## Pruning and Weighting of Keypoints Using the HSI Color Space for Image Recognition

Shao-Hu Peng Dept of Electronic Engineering, Inha University, Korea pengshaohu@inha.ac.kr

Seok-Lyong Lee School of Industrial and Management Eng., Hankuk University of Foreign Studies sllee@hufs.ac.kr

### Abstract

The detection of the stable local image features is one of the most critical tasks for many object recognition algorithms. The Scale Invariant Feature Transform (SIFT) has been shown to be effective to the image matching or object recognition. However, the large number of features generated by the SIFT is a disadvantage for the real time application. In this paper, we present a novel approach to detect more important local features by finding the higher information keypoints (HIKs). An input color image is firstly decomposed into an intensity image, a hue image and a saturation image. Then we detect the HIKs in these color component images in terms of the keypoint positions. *Furthermore, a weight for each HIK is assigned according* to the position relationship of the keypoints to improve the matching accuracy. Experiments show that the proposed approach can achieve higher matching accuracy and reduce the matching time by using the HIKs and their weights.

## 1. Introduction

The object recognition is one of the most important tasks in the intelligent system. In recent years, one of the dominant approaches to the content based object recognition [8] [1] [9] is to find local interest regions and use the local characteristics. In the process of the image recognition, firstly, the effective detectors such as the Harris-Affine detector [11], the Hessian-Affine detector [12] and DoG [8] were presented to find the interest points in the image. Besides, the local descriptor such as GLOH

Deok-Hwan Kim\* Dept of Electronic Engineering, Inha University, Korea deokhwan@inha.ac.kr

Chin-Wan Chung Division of Computer Science, Dept of EECS, KAIST chungcw@kaist.ac.kr

[10], moment invariants [3] and differential invariants [5] were proposed to represent local characteristics of the image. The detection of interest points and the representation of the local feature must be invariant to the illumination, viewpoint, rotation, size and image noisy, etc.

The detection of interest points for the object recognition and the image retrieval was pioneered by Cordelia Schmid and Roger Mohr [15]. Various approaches to detect interest points for the image recognition have been developed in the past years [2] [14] [16]. Recently, the Scale Invariant Feature Transform (SIFT) algorithm [8] has been proved to be invariant to the image scale, rotation and affine transforms. So, it is widely applied for the image retrieval and the object recognition these days. However, an inherent problem in applying SIFT to the image recognition is the generation of a very large number of keypoints. The large number of features, ranging from hundreds to thousands keypoints for an image, degrades the matching performance. Therefore, it is necessary to reduce the number of keypoints in the SIFT features.

In this paper, we propose a novel approach to find out more important keypoints by using the color components. We call them the higher information keypoints (HIKs) since they have more important information than other less important keypoints. Its advantages are to reduce the number of the keypoints and improve the efficiency and effectiveness of image matching. By analyzing the HSI color space in the color image, we get the HIKs by using the intensity image, the hue image and the saturation image together. In the first step, the input color image is

<sup>\*</sup> The corresponding author.

<sup>253,</sup> Yonghyun-dong, nam-gu, Incheon, 402-751, Korea



Figure 1 HIK detection and weight assigning

transformed into the intensity image, hue image and saturation image, respectively. For each color component image, we locate the keypoints by using the DoG detector [8]. In the next step, we set the keypoints in the intensity image as the reference keypoints. If there exits any other keypoints in the hue image or in the saturation image whose positions are overlapped or close to the reference keypoints, we determine the reference keypoints as the higher information keypoints.

We keep the HIKs and prune the others so that the number of the keypoints can be reduced effectively. Furthermore, we assign a weight for each HIK in terms of the relationship between the reference keypoint and the other keypoint. The weight of the HIK is applied to the image matching so that the matching accuracy is improved. Fig.1 describes the main process of finding the HIKs and assigning their weights. Since we only use the HIKs and their weights for the image retrieval or the object recognition, the proposed approach can improve the performance and reduce the processing time efficiently.

The remainder of this paper is organized as follows: in section 2 we review the SIFT algorithm, and in section 3 we present our new approach. The experiments and the conclusion are in section 4 and section 5, respectively.

