

# Local Descriptor by Zernike Moments for Real-time Keypoint Matching

Sun-Kyoo Hwang\*, Mark Billingham\*\* and Whoi-Yul Kim\*

*Image Engineering Lab., Hanyang University, Seoul, Korea\**

*The HIT Lab NZ, University of Canterbury, Christchurch, New Zealand\*\**

*sunkyoo.hwang@gmail.com, mark.billinghurst@hitlabnz.org, wykim@hanyang.ac.kr*

## Abstract

*This paper presents a real-time keypoint matching algorithm using a local descriptor derived by Zernike moments. From an input image, we find a set of keypoints by using an existing corner detection algorithm. At each keypoint we extract a fixed size image patch and compute a local descriptor derived by Zernike moments. The proposed local descriptor is invariant to rotation and illumination changes. In order to speed up the computation of Zernike moments, we compute the Zernike basis functions in advance and store them in a set of lookup tables. The matching is performed with an Approximate Nearest Neighbor (ANN) method and refined by a RANSAC algorithm. In the experiments we confirmed that videos of frame size 320×240 with the scale, rotation, illumination and even 3D viewpoint changes are processed at 25~30Hz using the proposed method. Unlike existing keypoint matching algorithms, our approach also works in real-time for registering a reference image.*

## 1. Introduction

Keypoints, often called as interest points, are points that contain distinctive information so that they can be found repeatedly in images when they undergo various transformations such as translation, scaling, and rotation. Finding and matching keypoints in a pair of images has been widely used in many computer vision problems for tasks such as object recognition [8], tracking [10], and 3D pose estimation problems [9][13].

Keypoint matching usually involves three steps: (1) Keypoint detection, (2) Keypoint description, and (3) Keypoint matching. In keypoint detection, it is important to find the same location of keypoints in an image even though the image has been geometrically deformed. Keypoint description means the computation of a feature vector that describes the characteristics of an image patch around the keypoint.

Keypoint matching is used to find the corresponding points in two images by using the keypoint descriptors.

Lepetit and Fua treat keypoint matching as a classification problem and have proposed a matching algorithm robust to camera pose [9]. They synthesized various views of keypoints and used randomized trees as statistical classification tools. However, with their method it takes more than 10 minutes to train a reference image because rotated images at every possible angle have to be synthesized. Tran and Marchand proposed a fast keypoint matching algorithm for visual servoing applications [13]. They computed the keypoint orientation using a histogram of gradient orientation. However, they used a local descriptor based on Principal Component Analysis (PCA) so that their method also needs a training step and takes time for the registration of a reference image.

In this paper we present a real-time keypoint matching algorithm with a local descriptor defined by Zernike moments. Zernike moments are rotation invariant and robust to noise, so the local descriptor also has these properties. We extract an image patch with a fixed size at every keypoint and compute a feature vector for the patch. For a real-time implementation, we compute the full set of Zernike basis functions with the fixed size in advance and store that in a set of look-up tables. Then, Zernike moments are computed by projecting image patches onto the pre-computed basis functions. Therefore Zernike moments at every keypoint can be computed very quickly and the whole process works in real-time. Unlike existing keypoint matching algorithms, the proposed method registers a reference image in real-time.

In the next Section we describe the keypoint detection method. In Section 3, we describe the local descriptor derived by Zernike moments in detail. Section 4 describes the keypoint matching techniques. Experimental results are given in Section 5 and the conclusions follow in Section 6.

## 2. Keypoint Detection

Over the last several decades many keypoint detectors have been proposed including Harris [4] and Laplacian of Gaussian (LoG) [10]. These methods are often combined with scale-space techniques to find scale invariant keypoints [10][11]. However, they are computationally too expensive to be used in real-time applications.

Recently a simple and fast corner detection algorithm has been proposed [12][14]. The method scans an image and checks the 16 points around a pixel as shown by the black squares in Figure 1. A pixel  $p$  is classified as a corner point if there are a set of  $n$  contiguous pixels in the surrounding 16 pixels which are brighter, or darker, than the intensity of  $p$ . This corner detector requires only 1~2ms for finding all corners in a 320×240 pixel image on a normal PC. We use the algorithm as a keypoint detector in our system.

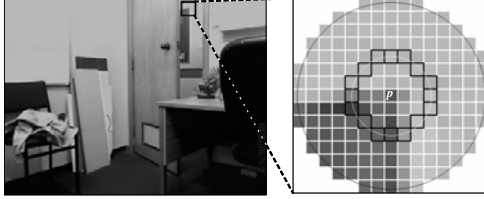


Figure 1. Keypoint detection in an image

## 3. Local Descriptor by Zernike Moments

### 3.1. Zernike moments

Zernike moments have been widely used in pattern recognition systems because of their ability to effectively represent the properties of an image [7]. They are computed by projecting an input image onto a set of Zernike basis functions. Zernike moments of order  $n$  with repetition  $m$  are defined as

$$Z_{nm} = \frac{n+1}{\pi} \iint_{\rho \leq 1} f(\rho, \theta) V_{nm}^*(\rho, \theta) \rho d\rho d\theta, \quad (1)$$

where  $f(\rho, \theta)$  is the input image function,  $V_{nm}(\rho, \theta)$  is the complex Zernike basis function, and \* denotes the complex conjugate.

