'_n(x' + u \cos \theta_2 - v \sin \theta_2, y' + v \cos \theta_2 + u \sin \theta_2), \end{aligned} \quad (3)$$

where $u, v \in [-w, w]$. A_{uv} and B_{uv} are calculated using bilinear interpolation. The similarity measure between p and q is defined as

$$C_{pq} = \frac{\sum_{u=-w}^w \sum_{v=-w}^w [A_{uv} - \bar{A}] \cdot [B_{uv} - \bar{B}]}{(2w + 1)(2w + 1)\sigma(A)\sigma(B)}, \quad (4)$$

where $\bar{A}(\bar{B})$ is the average and $\sigma(A)(\sigma(B))$ is the standard deviation of all the elements in $A(B)$. As mentioned in Section 3, w is set to be 5 in our experiments. We will provide the detailed discussion on w selection in Section 7.1. Since the similarity measure is computed with respect to a canonical orientation, the matching procedure is invariant to

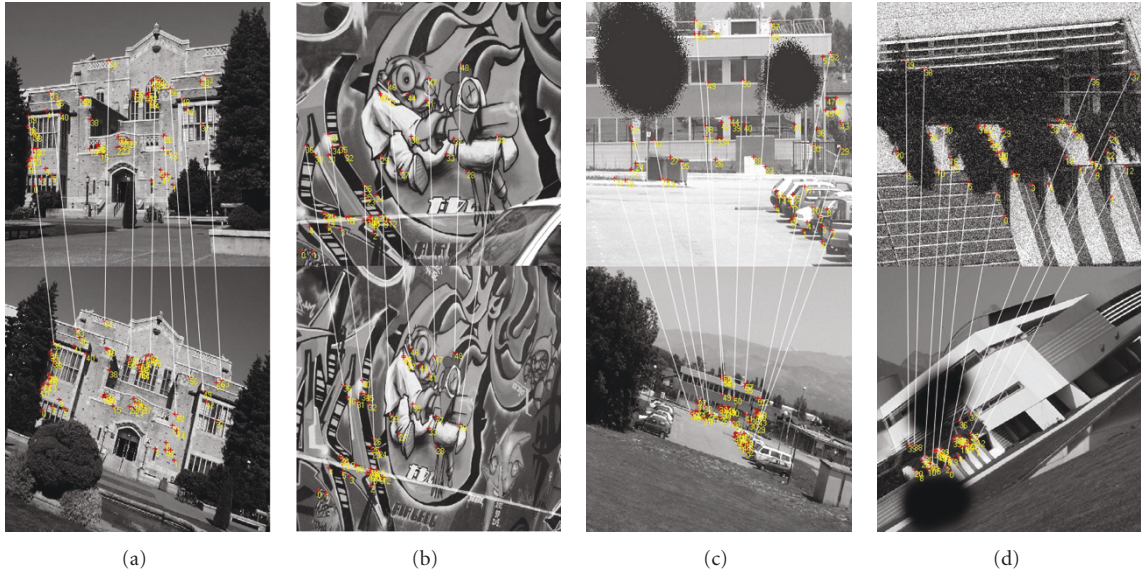


FIGURE 13: More matching results under viewpoint changes and other imaging conditions. (a) Matching result for image pair Graffiti (frames 1 and 3 of “Graffiti” sequence). The viewpoint angle is 30 degrees, and there are 49 matches, all of them are correct. (b) Matching result for image pair UBC (frames 8 and 10 of “UBC.v” sequence). The viewpoint angle is 30 degrees, and there are 65 matches with 1 false match. (c) Matching result for image pair East_park (frames 1 and 7 of “East_park” sequence). The scale factor is 3.3, and the rotation angle is 16 degrees. Occlusion and illumination changes are also added. There are 59 matches, all of them are correct. (d) Matching result for image pair Inria (frames 0 and 7 of “Inria” sequence). The scale factor is 3.4, and the rotation angle is 25 degrees. Occlusion and Gaussian noise are also added. There are 40 matches, all of them are correct.

image rotation. The similarity measure decreases monotonically from 1 to -1 with the increase of the difference between two feature points.

4.2. Acceleration of the Similarity Measure Calculation. We also employ a strategy that will help to restrict the total number of the feature points. We define NC_n ($n = 1, 2, 3$) as the threshold of the number of feature points in n th scale level. If the number of corners detected in n th scale level is larger than NC_n , all the detected corners in this level are sorted in decreasing order of corner strength measure C_H . Then we choose the first NC_n corners as feature points for matching. If the number of corners detected in n th scale level is smaller than NC_n , all the detected corners in this level are adopted as feature points. Considering the property of the pyramid representation, the values of NC_n are set to be $\{1500, 800, 600, 500\}$ in our implementation. Experimental results show that using this strategy can effectively speed up the matching procedure while almost not affecting the quality of matching result. For typical real images with medium resolution such as 850×680 pixels, the average number of feature points extracted using our multiscale corner detector is about 3000, with the above parameter setting.