#### 2. The review of the SIFT algorithm

The SIFT algorithm [8] includes four stages: (1) scale-space extrema point detection; (2) keypoint localization; (3) orientation assignment; (4) keypoint descriptor.

First, the image is smoothed by the Gaussian function with different scales and down sampled to

build the Gaussian pyramid with different octaves. Then the Difference-of-Gaussian (DoG) pyramid is built by subtracting the adjacent smoothed images in the same octave. Extrema points are then determined by comparing the value of each point in the DoG image with its 26 neighbors around it. If its value is the maximum or the minimum, then it is determined as a keypoint.

Second, the keypoints are accurately located and those with low contrast or those close to the edge in the image are pruned for the stability.

Third, one or more consistent orientations are assigned for the keypoint to achieve the invariance of the image rotation.

Finally, the descriptor is calculated in the 16\*16 region around the keypoint. The region is separated into 16 blocks with the same size of 4\*4 pixels. For each block, the gradient magnitude and the orientation of each pixel are calculated to form the feature histogram with 8 bins and finally the SIFT descriptor is formed with 128 dimensions.

## 3. The proposed approach

Finding important regions in the image is the first step in the process of image matching or recognition [6]. As we have mentioned in section 2, the keypoint is the local extrema point that represents the local features with high informational content. It has been proved to be scale invariant. A color image can be represented by different color spaces such as RGB, HSI, CMY etc. The different color component describes different color information of the image. Thus, we can decompose the color image to form different color component images. For an image region, if we can find keypoints in different color component image, it denotes that the region includes higher local information. Thus, we can improve the matching performance by using these higher information regions.



Figure 2 detection of the higher information keypoint

# 3.1. The keypoint location and higher information keypoint detection

Firstly, the color image is decomposed into the hue, saturation and intensity images. Then the keypoints for each image are located by using the same approach as the SIFT. Therefore, for each image, we can form a set to represent the keypoints as follows:

$$SET_{I} = \{P_{i}(x_{i}, y_{i}) \mid 0 \le i \le I\}$$
(5)

$$SET_{h} = \{P_{h}(x_{h}, y_{h}) \mid 0 \le h \le H\}$$
(6)

$$SET_{s} = \{P_{s}(x_{s}, y_{s}) \mid 0 \le s \le S\}$$
(7)

where *I*, *H* and *S* denotes the number of the keypoints of the intensity image, the hue image and the saturation image, respectively.  $P_i(x_i, y_i)$ ,  $P_h(x_h, y_h)$ and  $P_s(x_s, y_s)$  denotes the keypoints in the intensity, hue and saturation image while  $(x_i, y_i)$ ,  $(x_h, y_h)$  and  $(x_s, y_s)$  denotes the position of the keypoint  $P_i(x_i, y_i)$ ,  $P_h(x_h, y_h)$  and  $P_s(x_s, y_s)$ , respectively.

The intensity component contains the information of the RGB components since it is the average of the RGB. Thus, we set the keypoints in the intensity image as the reference keypoint to find higher information keypoints. Fig.2 shows the detailed process of detecting the HIKs. Here, we give a definition of the higher information keypoint. Let the keypoints in the set  $SET_I$  be the reference keypoints. We overlap the intensity image and the hue image. Suppose that the keypoint  $P_h(x_h, y_h)$  in the hue image is the closest to the reference keypoint  $P_i(x_i, y_i)$ , then the position relationship between  $P_h(x_h, y_h)$  and  $P_i(x_i, y_i)$  can be denoted as follows:

$$d_{ih} = \sqrt{(x_i - x_h)^2 + (y_i - y_h)^2} \quad (8)$$

Similarity, we overlap the intensity image and the saturation image. If the keypoint  $P_s(x_s, y_s)$  in the saturation image is the closest to  $P_i(x_i, y_i)$ , then position relationship between  $P_s(x_s, y_s)$  and  $P_i(x_i, y_i)$  can be denoted as follows.

$$d_{is} = \sqrt{(x_i - x_s)^2 + (y_i - y_s)^2} \quad (9)$$

We set a circular window W with radius R. If  $\min(d_{ih}, d_{is}) \leq R$ , we determine that the reference keypoint is a Higher Information Keypoint (HIK).