Zernike basis functions are defined as

$$V_{nm}(\rho, \theta) = R_{nm}(\rho) \exp(jm\theta), \quad |\rho| \leq 1, \quad (2)$$

where

$$R_{nm}(\rho) = \sum_{s=0}^{(n-|m|)/2} (-1)^s \frac{(n-s)!}{s! \left(\frac{n+|m|}{2} - s\right)! \left(\frac{n-|m|}{2} - s\right)!} \rho^{n-2s}. \quad (3)$$

In the equation (3), the order  $n$  is a non-negative integer and the repetition  $m$  is an integer satisfying  $n-|m| = (\text{even})$  and  $|m| \leq n$ .  $R_{nm}(\rho)$ , called a Zernike radial polynomial, is orthogonal with different orders. Therefore Zernike moments can represent the properties of an image with no redundancy or overlap of information between the moments.

To compute Zernike moments from a digital image, the equation (1) is changed to

$$Z_{nm} = \frac{n+1}{\lambda_N} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x, y) V_{nm}^*(x, y), \quad (4)$$

where  $\lambda_N$  is the number of pixels located in a unit circle by a mapping [6].

### 3.2. Local descriptors

Recently Zernike moments were successfully used as local descriptors for model-based object recognition [8]. The local descriptor is invariant to rotation transform and robust to noise and illumination changes.

For a rotated image,  $f(\rho, \theta + \alpha)$ , the Zernike moments are defined as

$$Z_{nm}^r = Z_{nm} \exp(-jm\alpha). \quad (5)$$

This means that the rotation of an input image affects only the phase of the Zernike moments and the absolute value of the Zernike moments is invariant,

$$|Z_{nm}^r| = |Z_{nm}|. \quad (6)$$

To cope with the illumination changes, we normalized the absolute value of Zernike moments with  $|Z_{00}|$ . Zernike moments with order 0 and repetition 0,  $Z_{00}$ , represent the average intensity of an input image. We assumed that the intensity of an input image is

$$f'(x, y) = \gamma \cdot f(x, y). \quad (7)$$

Then, dividing the absolute values of Zernike moments with  $|Z_{00}|$ , the feature vector stays unchanged,

$$f_{nm} = \frac{|Z_{nm}|}{|Z_{00}|}. \quad (8)$$

Therefore the feature vector for the  $i$ -th keypoint is defined as

$$\mathbf{f}^i = \{f_{nm}^i \mid 1 \leq n \leq N, n-m = \text{even}, 0 \leq m \leq n\}, \quad (9)$$

where  $N$  is the maximum order to be computed. Note that the order  $n$  starts from 1. In our experiments, we set  $N = 7$  and the length of a feature vector to be 19.

### 3.3. Fast computation

For a real-time implementation, fast computation of Zernike moments is essential. When Zernike moments are extracted from an image, the computation of 2D complex Zernike basis functions usually takes the most time because there are many factorial and sinusoidal operations.

In our method, we extract an image patch with a fixed size around every keypoint and compute a feature vector to represent the image patch. As the size of the image patches is fixed, a set of Zernike basis functions of that size is computed in advance and can be used in the computation of Zernike moments for the image patches. In our implementation, we set the image patch size to  $15 \times 15$ , where the size was empirically determined. Since Zernike basis functions are defined within a unit circle, only 177 pixels in an image patch are used for the computation as shown in Figure 2. So, computing a feature vector for a keypoint requires 177 multiplications and 176 additions. The number of multiplication is effectively reduced by using the symmetrical properties of Zernike basis functions [6].

## 4. Keypoint Matching

We find the candidates for initial correspondences of keypoints using an Approximate Nearest Neighbor (ANN) algorithm [1][14]. ANN is an algorithm to find the approximate nearest neighbor in a set of points. Using a set of data points, ANN constructs a  $kd$ -tree, where  $k$  denotes the dimension of the data points and, in this paper,  $k$  is 19. If a query point  $\mathbf{q}$  is given then ANN finds an approximate nearest point  $\mathbf{p}$  in the data set. The distance between two points is generally defined as

$$\text{dist}(\mathbf{p}, \mathbf{q}) = \left( \sum_{i=0}^k (p_i - q_i)^2 \right)^{1/2}. \quad (10)$$

The advantage of the ANN method is that it produces a fast response for a given query. In addition, since the construction of the  $kd$ -tree has a complexity of  $O(K \log K)$ , where  $K$  is the number of feature

vectors, the registration of a reference image can be performed very quickly.