The calculation of rotation invariant normalized cross-correlation can be further accelerated by the following strategy. Note that the size of the correlation window is invariant in the whole matching procedure. For a given feature point, the angle of rotation compensation is also

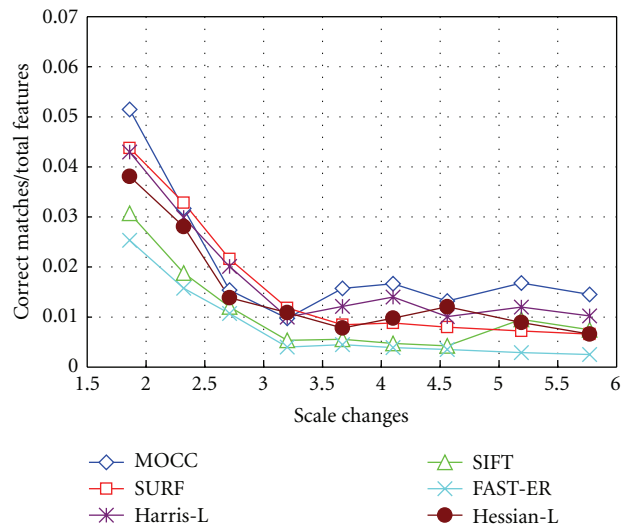


FIGURE 14: Performance evaluation.

fixed. Therefore, for (4), A_{uv} , B_{uv} , the average and the standard deviation of all the elements in A and B can be precalculated before the matching process. For each feature point, we use a feature array F to store all the elements and the standard deviation. The number of the pixels in the correlation windows is $N_F = 121$ (11×11). Thus, the length of F should be $N_F + 1$ because we need one more element for the standard deviation. The first N_F elements of F are the

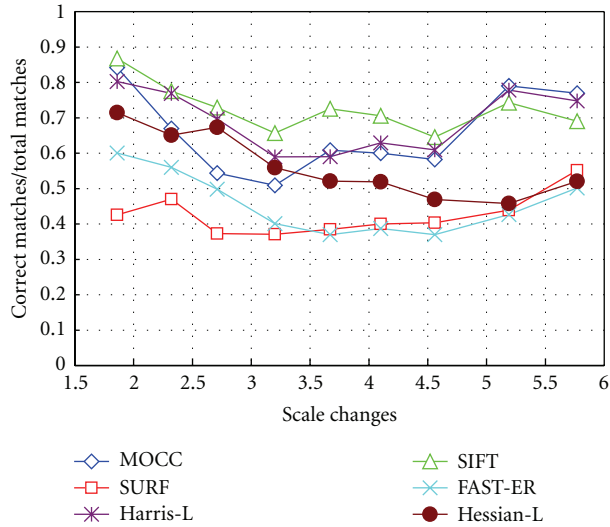


FIGURE 15: Precision comparison.

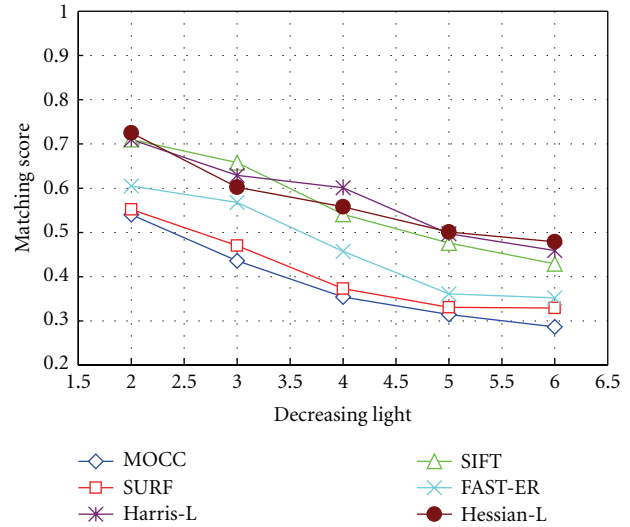


FIGURE 17: Matching results on images with illumination changes (Leuven sequence).

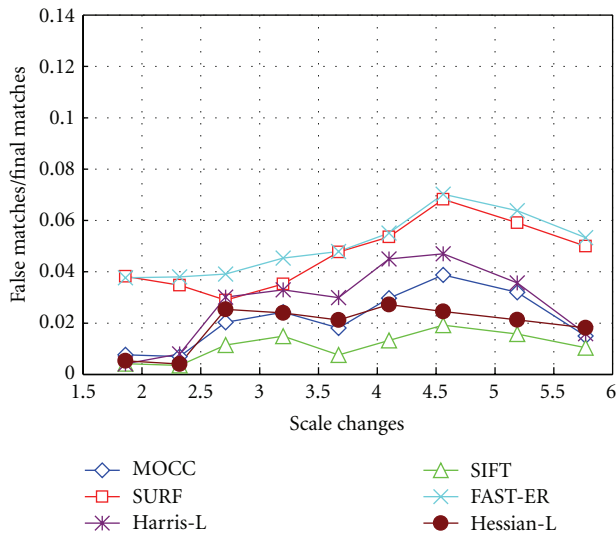


FIGURE 16: False match rate comparison.

gray values of the pixels in the correlation windows, which have been subtracted by the average. For two feature points, p and q , the feature array for each feature point is denoted as F_p and F_q , respectively. The similarity measure between them can be calculated as

$$C_{pq}^F = \frac{\sum_{i=1}^{N_F} F_p[i] \cdot F_q[i]}{N_F \cdot F_p[N_F + 1] \cdot F_q[N_F + 1]}. \quad (5)$$

Now we need to perform 123 multiplications, 120 additions, and 1 division to calculate the similarity measure between two feature points. Comparing with the similarity measure calculation in SIFT, if the length of feature vector is also 121, then we need to perform 121 multiplications and 241 additions to obtain the square of Euler distance between two feature descriptors. This strategy increases the speed of calculating rotation invariant normalized cross-correlation

by storing the precalculated correlation windows in the form of feature arrays.