Finally, we can get a set of the HIKs.

$$P_o = \{ p_o(x_o, y_o, d_o) \mid 0 \le o \le N \} \quad (10)$$

Where  $(x_o, y_o)$  is the position of the HIK, N is the number of the HIKs and  $d_o = \min(d_{ih}, d_{is})$ .

#### 3.2. Setting the weights of the HIKs

Since the region around the keypoint represents the local feature with high information, the smaller  $d_o$  is, the richer the local information becomes [7]. Therefore, we set a weight for each HIK according to  $d_o$  to distinguish their importance. Here, we apply the Gaussian function for the distinction as shown in equation (11):

$$w = e^{-\frac{d_0}{2R^2}}$$
(11)

where R is a radius of the circular window. Finally, the HIKs with the weights can be denoted as follows:

$$S_{HIK} = \{ p_n(x_n, y_n, w_n) \mid w_n = e^{-\frac{u_n}{2R^2}} \}$$
(12)

where  $0 < n \le N$  and N is the number of the HIKs,  $(x_n, y_n)$  is the position of the HIK and  $w_n$  is the weight of the HIK. Finally, we get the local descriptors for the HIKs as the SIFT algorithm did and use these descriptors for the image matching.

#### 4. The experiment

To evaluate our proposed approach, we perform the experiments by using two datasets and compare the experimental results with the SIFT algorithm. All the experiments are run on a dual core (TM) 2.4 GHz machine with 2GB of main memory and the Windows XP operating system.

#### 4.1. Image collection

The first dataset is the 3D generic object categorization [17]. We collect 20 groups from the dataset randomly. In each group, there are 5 images with the same object but different viewing angle, different viewing height and different viewing distance.

The second dataset is the Frontal Face Dataset [18]. We collect 50 images from the dataset and separate the images into 10 groups. For each group, the images are with the same person, but different background, expression and brightness.

#### 4.2. Experimental methodology

In this paper, we apply the matching accuracy to evaluate the performance of the proposed approach. The process of getting the matching accuracy includes the next three stages.

1) The distance of two HIKs.

Here, we apply the Euclidean distance to compare two HIKs. For the HIK  $P_i$  in the image I and the HIK  $P_j$  in the image J, the Euclidean distance between them is defined as follows:

$$D_{ij} = |F_i - F_j| = \sqrt{\sum_{k=1}^{N} (F_{i,k} - F_{j,k})^2}$$
(13)

where  $D_{ij}$  is the distance of the HIK  $P_i$  and  $P_j$ .  $F_i$  and  $F_j$  are the features of  $P_i$  and  $P_j$ , respectively. N is the dimension of the feature. The more similar two features are, the closer distance becomes.

2) Determination of the matching pairs and the matching weight.

For a single HIK  $P_i$  in the image I, we calculate the distance values between  $P_i$  and all the HIKs in the other image J. Then we rank the distances from the closest to the farthest. To decide whether  $P_i$  is matched to any point in image J, we use the ambiguity rejection metric, which is the ratio of the closest matching and second closest matching. If the ratio is smaller than 0.8, we determine that the image pair I and J is matched. Since each HIK contains a weight as we have discussed in subsection 3.2, we can get the matching weight for this matching pairs. Suppose that the closest HIK in image J is  $P_i$ , and then the matching weight of this pair is given by the average weight of  $P_i$  and  $P_i$ . Therefore, by comparing the HIKs in images I and J, we can get the total matching weight between both images.

3) Calculation of the matching accuracy

For accessing the matching accuracy, we randomly select one of the images from the image set and use it to match the other images. For one selected image, we can get the matching weight with the other images as shown in Fig.3.