However, since the initial correspondences by ANN usually contain a number of incorrect matchings because of noise or distortion of the input images. So a process to refine the initial correspondence has to follow. In this paper, a RANSAC based algorithm is performed to remove the undesirable correspondences. We assumed that the keypoints are on a single plane for simplicity. In this case, a planar homography can be computed with four correspondences and it is used in a RANSAC algorithm as a model [3][5]. In our method, we used the LO RANSAC algorithm [2] because it is more robust and faster than the traditional RANSAC algorithm.

## 5. Experiments

To evaluate the performance of our method, we performed experiments with different sets of images and measured the running time. All the experiments were carried out with an implementation developed using Visual C++ 2005 on a 2.8GHz PC.

In our first experiment we synthesized query images from a reference image by transforming scale, rotation, translation, and brightness. We constructed a  $kd$ -tree with a reference image only once and found the corresponding points from input images. Figure 2 shows the matching results for each input image. Figure 2(a) shows the initial correspondences processed by the ANN algorithm, where the input image was scaled 110% from a reference image. As shown in the figure, the initial correspondences contain many outliers as well as inliers. The RANSAC algorithm was then performed to remove the outliers and generated the result shown in Figure 2(b), where only inlier correspondences remained. Figure 2(c) shows the matching result for an image rotated  $160^\circ$  from the image in Figure 2(b). The proposed method found the matching points well, even though the rotated image became 70% darker as shown in Figure 2(d).

The elapsed times of the processes for the first experiment are listed in Table 1. It took 24ms for registering a reference image. For finding corresponding points from input images, it took less than 40ms. The computation of the local descriptor takes the most of time in proportion with the number of keypoints. For Figure 2(d), the matching procedure took only 20ms because the number of keypoints found was reduced.

In our second experiment we acquired images of a magazine from a webcam to introduce changes in 3D camera pose. The matching results with the test images

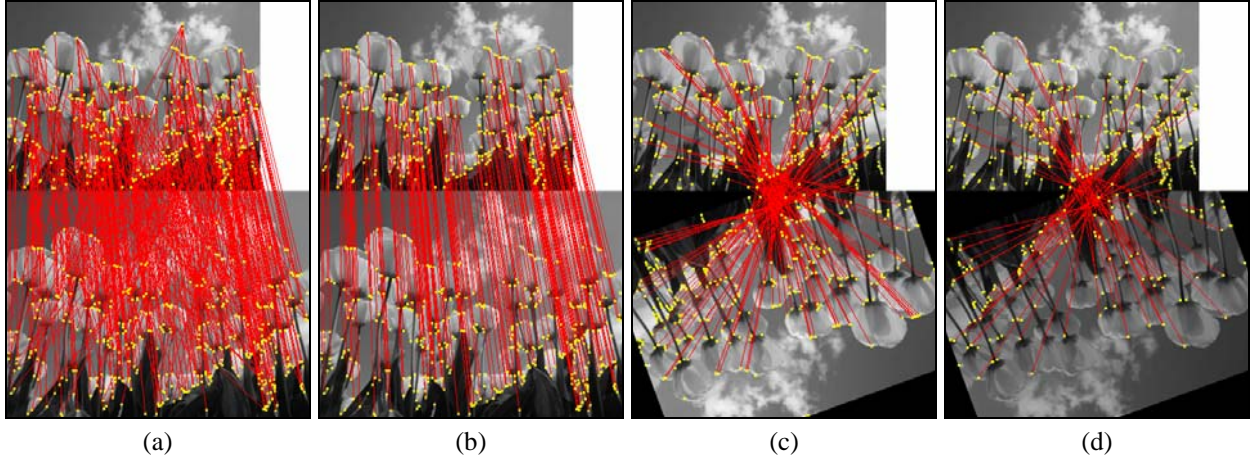


Figure 2. Keypoint matching for synthesized images

Table 1. Computation time for each process for Figure 2

Operations	Times (ms)			
	Reference image	Figure 3(b)	Figure 3(c)	Figure 3(d)
Keypoint detection	1			
Local descriptor (# of keypoints)	22 (388)			
Kd-tree construction	1			
Keypoint detection		2	2	1
Local descriptor (# of keypoints)		22 (387)	20 (350)	10 (146)
Initial correspondence		4	4	2
Outlier removal		10	9	7

are shown in Figure 3 and the elapsed time for each process is tabulated in Table 2. We confirmed that the matching results are reasonable and the elapsed time is very low because the number of keypoints is less than the first experiment. In this experiment, the number of the final correspondences is small but it is enough to