4.3. The Effectiveness of the Similarity Measure. One may argue that the pyramid-building process violates the Nyquist theorem and the large sampling factor will cause aliasing problems. Then simple similarity measures that rely on gray value distribution will probably fail, especially when finding correspondences between the top level and the bottom level. We believe that the keys to the success of our method are the following two characteristics. The first one is the appropriate way of building the pyramid representation. As mentioned in Section 3, we have tried the traditional Gaussian pyramid, but it fails when the scale change between two images is large. The current pyramid representation indeed generates aliasing images for the top 2 levels, but it provides the good basis for corner detection and matching. The second one is the robustness of Harris corner detector. The high accuracy of Harris corner locations contributes to the successful matching under significant scale changes. Figure 3 demonstrates an example for matching feature points in two images with remarkable scale changes.

The two images used for matching test have the same resolution 850×680 , and the scale factor between them is 5.2. Figures 3(a) and 3(b) show the sample feature points (white cross) that are detected in the corresponding levels. The two feature points are correctly matched in our method. Note that Figure 3(a) is the image of the fourth level of the pyramid representation for the first image. Therefore, it is downsampled by a factor of 5 from the original resolution. Figure 3(b) is the second image itself (first pyramid level). We make them the same size for convenient illustration purpose. The real size of Figure 3(b) is five times larger than that of Figure 3(a). Figures 3(c) and 3(d) show the 11×11 correlation window of the two feature points before orientation normalization. Figures 3(e) and 3(f) show

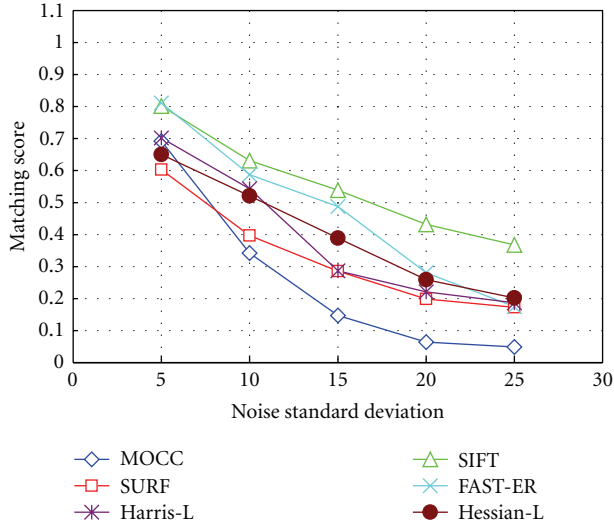


FIGURE 18: Matching results on images with Gaussian noise corruptions.

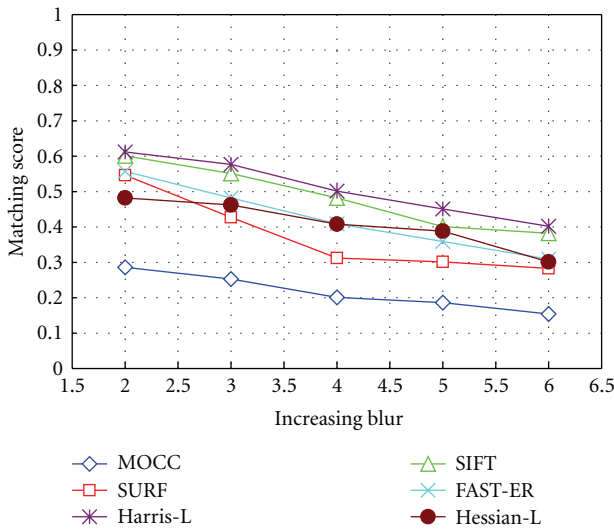


FIGURE 19: Matching results on images with blurring (Bikes sequence).

the correlation window of the two feature points after orientation normalization.

The dominant orientations of the two feature points are 330 degrees and 260 degrees, respectively. While the real rotation angle between the two images is 59 degrees, the rotation angle between the dominant orientations of the two feature points is 70 degrees. Experimental results on matching without orientation normalization show that the standard correlation-based matching can tolerate up to 20 degrees rotation. Therefore, the approach of using orientation normalization still works in our method.

We can see that the distribution of gray value for Figures 3(e) and 3(f) is similar (There are also some illumination changes between the two images.) The normalized cross-correlation score between Figure 3(e) and Figure 3(f) is

0.939, which means that the two feature points are very similar under this similarity measure.

