# Recognizing Hand Gestures using Local Features: A Comparison Study

Zuhair Zafar[1], Karsten Berns[1], and Aleksandar Rodić[2]

[1] RRLAB, Department of Computer Science,
University of Kaiserslautern,
67663 Kaiserlsautern, Germany
{zafar,berns}@cs.uni-kl.de
http://rrlab.cs.uni-kl.de
[2] Mihailo Pupin Institute,
University of Belgrade,
11060 Belgrade, Serbia
{aleksandar.rodic@pupin.rs}

**Abstract.** Interest point approaches that extract local features from images are commonly used in human action recognition field. In this paper, a comparison study is performed in which different interest point approaches are used. Each approach is discussed with its advantages and drawbacks. Common keypoints extractor like scale invariant features transform (SIFT), speeded up robust features (SURF), etc. are used in context to human hand gestures recognition. In human-robot interaction, efficiency is important in any recognition task along with recognition rate. Hence in this work, performance of 8 different versions of keypoints are evaluated in terms of recognition rates along with their robustness and efficiency with respect to time. SIFT features show best recognition results but SURF and maximally stable extremal regions features (MSER) show better efficiency.

**Keywords:** Keypoints, Human-Robot Interaction, Local Features, Hand Gestures, Depth Sensor

## 1 Introduction

Hands are the most important part of human body. The major purpose of hands is to perform daily life activities. However, they are not only limited to help us in performing different tasks but also, they are used to express different states and behaviors of human. Research demonstrates that the movements we make with our hands when we talk constitute a kind of second language, adding information that is absent from our words. After facial expressions, hand gestures play an important part in expressing the inner state of human.

Human-robot interaction is one of the most emerging topics studied recently. The objective is to develop robots that can not only help humans but also understand their needs, their emotions their actions and their surroundings. For

this robots must have a system that can recognize humans and their actions. Not only the system should be reliable, but also it should be robust and efficient too. As already discussed the importance of hand gestures in different domains of life, there is a need to develop a robotic system that can recognize different hand gestures reliably and efficiently.

Numerous hand gesture applications have been reported so far in the literature. In [1], hand gestures are used to control the VLC player in real time using principle component analysis and k-nearest neighbor algorithm. Another classical appearance-based approach for hand tracking is used in [2]. They used a eigen tracker to be able to detect two hands. Color and motion cues are used for initialization. The eigen space is updated online to incorporate new viewpoints. Neural network is used to handle illumination variations. In [3], hand positions are localized by detecting skin colored blobs in the images using Bayesian classifier. Then, hand pose is estimated by detecting the fingers. In [4], hand position are detected in the image using Camshift. A scale and rotation invariant hand descriptor is obtained by computing a contour. After locating the hand position, a semicircle detector is used to detect the finger tips. Using particle filtering and k-means algorithm, the finger positions are computed. Jain [5] implemented a vision based hand gesture pose estimation based application for mobile devices. Pavlovic et al. [6] accomplished in their work that the gestures of users must be explained logically for developing a good human computer interaction based system. Another hand gesture recognition method based on input-output Hidden Markov Models of tracking skin color blobs was proposed by Marcel et al.[7]. The sign language tutoring tool studied by Aran et al. [8] which their research designed to teaching the fundamental of the sign language in interactive way.

Microsoft Kinect SDK and OpenNI (NiTe Middleware library), both provide human joints and skeletal information. Since the body and hand tracking are reliable, hence researchers focus more on classification of hand gestures instead of localization of hands. NiTe Middleware library provides 3-D position of hand which can be used to segment the hand. In this domain of study, we present a system that recognize automatically different hand gestures in real-time. We present different interest point approaches and show their effectiveness and robustness during human robot interaction. We discuss the advantages and disadvantages of each interest point approach and based on reliability and efficiency, we select the best one out of them. This paper is organized as follows: Section 2 describes introduces different interest point detection approaches in detail. Classification of hand gestures are discussed in Section 3. We evaluate our methodology in section 4 and in the end, we conclude the paper in section 5.