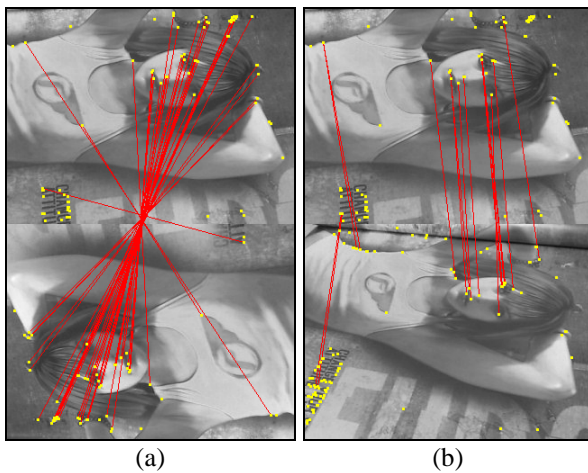


Figure 3. Keypoint matching for planar images with a different camera pose

compute the 3D camera pose by a conventional pose estimation method.

In the last experiment we tested our method with actual 3D views. From a image sequence by a webcam, we registered a frame as a reference image. The reference image contains 3D objects such as chair and bookshelf. The proposed keypoint matching was performed for every further input frame. We confirmed that the proposed keypoint matching worked at 25~30 Hz and the results are displayed in Figure 4.

## 6. Conclusions

In this paper, we have presented a real-time keypoint matching method with a moment-based local descriptor. The local descriptor is defined by Zernike moments so that they are robust to rotation and noise. In addition, the descriptor is normalized to cope with the change of illumination. The matching of keypoints are then performed with an ANN method and followed by a RANSAC algorithm for refinement. Our experimental results show that our method works in real-time on a normal PC. The registration of a reference image works in real-time, which is a significant improvement compared to existing keypoint matching methods.

**Table 2. Computation time for each process for Figure 3**

Operations	Times (ms)		
	Reference image	Figure 4(a)	Figure 4(b)
Keypoint detection	1		
Local descriptor (# of keypoints)	5 (75)		
Kd-tree construction	0		
Keypoint detection		0	1
Local descriptor (# of keypoints)		3 (52)	7 (101)
Initial correspondence		0	1
Outlier removal		1	6



**Figure 4. Keypoint matching from a 3D scene**

In our experiments we confirmed that the slight change of the scale of an input image does not affect the performance of the proposed method. However, large changes of the scale will be critical to this method. Therefore, in the future work, we will consider registration of multi-scale images to cope with large changes of scale. We will also explore applications of this technique in domains such as camera pose tracking for Augmented Reality tracking, or other areas.

## 7. Acknowledgement

*This work was supported by the Korea Research Foundation Grant funded by the Korean Government (MOEHRD) (KRF-2006-352-D00150)*

## 8. References

- [1] S. Arya, D. M. Mount, N. S. Netanyahu, R. Silverman, and A. Y. Wu. "An optimal algorithm for approximate nearest neighbor searching in fixed dimensions", *Journal of ACM*, vol. 45, no. 6, pp. 891-923, Nov. 1998.
- [2] O. Chum, J. Matas, and J. Kittler, "Locally optimized RANSAC," *Proceedings of the 25th DAGM Symposium*, pp. 236-243, Sep. 2003.
- [3] M. Fischler and R. 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, June 1981.
- [4] C.G. Harris and M.J. Stephens, "A combined corner and

edge detector," in *Fourth Alvey Vision Conference*, Manchester, 1988.

- [5] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, Cambridge University Press, 2000.
- [6] S. K. Hwang and W. Y. Kim, "A novel approach to the fast computation of Zernike moments," *Pattern Recognition*, vol. 39, no. 11, pp. 2065-2076, Nov. 2006.
- [7] A. Khotanadz, Y.H. Hong, "Invariant image recognition by Zernike moments," *IEEE Trans. on Pattern Anal. Mach. Intell.* vol. 12, no. 5, pp. 489-497, 1990.
- [8] S. Kim and I. S. Kweon, "Robust Model-based Scene Interpretation by Multilayered Context Information," *Computer Vision and Image Understanding*, vol. 105, no. 3, pp. 167-187, 2007.
- [9] V. Lepetit and P. Fua, "Keypoint recognition using randomized trees", *IEEE Trans. on Pattern Anal. Mach. Intell.*, vol. 28, no. 9, pp. 1465-1479, Sept. 2006.
- [10] D. 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 point," *IEEE International Conference on Computer Vision*, vol. 1, pp. 525-531, 2001.
- [12] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," *European Conference on Computer Vision*, pp. 430-443, May 2006.
- [13] T. T. H. Tran and E. Marchand, "Real-time key point matching: application to visual servoing", *IEEE International Conference on Robotics and Automation*, pp. 3787-3792, Roma, Italy, April 2007.
- [14] <http://www.cs.umd.edu/~mount/ANN/>