4.4. Nearest Neighbor-Based Matching. Suppose that there are m feature points in the first group and n feature points in the second group. Consider a matrix $G \in M_{m,n}$ whose element G_{ij} stands for the similarity measure between the i th feature point in the first group and the j th feature point in the second group. If G_{ij} is the greatest element both in its row and in its column, these two points will be identified as a candidate match. A threshold t_c is used to reject the unstable candidate matches with a low correlation score, which is set to be 0.75 in our experiments. We will present detailed discussion on t_c selection in Section 7.1.

5. Multilevel Matching Strategy

A multilevel matching strategy is used to establish the correspondences of feature points. Feature points are divided into 4 groups according to the pyramid level at which they are detected. The traditional matching strategy performs full group-to-group matching, which requires 16 group-to-group matching operations. We can speed up the matching procedure by removing the redundant computation. Only 7 group-to-group matching operations are required in our matching strategy, as shown in Figure 4.

The black dots labeled with numbers represent the feature point groups of different pyramid levels in the two images. Each line segment connecting two feature point groups denotes one group-to-group matching operation. The multilevel matching strategy reduces the computation cost in the matching procedure and makes our method more efficient.

For example, let $m_1, m_2, m_3,$ and m_4 be the numbers of features detected by MOCC for each pyramid level in the first image, and let n_1, n_2, n_3, n_4 be the numbers of features detected by MOCC for each pyramid level in the second image. The numbers of features detected by traditional matching method for the two images are $(m_1 + m_2 + m_3 + m_4)$ and $(n_1 + n_2 + n_3 + n_4)$, respectively. Then the number of feature comparisons for MOCC is $(m_1n_2 + m_1n_3 + m_1n_4 + m_2n_1 + m_3n_1 + m_4n_1 + m_1n_1)$, and the number of feature comparisons for traditional matching method is $(m_1 + m_2 + m_3 + m_4) \times (n_1 + n_2 + n_3 + n_4)$. It shows that using multilevel matching strategy reduces $(m_2 + m_3 + m_4) \times (n_2 + n_3 + n_4)$ feature comparisons. Since the number of the features detected in the first pyramid level is about 50% of the total feature number experimentally, the average performance improvement is 25%.

MOCC has an inherent parallel architecture. The feature detection in MOCC can be processed simultaneously in each scale level, and the 7 group-to-group matching procedures can also be processed at the same time. Therefore, potential speedup can be achieved using parallel computing hardware.

Figure 5 shows group-to-group matching results for image pair Laptop with a scale factor of 7. The first image (Figure 5(a)) is frame 1 of “Laptop” sequence, and the second one (Figure 5(b)) is the rescaled image of the frame

21 of the same sequence with a scale factor of 0.9 (the scale factor between frame 1 and 21 is 6.3).

Here we only present 4 group-to-group matching results due to space limitations. Other 3 group-to-group matches are matches between pyramid level 1 for the first image and pyramid levels 2, 3, and 4 for the second image. Apparently they will not give correct matches. These matching results in Figure 5 are the final results after false match rejection, and the coordinates have been mapped to the original image. Obviously, the first 3 group-to-group matching results give no correct matches. The fourth group-to-group matching result contains 16 matches, and all of them are correct.

6. Rejection of False Matches

For each group-to-group matching operation, we obtain an initial set of feature point matches. The initial set of feature point matches usually contains some false matches due to the inaccurate characterization of feature points or the improper matches established in the matching procedure. In the case of matching two uncalibrated images, the epipolar constraint can be used to reject the false matches [32]. In our experiments, the epipolar constraint is imposed based on the robust estimator RANSAC [33]. The feature point matches that are not consistent with the estimated epipolar geometry are identified as false matches and rejected.

Suppose that F is the fundamental matrix. Point p (x, y) can be represented as $\tilde{p} = [x, y, 1]^T$. For a feature point match (p, q) , the epipolar line of point p is defined as $l_p = F\tilde{p}$. If the match is perfect, point q should lie on the epipolar line l_p exactly. The distance d_q of point q to the epipolar line l_p is calculated by

$$d_q = \frac{|\tilde{q}^T F \tilde{p}|}{\sqrt{(F\tilde{p})_1^2 + (F\tilde{p})_2^2}}, \quad (6)$$

where $(F\tilde{p})_i$ is the i th component of vector $F\tilde{p}$. The distance d_p of point p to the epipolar line l_q is calculated similarly. Then a threshold t_e can be used to find the bad matches. A feature point match will be identified as a false match if $\max(d_p, d_q) > t_e$. False matches are removed from the initial set of feature point matches.

After rejecting the false matches by using epipolar constraint, we obtain the refined matching result for each group-to-group matching. The matching result that has the largest number of feature point matches will be selected as the matching result between the two images.

We find that there still exist a few false matches in the matching result after applying epipolar constraint. The feature points of these false matches happen to locate around the epipolar lines. Therefore, they cannot be identified only using epipolar constraint. A simple constraint is employed to further improve the quality of the selected matching result. For all good matches, the difference between the dominant orientations of the two feature points should be almost equal. Suppose the dominant orientation of one feature point is θ . After rotating the image by φ counter-clockwise, the dominant orientation of this feature point should be $\theta + \varphi$.