#### Figure 3 The process of the getting the matching weight

In Fig.3,  $I_s$  is the selected image while  $I_1$ ,  $I_2$ ,  $I_3, \ldots, I_n$  are the other images in the image set.  $W_1, W_2, W_3$  $W_{3,...,}W_{n}$  are the corresponding matching weights between the image  $I_s$  and the images  $I_1, I_2, I_3, \dots, I_n$ . Then we rank  $I_1$ ,  $I_2$ ,  $I_3$ ,..., $I_n$  in the order of the matching weights  $(W_1, W_2, W_3, ..., W_n)$  from the greatest to the smallest. The images that are similar to the selected image, which means that they are in the same group with the selected image, will be in the top rank if the matching is correct. By repeating the process of selecting each image from the set and matching it to the other images repeatedly, we can get the total number of correct matches *correctN*. Suppose that the total number of the similar images corresponding to the selected images in the image set is *similarN*. Then, the matching accuracy can be defined as follows:

$$accuracy = \frac{correctN}{similarN}$$
 (14)

#### 4.3. Determination of the windows radius

To determine the suitable radius of the circular window, we execute experiments by using the first dataset. Table 1 shows the matching accuracy according to different radius of the window. In Table 1 the number in the first column denotes the radius of the window and the last column denotes the average time consuming for matching an image pair.

As shown in Table 1, the matching accuracy increases until the radius of the window is 10. After that, the accuracy decreases as the radius of the window increases, and we can find that the matching accuracy doesn't change when the radius of the window is larger than 12.

As we have mentioned in section 2, after detecting the keypoint, the local descriptor is extracted from the region around the keypoint. Since the size of the region around the keypoint is 16\*16 pixels,

considering the image rotation, we get the maximum radius of the window by means of the image rotation formula:

$$x' = (x - x_k)\cos\theta - (y - y_k)\sin\theta + y_k \quad (15)$$

$$y' = (x - x_k)\sin\theta + (y - y_k)\cos\theta + y_k \quad (16)$$

where x, y are the position of points in the region,  $x_k$ ,

 $y_k$  are the position of the keypoint centering in the

region,  $\theta$  is the angle of the rotation, and x', y' denote the new position of the points after the rotation. Thus, the maximum radius of the window can be calculated as follows:

$$R_{\text{max}} = \max(\sqrt{(x' - x_k)^2 + (y' - y_k)^2}) \quad (17)$$
  
Therefore, we can get the maximum radius of the

window with  $8\sqrt{2} \approx 11$ . It is reasonable to set the radius of the window approximately equal to the maximum radius because the descriptor is extracted from this region.

#### 4.4. The evaluation of the proposed approach

As shown in Table 1, when the radius of the window is smaller than 4, the number of the points is less than 50% of the SIFT. Since we prune too many keypoints, the matching accuracy is decreased. However, we can find that when the radius of the window is 4, we can get as same matching accuracy as the SIFT while about 40% of the keypoints are pruned and the average time consuming is only about 36% of the SIFT. Furthermore, Table 1 shows that when the radius of the window is not less than 4, the matching accuracy of the proposed method is equal to or higher than that of the SIFT even though the small number of the keypoints is used. By setting the radius of the window as 10, we can get the best matching accuracy.

To confirm the proposed approach, another experiment is executed by using the second dataset. In the experiment, the radius of the window is set to 10. Table 2 shows that the proposed approach also can get better matching accuracy than the SIFT and improve the efficiency of the matching process.

Fig.4 shows a list of the matching result by using the SIFT algorithm and the proposed approach. Since we use the HIKs and apply the weights to distinguish the importance of the HIKs, we can see that the proposed method gets two correct matching images whereas there is no correct matching image by using the SIFT algorithm.

Table 1 the matching result of the first dataset

Radius of	Matching	Keypoint	Average
window	(%)	number	time
	(70)		consuming
1	19.74	11,873	84.52 ms
2	25.00	16,769	150.44 ms
3	27.63	22,739	284.30 ms
4	31.58	27,965	445.91 ms
5	34.21	32,090	601.88 ms
6	31.58	35,186	711.27 ms
7	34.21	37,594	817.91 ms
8	35.53	39,267	904.93 ms
9	36.84	40,715	1002.73 ms
10	38.16	41,699	1054.01 ms
11	34.21	42,472	1065.21 ms
12	32.89	43,107	1100.82 ms
13	31.58	43,608	1124.81 ms
14	31.58	43,968	1158.45 ms
15	31.58	44,234	1174.36 ms
SIFT	31.58	46,448	1205.18 ms