## 2 Hand gesture recognition using interest points

The proposed work is presented in context of human-robot interaction. Using depth data, which is independent of illumination variation, hands position can be determined easily. The challenge in registering hand is to extract it irrespective of the scale. If hand is near to the sensor, hand region would be greater and if it

is far from the sensor, then hand region would be smaller. In order to segment the hand region, there exists different strategies. The most suitable and efficient way of segmentation of hands is to use the depth value of hand. In our study, we use depth value to construct a square window size [15]. Using empirical studies, a linear relationship in between depth of hand and window size is generated as shown in 1,

$$d = 100 - \frac{z - 500}{15} \ . \tag{1}$$

In above formulation, $d$ is the side of square window and $z$ is the depth value of hand. $d$ would be bigger if depth of the hand, $z$, is lower. This formulation is valid for $640 \times 480$ resolution. For $320 \times 240$ resolution, the side of the window will be half. As can be seen from 1, if the depth value of hand is increased, the window size decreases. Since the NiTe library provides hand positions accurately in the depth range of 500 to 2000 mm, hence the system detects and segment the hands in this region. After 2000 mm distance, hand window becomes too small and recognition of hand gesture is impossible. Hand window is segmented from the color image which is further processed for hand gesture recognition.

The next step after segmentation and preprocessing is to extract features. In this work, we use interest point features to recognize hand gestures. Interest point features are type of local features which exploits the pattern locally. These local features are then translated in a specific way to extract global features. In the following section, detection of different local features and computing of descriptors have been explained.

### 2.1 SIFT features

Scale invariant feature transform (SIFT) features are local features and based on the appearance of the object at specific interest points. SIFT is presented by Lowe [9]. The algorithm extract keypoints (interest points) by convolving the image with Gaussian filters at different scales. The difference of Gaussian (DoG) is computed between consecutive Gaussian blurred images and keypoints are then identified as local maxima or minima of the DoG. In order to process these keypoints, a descriptor is needed to encode all the information of the keypoint. A set of orientation histograms are computed on $4 \times 4$ pixel neighborhoods with 8 bins. This orientation information stored in the descriptor makes SIFT descriptor rotation invariant to an extent. There are 16 histograms, each with 8 bins, hence the descriptor size becomes 128.

SIFT features are rotation invariant upto affine transformation of 50 degrees and are illumination invariant. The features extracted are unique and captured the most amount of variance as compared to other features. SIFT features are also scale invariant upto 2.5 meter and outperforms other local features apart from SURF. The main disadvantage of these features is that SIFT descriptors are high dimensional and therefore, it can make the system computationally intensive.

### 2.2 SURF features

Speeded up robust features (SURF) performs much faster as compared to SIFT features and slightly more robust than SIFT. SURF is presented by Bay et al. [10]. In SURF, box shaped filters are used as an approximation of Gaussian smoothing. Integral image is used for filtering the original image with box shaped filters of different sizes. In order to determine interest points, a blob detector based on Hessian matrix is used. For scale and location of keypoints, the SURF algorithm depends on the determinant of Hessian matrix. For orientation assignment, SURF uses wavelet responses in horizontal and vertical direction for a neighborhood of size 6s. For description of features, SURF uses Haar wavelet responses in horizontal and vertical direction in an integral image. A neighborhood of size $20s \times 20s$ is considered around the interest point where s is the size. This neighborhood is further divided in $4 \times 4$ subregions. For each subregion, horizontal and vertical wavelets responses are taken and vector is formed which consists of 64 dimensions.

SURF features are local features and like SIFT features, SURF features also are scale and rotation invariant. The most important advantage of using these features is the lesser computational cost in contrast to SIFT. The reason lies in the descriptor dimensionality which is 64 as compared to the SIFT descriptor of 128 dimensions. However, a short descriptor may be more robust against appearance variations, but may not offer sufficient discrimination and thus give too many false positives.

### 2.3 Dense features

Features that are sampled densely of the same scale and orientations are known as dense features [11]. In this type of feature detection, more features are computed at each location and scale in an image. This provides all possible information at every location in the image. Since dense features are detected at each location of an image, therefore, SIFT or SURF descriptor also has to be computed at each location. This process makes the system computationally more intensive. Cases where the notion of time is not important, dense SIFT can report better results than normal SIFT.

### 2.4 FAST features

Features from accelerated segment test (FAST) is a corner detection method, which can be used to compute interest points and then used them for tracking, classification or recognition tasks. FAST corner detector [12] uses a circle of 16 pixels to validate whether the point $p$ is a corner. Each pixel in the circle is labeled from 1 to 16. If all the pixels or $N$ pixels in the circle are brighter than the pixel or darker than the pixel with some threshold then $p$ is classified as a keypoint. These keypoints are then describe by using SIFT/SURF descriptors. The dimensionality remain 128 in case of SIFT and 64 in case of SURF, but to some extent these features are scale invariant.