Since the feature points are detected separately in two images, there always exist location errors and the assignment of dominant orientation cannot be so accurate. Ideal in-plane rotation is also hardly found in practice. So even for good matches, the differences between the dominant orientations of the two feature points could not be of the same value. Considering the fact that the number of the false matches is usually very small, we use the following process to identify these false matches.

We first calculate the average of the differences between the dominant orientations for all feature point matches. Suppose $\bar{\theta}$ is the average of the differences and θ_i is the difference between the dominant orientations of the i th feature point match. t_θ is a threshold. If $|\theta_i - \bar{\theta}| > t_\theta$, the i th feature point match will be identified as a false match. The thresholds t_e and t_θ are empirically set to be 1.0 and 40 degrees, respectively.

7. Experimental Results

In this section, we will demonstrate more experimental results on real images. The images used in our experiments are mainly from the public image database in INRIA [34]. We will first discuss the parameter selection and the comparison of different downsampling methods. Then our proposed method is evaluated with respect to the matching performance under different imaging conditions such as scale changes, image rotation, illumination changes, and noise corruption. We also provide the comparison of matching performance and matching speed with other state-of-the-art matching techniques.

7.1. Parameter Discussions. We can see that the only parameter needed to be determined in the definition of the similarity measures (see (3) and (4)) is the size of correlation window. The experimental results on different correlation window radiuses are shown in Figure 6.

These results are broadly similar for other images in the INRIA image database under different scale changes. As this graph shows, the maximum number of correct matches is obtained when the radius of correlation window is around 6. The radius of correlation window also affects the detection and matching speed significantly. For example, the whole detection and matching time increases nearly 50% when w changes from 5 to 6. The value of w is set to 5 in all other experiments through this paper in consideration of speed. Note that smaller values of w such as 3 and 4 will cause unstable results when scale changes are large.

Figure 7 shows the evaluation for different thresholds. The initial set of feature point matches between the two groups can be established by selecting all such elements in G .

Since the nearest neighbor-based matchings are performed in both the row and the column of G , decreasing in t_c has less impact on the number of correct matches when t_c is below 0.7. Although the percentage of correct matches keeps increasing as t_c increases, the number of correct matches decreases significantly when t_c is above 0.8.

TABLE 1: Matching results for Figures 7–10.

Image pair	Feature points	Initial matches	Final matches	Average distance	False matches
Residence	213/713	115	75	0.641	1
Boat	532/1500	140	64	0.669	0
East_south	315/1500	124	100	0.759	0
Bark	220/1500	103	62	0.638	0

7.2. Comparison of Different Downsampling Methods. We use Gaussian smoothing combined with bilinear interpolation for the purpose of generating the image pyramid. More complicated approaches such as Bicubic, Lanczos, and Mitchell have also been considered and tested. The performance comparison is shown in Figure 8. The value of the x -axis is the scale changes between matching images, and the y -axis stands for the number of correct matches. The window size of Lanczos algorithm is 3, and for Bicubic method we use Bicubic spline interpolation.

From the results we can see that, when the scale changes are small, our method and Bicubic spline interpolation out-perform the rest of the methods. As the scale changes increase, the performance differences of the four methods decrease. Bicubic spline interpolation gives slightly better matching results than other methods. Therefore, if the computational cost of Bicubic spline interpolation is tolerable, one can use Bicubic spline interpolation to acquire more correct matches under large scale changes. By using Bicubic spline interpolation method, the feature detection time is about 1.5 times longer than that of our proposed downsampling method takes.

As we can observe from above discussions, the downsampling schemes will affect the matching performance. It is necessary to consider more downsizing methods to enhance the matching performance. Therefore, in future work, we will investigate different downsizing methods in order to further improve the proposed method.

7.3. Performance Evaluation under Significant Camera Motions. Figures 9–12 show the final matching results for four image pairs with significant camera motions (translation, rotation, and scaling). The details of the results are illustrated in Table 1.

The second column of Table 1 gives the number of feature points in the corresponding pyramid levels. For example, the final matching result for image pair Bark falls into the group-to-group matching between the third pyramid level of the first image and the first pyramid level of the second image. There are 220 and 1500 feature points detected in the two pyramid levels, respectively. The third and fourth columns provide the number of initial matches and final matches obtained by our method. The fifth column gives the average distance to epipolar lines. The fundamental matrix is recalculated after performing the coordinate transformation in final matches, and the average distance is computed from the transformed coordinates. The last column presents the number of false matches that exist in final matches.

False matches are determined using independently estimated homography matrices (ground truth). These homography matrices are included in the dataset [34]. The way of using the ground truth homographies to evaluate the quality of matches is also adopted in other research work [35, 36].

Figure 9 shows the matching result for image pair Residence with significant scale changes and translation. There also exist self-similarity structures in the two images. Figures 10 and 11 show the matching results for image pairs Boat and East_south with large rotation and scale changes. We also test our method on the textured scene. Figure 12 shows the matching result for the image pair Bark of a textured scene with large rotation and scale changes. Note that in each pair of images only several matched points are connected by lines for a clear vision.