HIK

SIFT



The second dataset Figure 4 a list of the matching result

Lataoot					
Algorithm	Matching accuracy (%)	Keypoint number	Average time consuming		
SIFT	72.5	32462	3,052 ms		
HIK	77.5	24805	1,177 ms		

Table 2 the matching result of the second dataset

## 5. Conclusion

This paper presents a novel approach to detect the higher information keypoints based on the HSI color space. By decomposing the color image into the intensity, hue and saturation image, we get the HIKs in terms of the keypoint positions in these color component images. To improve the performance, a weight for each HIK is assigned according to the position relationship of the keypoints in the color component images. Since we only use the HIKs with their weights for the image matching, the new approach achieves higher matching accuracy and lower time consuming in the matching process.

#### 6. Acknowledgement

This research was supported by the Defense Acquisition Program Administration and Agency for Defense Development, Korea, through the Image Information Research Center at Korea Advanced Institute of Science & Technology under the contract UD070007AD.

#### 7. References

[1] M. Brown and D. Lowe, 2002, Invariant Features from Interest Point Groups, British Machine Vision Conference, BMVC, pp. 656-665.

[2] Y. Dufournaud, C. Schmid and R. Horaud, 2000, Matching images with different resolutions, In Proceedings of the Conference on Computer Vision and Pattern Recognition, Hilton Head Island, South Carolina, USA, pp. 612-619.

[3] L. J. V. Gool, T. Moons and D. Ungureanu, 1996, Affine/Photometric Invariants for Planar Intensity Patterns, Proc. Fourth European Conf. Computer Vision, pp. 642-651.

[4] C. Harris and M. Stephens, 1988, A combined corner and edge detector, Proceedings of The Fourth Alvey Vision Conference, pp.147-151.

[5] J. J. Koenderink and A. J. V. Doorn, 1987, Representation of Local Geometry in the Visual System, Biological Cybernetics, vol.55, pp.367-375.

[6] D. H. Kim, J. S. Song, J. H. Lee and B. G. Choi, 2007, Support Vector Machine Learning for Region-Based Image Retrieval with Relevance Reedback, ETRI JOURNAL, Vol. 29, No. 5, pp.700-702.

[7] D. H. Kim, S. H. Yu, 2008, A new Region Filtering and Region Weighting Approach to Relevance Feedback in Content-Based Image retrieval, To be appeared in Journal of Systems and Software.

[8] D. G. Lowe, 2004, Distinctive Image Features from Scale-Invariant Keypoint, Computer Vision, pp.91-110.

[9] K. Mikolajczyk and C. Schmid, 2002, An affine invariant interest point detector, in Proc. ECCV. SpringerVerlag, pp.128-142.

[10] K. Mikolajczyk and C. Schmid, 2005, A Performance Evaluation of Local Descriptors, IEEE transactions on pattern analysis and machine intelligence, pp.1615-1630.

[11] K. Mikolajczyk and C. Schmid, 2004, Scale and Affine Invariant Interest Point Detectors, International Journal of Computer Vision, pp.63-86.

[12] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L.V. Gool, 2005, A Comparison of Affine Region Detectors, International Journal of Computer Vision, pp.43-72.

[13] K. Mikolajczyk, 2002, Detection of local features invariant to affine transformations, Ph.D. thesis, Institut National Polytechnique de Grenoble, France.

[14] K. Mikolajczyk and C. Schmid, 2001, Indexing based on scale invariant interest points, International Conference on Computer Vision, Vancouver, Canada, pp.525-531.

[15] C. Schmid and R. Mohr, 1997, Local Grayvalue Invariants for Image Retrieval, IEEE Transactions on Pattern Analysis and Machine Intelligence, pp. 530-535.

[16] T. Tuytelaars and L.V. Gool, 2000, Wide baseline stereo matching based on local, affinely invariant regions, In The Eleventh British Machine Vision Conference, University of Bristol, UK, pp.412-425.

[17]file:///I:/Download/3ddataset/3D%20Object%20Category %20Dataset,%20Savarese%20et%20al.htm

[18] http://www.vision.caltech.edu/html-files/archive.html