The advantage of FAST corner detector is its computational efficiency. As the name suggest, it is fast and indeed it is faster than many other well-known feature extraction methods, such as difference of Gaussian (DoG) used by SIFT, SUSAN and Harris. FAST corner detector is very suitable for real-time video processing application because of high-speed performance. However, the main disadvantage lies in computing keypoints. There is a trade off in selecting $n$ number of pixels. Number of keypoints detected should not be too many and on the other hand if they are too few then it may effect on the recognition rate.

### 2.5 MSER features

Maximally stable extremal regions (MSER) are used as a blob detection method in an image. MSER is based on the idea of taking regions which stay nearly the same through a wide range of thresholds. The algorithm [13] works in a way that initially all the pixels below a given threshold are white and pixels above or equal to the threshold are black. All those white spots are then merge together, till all the image is white. The set of all the connected components in the image is basically the extremal regions of an image. Additionally, elliptical frames are attached to MSERs by fitting ellipses to the regions. These elliptical regions are used as feature points of the image. For describing the feature points, SIFT/SURF descriptor are used.

The biggest benefit of MSER features is that they are invariant to affine transformation of image intensities. The extracted regions are stable and can still report stable regions even the image is skewed. However, the approach is sensitive to illumination changes or shadows and motion blur. Despite this, MSER performs well for small regions and shows good repeatability and computationally lighter than other region detectors.

## 3 Classification

Before classification, the extracted local features generated by different keypoints algorithms are represented globally by using bag-of-features approach (BoF) and k mean clustering [15]. This step represents local features into global features for whole image without effecting the characteristics of local features.

In our study, we use support vector machines (SVM) for classification task. SVM is a supervised learning algorithm that is able to perform classification and regression [14]. For training and testing phases, a database, containing 6 subjects with different ethnicities, is generated. The database is recorded by using ASUS Xtion sensor. Only segmented hand regions are stored. Total of 10 different static gestures are recorded. 150 frames are taken for each gesture of a subject. Fig. 1 shows the segmented images from the database.

We use one-vs-all SVM approach for our multi-class problem. The feature vector, which we have created using bag-of-features approach with labels on training images, are fed into the multi-class support vector machines in order to obtain the model. This model is further used in testing stage for hand gesture

recognition. In this domain of study, SVM with RBF kernel is used for training purpose.

## 4 Experimentation and evaluation

Since this work is in context of Human-Robot interaction, hence a humanoid robot, Robothespian, is used in the study. It also consists of intelligent hands with 8 degrees of freedom. The whole arm has 14 DoF. The robot also have speech synthesis module, through which it can speak in English as well as German language. An RGB-D sensor is installed on the chest of the robot in addition with HD camera, which is attached on the head of the robot. Robot can also move his head 45 degrees and the torso to 20 degrees each side.

Table 1 shows the performance of hand gestures using different interest point approaches. Recognition rates for different hand gestures has been reported. We recognize number gestures as shown in Fig. 1. From the Table 1, it can be seen that SIFT features reports better results any other type of features. MSER-SURF performs better than MSER-SIFT. In MSER-SIFT, the gesture *two* and *three* are confused with each other. The same phenomena happened in MSER-SURF as well. The recognition rate of gesture *three* is 65%, which shows that the blob detected for this gesture is not accurate enough. In FAST-SURF and FAST-SIFT, gesture *one* and *six* are interchangeably recognized, resulting in low recognition rate for both these gestures. The reason is easier to understand as in both these gestures, there is only one finger open (index finger for *one* and little finger for *six* gesture). On the other hand, FAST-SIFT recognize gesture *eight* rarely as compared to all the other kinds of interest point approaches which also makes the overall recognition rate to reduce.

Dense SIFT also reports good result, however, because of too many keypoints, recognition rate is effected. Generally, dense SURF failed badly as the SVM classifier is unable to generate hyperplane, which results in poor classification results. The major reason is that the extracted keypoints are randomly distributed and even SURF descriptor does not describe them efficiently, resulting in poor classification.

Another more critical aspect in human-robot interaction is the efficiency and robustness of implemented approach. Table 2 shows the average processing time for a frame for each interest point approach and also and average frames per second information as well.