Figure 13 demonstrates matching results under viewpoint changes and other examples under illumination changes, partial occlusion, and additive Gaussian noise. The experiments on matching image pairs with viewpoint changes show that our method can also tolerate weak affine distortion (up to 30 degrees of viewpoint change).

7.4. Performance Comparison with Other Matching Techniques. In this section, the proposed method MOCC is compared with other 5 state-of-the-art matching techniques with respect to ratio between correct matches and total features, precision, and false match rate. The results on a standard evaluation set are presented. The first two classic methods are SIFT [10] and SURF [25]. We use the latest binary code available on the authors' website for evaluation [37, 38]. We also compare MOCC with 3 state-of-the-art interest point detectors: FAST [39], Harris-Laplace [26], and Hessian-Laplace [26], which are interest point detectors with comparable performance to the above two methods. For these detection methods, we use them to extract interest points and then compute SIFT descriptors for matching purpose. We use SIFT here because SIFT has been proven to provide better matching results with these interest point detectors [35]. The matching strategy used in our evaluations for all matching techniques is nearest neighbor distance ratio method, which is implemented in [37].

Scale invariant feature transform (SIFT) proposed by Lowe [10] is one of the state-of-the-art matching techniques. It combines a scale invariant region detector and a descriptor based on the gradient distribution of the regions. The region detector convolves the image with a Difference of Gaussian (DoG) kernels at different scales and selects local maxima in both space and scale. A 3D histogram of gradient locations and orientations is utilized to represent the descriptor. Since SIFT uses DoG as the approximation for Laplacian of Gaussian (LoG) and DoG is much faster to compute, SIFT achieves faster detecting speed than other scale invariant feature detectors like Harris-Laplace and Hessian-Laplace while keeping comparable matching results [35, 36]. Bay et al. [25] propose speedup robust features (SURF) partially inspired by SIFT, which is another popular scale- and rotation-invariant detector and descriptor. SURF accelerates the feature detection procedure by utilizing integral images

for image convolutions and by using a Hessian matrix-based measure for the detector. Its descriptor is built on the distribution of first-order Haar wavelet responses in x and y directions. The experimental results on SURF show that SURF approximates or outperforms previously proposed matching schemes with much faster speed in feature detection, descriptor calculation, and matching process [40].

The evaluation image set consists of 5 groups of 10 test images, which are chosen from the public image database in INRIA. The test images contain rotation and scale changes. For each group, we match the first image (reference image) with the rest of the images in the same group, which are indexed from 1 to 9. The match with the same index in the different group has the approximate scale changes. It makes the average value response to the average performance under such scale changes. With the increase of the match index, the scale changes between images also increases from 1.9 to 5.8. Note that we manually generate a few images in some groups as the original image sets lack images at certain scale levels. The method of synthesizing is simply rescaling the images that have closest scale changes in the original database.

Figures 14–16 show the evaluation results. Figure 14 shows the ratio of correct matches/total features with respect to different scale changes. Here the numerator means the number of correct matches in the final results. The denominator is the sum of the numbers of features that are detected in both images. This measure is similar to the measure of the matching score used in [35]. It evaluates the capability of generating more correct matches with less detected features.

MOCC gets slightly better performance when the scale change is above 3.5. The performance of FAST method is lower than other methods when scale changes increase. This is because FAST is not a scale invariant or multiscale feature detection method.

Figure 15 shows the precision comparison of the three methods. The precision is defined as the ratio of correct matches to total matches [36]. The average precision of MOCC is higher than SURF and FAST but lower than SIFT.

Figure 16 shows the false match rate comparison of the three methods. The false match rate is defined as the ratio of false matches to final matches. This measure indicates how many false matches can survive from the outlier detection using epipolar geometry. This time SIFT still gets the best result among the three methods. MOCC has a higher false match rate than SIFT but lower than SURF and FAST. From these comparisons we should say that the methods are competitive with each other according to the above performance measures. However, the design aim of MOCC is to provide a fast and effective matching method. Therefore, we will provide some results on speed tests with these methods.

7.5. Speed Comparison with Other Matching Techniques. The speed test is performed on a notebook running Windows XP Professional (Intel Pentium M 750, 1.86 GHz, 512 M memory). The feature detection experiment is performed on images from INRIA database with the same resolution

TABLE 2: Speed comparison.

Method	Detection time (ms)	Matching time (ms)
MOCC	468	2020
SIFT	4350	12386
SURF	1800	5033
FAST-ER	317	4722
Harris-L	11984	28673
Hessian-L	5275	15289

$850 * 680$, and the detected features for the three methods are about 2400. The feature detection time includes feature detecting time, descriptor computing time, and time of writing to files. The matching experiment is also performed on images with the same resolution $850 * 680$, and the total detected features in two images for the three methods are about 5400. The image matching time includes feature detecting time, descriptor computing time, and matching time. Note that we use the same set of parameters throughout the paper for MOCC. The timings given in Table 2 are evaluated by repeating the experiments 50 times and calculating the average.

Both the detection and matching speeds of MOCC are faster than SIFT and SURF. The detection speed of MOCC is 9.3 times faster than SIFT and 3.85 times faster than SURF. The matching speed of MOCC is more than 6 times faster than SIFT and 2.5 times faster than SURF. Note that, from the results in Table 2, FAST-ER is the fastest feature detector in this comparison. However, it has limitations when handling large scale changes as it does not generate scale space representation when detecting features.

We believe that the following 4 aspects contribute to the high speed of MOCC. First, we introduce a fast multiscale representation for generating image pyramid. Second, the adopted scale space representation allows using the same scale parameter in Harris algorithms. Therefore, two 1D Gaussian convolutions in the x -direction and y -direction could be applied to perform the Gaussian smoothing, which greatly reduces the time needed for convolution operations. Third, we compute normalized cross-correlation via feature arrays in order to avoid repeated generation processes of correlation windows. Fourth, the multilevel matching strategy notably increases the matching speed.

7.6. Robustness against Imaging Variations. In order to further evaluate our proposed method, we perform matching experiments on images having different imaging variations. The following experiments use the Oxford image database [41], which is used as a standard test set for image matching tasks [35, 36, 40]. Note that we use images from INRIA database to perform the experiments under different scale changes in Section 7.4, because the test set of images with scale changes in Oxford database is a subset of that in INRIA database.

Figures 17–19 illustrate the comparison of matching results on the images with illumination changes, Gaussian noise corruptions, and image blurring. The measure of

matching score [35] is used for the evaluation. Here the matching score is computed as the ratio between the number of correct matches and the smaller number of detected features in the pair of images. Note that for MOCC we use the smaller number of detected features in the matched level to compute the matching score. In Figures 17 and 19, the x -axis is the image number in Leuven and Bikes sequences, respectively. In Figure 18, the x -axis is the standard deviation of additive Gaussian noise. We take the first image in Graffiti sequence for noise corruption test.

From the results we can see that the performance of MOCC drops fast when the image noise increases. And its performance under image blur is not as good as others. This is partially because we use NCC as feature description, while it is not as powerful as the descriptors used by SIFT and SURF, which are based on gradient distributions. Another point that needs to be addressed is that NCC should be robust against local illumination changes. But the inaccurate localization and the approximation of gray values in local patches caused by downsampling will decrease its performance.

In the proposed method, we focus on the speed of the whole system; therefore, we make a compromise between accuracy and efficiency. The great speedup achieved by MOCC relies on the fast algorithms on scale representation, feature point localization, simplified feature descriptor, and group-based matching strategy. These fast algorithms also lead to inaccurate localization of feature points and less powerful feature description and will decrease the matching performance under different imaging variations. The other fast method like SURF also raises similar problems as shown in the above figures. As for the practical use, our method could generate enough number of correct point matches even in the current parameter settings. In future work, we will improve the method by using more powerful descriptors to enhance the performance under different imaging variations.

8. Conclusions

This paper presents a new method named MOCC (Multiscale Oriented Corner Correlation) for matching two uncalibrated images under large scale changes. The method is based on matching multiscale feature points using rotation invariant normalized cross-correlation. Experimental results on real images demonstrate that our method is effective and efficient for matching two uncalibrated images with large rotation and significant scale changes. The new method is able to match image pairs with scale changes up to a factor of 7. Although there have been a great many studies in the image matching field, to the best of our knowledge, none of the existing correlation-based approaches allows to deal with such large changes in scale. Additional contribution is the fast speed of MOCC. It is significantly faster than the state-of-the-art matching schemes, and MOCC has potential ability of speedup due to its inherent parallel architecture. Future work will focus on further performance improvements in order to make MOCC utilizable on mobile platforms.

Acknowledgments

This paper was supported in part by the National High-Tech Research and Development Program of China (863 Program) under Grant 2006AA01Z117, the National Basic Research Program of China (973 Program) under Grant 2009CB320906, the National Natural Science Foundation of China under Grants 60773136 and 60833006, and the Natural Science Foundation of Beijing under Grant 4092042.

References

- [1] L. G. Brown, "Survey of image registration techniques," *ACM Computing Surveys*, vol. 24, no. 4, pp. 325–376, 1992.
- [2] C. Heipke, "Overview of image matching techniques," *OEEPE Official Publication*, vol. 33, pp. 173–189, 1996.
- [3] B. Zitová and J. Flusser, "Image registration methods: a survey," *Image and Vision Computing*, vol. 21, no. 11, pp. 977–1000, 2003.
- [4] W. Förstner, "A feature-based correspondence algorithm for image matching," *International Archives of Photogrammetry and Remote Sensing*, vol. 26, no. 3, pp. 150–166, 1986.
- [5] Z. Zhang, R. Deriche, O. Faugeras, and Q.-T. Luong, "A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry," *Artificial Intelligence*, vol. 78, no. 1-2, pp. 87–119, 1995.
- [6] H. Hanaizumi, "Automated method for registration of satellite remote sensing images," in *Proceedings of the 13th Annual International Geoscience and Remote Sensing Symposium*, pp. 1348–1350, August 1993.
- [7] R. Berthilsson, "Affine correlation," in *Proceedings of the International Conference on Pattern Recognition*, pp. 1458–1461, 1998.
- [8] A. P. Witkin, "Scale-space filtering," in *Proceedings of International Joint Conference on Artificial Intelligence*, pp. 1019–1023, 1983.
- [9] T. Lindeberg, "Feature detection with automatic scale selection," *International Journal of Computer Vision*, vol. 30, no. 2, pp. 79–116, 1998.
- [10] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.
- [11] K. Mikolajczyk and C. Schmid, "Indexing based on scale invariant interest points," in *Proceedings of the 8th International Conference on Computer Vision*, pp. 525–531, July 2001.
- [12] E. H. Adelson, C. H. Anderson, J. R. Bergen, P. J. Burt, and J. M. Ogden, "Pyramid methods in image processing," *RCA Engineer*, vol. 29, no. 6, pp. 33–41, 1984.
- [13] P. J. Burt and E. H. Adelson, "The laplacian pyramid as a compact image code," *IEEE Transactions on Communications*, vol. 31, no. 4, pp. 532–540, 1983.
- [14] J. L. Crowley and A. C. Parker, "A representation for shape based on peaks and ridges in the difference of low pass transform," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 6, no. 2, pp. 156–170, 1984.
- [15] N. Sebe, Q. Tian, E. Loupias, M. S. Lew, and T. S. Huang, "Evaluation of salient point techniques," *Image and Vision Computing*, vol. 21, no. 13-14, pp. 1087–1095, 2003.
- [16] H. Moravec, "Towards automatic visual obstacle avoidance," in *Proceedings of the International Joint Conference on Artificial Intelligence*, p. 584, 1977.

- [17] C. Harris and M. Stephens, "A combined corner and edge detector," in *Proceedings of the 4th Alvey Vision Conference*, pp. 147–151, 1988.
- [18] C. Schmid, R. Mohr, and C. Bauckhage, "Evaluation of interest point detectors," *International Journal of Computer Vision*, vol. 37, no. 2, pp. 151–172, 2000.
- [19] C. Schmid and R. Mohr, "Local grayvalue invariants for image retrieval," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 5, pp. 530–535, 1997.
- [20] A. Baumberg, "Reliable feature matching across widely separated views," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR '00)*, pp. 774–781, June 2000.
- [21] Y. Dufournaud, C. Schmid, and R. Horaud, "Image matching with scale adjustment," *Computer Vision and Image Understanding*, vol. 93, no. 2, pp. 175–194, 2004.
- [22] F. Zhao, Q. M. Huang, and W. Gao, "Image matching by multiscale oriented corner correlation," in *Proceedings of the Asian Conference on Computer Vision*, pp. 928–937, 2006.
- [23] T. Kadir and M. Brady, "Saliency, scale and image description," *International Journal of Computer Vision*, vol. 45, no. 2, pp. 83–105, 2001.
- [24] F. Jurie and C. Schmid, "Scale-invariant shape features for recognition of object categories," in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '04)*, pp. 90–96, July 2004.
- [25] H. Bay, T. Tuytelaars, and L. V. Gool, "SURF: speeded up robust features," in *Proceedings of the European Conference on Computer Vision*, pp. 404–417, 2006.
- [26] K. Mikolajczyk and C. Schmid, "Scale & affine invariant interest point detectors," *International Journal of Computer Vision*, vol. 60, no. 1, pp. 63–86, 2004.
- [27] T. Tuytelaars and L. Van Gool, "Matching widely separated views based on affine invariant regions," *International Journal of Computer Vision*, vol. 59, no. 1, pp. 61–85, 2004.
- [28] J. Matas, O. Chum, U. Martin, and T. Pajdla, "Robust wide baseline stereo from maximally stable extremal regions," in *Proceedings of the British Machine Vision Conference*, pp. 384–393, 2002.
- [29] T. Kadir, A. Zisserman, and M. Brady, "An affine invariant salient region detector," in *Proceedings of the European Conference on Computer Vision*, pp. 228–241, 2004.
- [30] M. Brown, R. Szeliski, and S. Winder, "Multi-image matching using multi-scale oriented patches," in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '05)*, pp. 510–517, June 2005.
- [31] I. T. Young and L. J. van Vliet, "Recursive implementation of the Gaussian filter," *Signal Processing*, vol. 44, no. 2, pp. 139–151, 1995.
- [32] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, Cambridge University Press, Cambridge, UK, 2000.
- [33] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.
- [34] October 2007, <http://lear.inrialpes.fr/people/Mikolajczyk/Database/index.html>.
- [35] K. Mikolajczyk, T. Tuytelaars, C. Schmid et al., "A comparison of affine region detectors," *International Journal of Computer Vision*, vol. 65, no. 1-2, pp. 43–72, 2005.
- [36] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 10, pp. 1615–1630, 2005.
- [37] "SIFT demo program (Version 4, July 2005)," accessed in October 2007, <http://www.cs.ubc.ca/~lowe/keypoints/>.
- [38] "SURF version 1.0.9," October 2007, <http://www.vision.ee.ethz.ch/~surf/download.html>.
- [39] E. Rosten, R. Porter, and T. Drummond, "Faster and better: a machine learning approach to corner detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 1, pp. 105–119, 2010.
- [40] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (SURF)," *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346–359, 2008.
- [41] June 2010, <http://www.robots.ox.ac.uk/~vgg/research/affine/>.