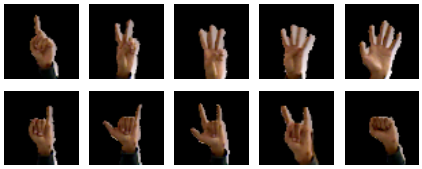

**Fig. 1.** Hand segmented images of number gestures (1-10).

**Table 1.** Recognition Rates in percentage (%) of number gestures using different approaches with SIFT(I) or SURF(U) descriptors on 150 images for each gesture.

| Gestures | SIFT | SURF | Dense(I) | Dense(U) | FAST(I) | FAST(U) | MSER(I) | MSER(U) |
|---|---|---|---|---|---|---|---|---|
| One | 90 | 93.3 | 96 | 0 | 86 | 74 | 88.6 | 92 |
| Two | 98.6 | 97.3 | 96.6 | 0 | 92 | 91.3 | 60 | 94.6 |
| Three | 92 | 84 | 73.3 | 0 | 88.6 | 92.6 | 69.3 | 64.6 |
| Four | 88.6 | 86 | 86 | 0 | 77.3 | 92.6 | 86.6 | 96 |
| Five | 98 | 96 | 98 | 100 | 98.6 | 94.6 | 86.6 | 92.6 |
| Six | 94.6 | 97.3 | 95.3 | 0 | 48 | 72 | 98.6 | 98.6 |
| Seven | 98.6 | 99.3 | 99.3 | 0 | 98 | 96.6 | 97.3 | 99.3 |
| Eight | 97.3 | 96.6 | 99.3 | 0 | 56 | 91.3 | 89.3 | 88 |
| Nine | 97.3 | 96.6 | 75.3 | 0 | 90 | 90.6 | 95.3 | 98 |
| Ten | 91.3 | 89.1 | 91.3 | 0 | 86.9 | 89.1 | 91.3 | 86.9 |
| Avg. | **94.8%** | 93.7% | 91% | 10% | 82% | 88.5% | 86.1% | 91.2% |

**Table 2.** Table shows average processing time in millisecond for a frame and average frames processed per second for each interest point approach.

| | SIFT | SURF | Dense(I) | Dense(U) | FAST(I) | FAST(U) | MSER(I) | MSER(U) |
|---|---|---|---|---|---|---|---|---|
| Avg. Time | 77ms | 64ms | 83ms | 89ms | 58.7ms | 111ms | 65.5ms | 61.6ms |
| Frames/sec | 13 | 15.6 | 12 | 11.2 | 17 | 9 | 15.3 | 16.2 |

From Table 2, it can be seen that generally, SIFT features takes more time as compared to SURF features. Dense SIFT performs above 90% recognition rate however, it takes more time to process single frame, which constitutes 12 frames per second as compared to 13 frames per second. On the other hand, MSER interest point are one of the efficient features. However, the recognition rate for MSER features are not so high. FAST-SIFT reports most efficiency in terms of frames processed but again, the recognition rate is average especially for *six* and *eight* gestures. Approaches, with processing over 12 frames per second, can be regarded as real time. In our case, SIFT features reports better recognition rate and satisfactory processing time and can be selected over other approaches.

## 5 Conclusion

Interest point features are commonly used in human action recognition field. In this work, we study different interest point approaches and discuss their advantages and drawbacks. SIFT, SURF, MSER-SIFT, MSER-SURF, FAST-SIFT, FAST-SURF, Dense SIFT and Dense SURF features are used in human hand gesture recognition. Efficient hand segmentation using OpenNI and NiTe library can be done by using a linear relation ship between the window size and the depth. We classified each approach using feature vectors generated by bag-of-features method. Multi-class SVM classification algorithm is used for classifi-

cation. After experimentation, SIFT features reports best recognition rate and SURF follows just behind it. MSER, FAST and SURF tends to perform more efficiently as compared to SIFT with respect to time. In future work, we propose to use SIFT features in recognizing dynamic hand gestures along with hidden markov model.

## References


1. Rautaray, S.S., Agrawal, A.: A Novel Human Computer Interface Based On Hand Gesture Recognition Using Computer Vision Techniques. In: Proceedings of ACM IITM'10, pp. 292-296 (2010)
2. Barhate, K.A., Patwardhan, K.S., Roy, S.D., Chaudhuri, S., Chaudhury, S.: Robust shape based two hand tracker. In: Image Processing, 2004. ICIP '04. 2004 International Conference on, pp. 1017-1020 (2004)
3. Argyros, A.A., Lourakis, M.I.A.: Tracking multiple colored blobs with a moving camera. In: Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on, pp. 1178 (2005)
4. Wang, X., Zhang, X., Dai, G.: Tracking of deformable human hand in real time as continuous input for gesture-based interaction. In: Proceedings of the 12th international conference on Intelligent user interfaces, pp. 235-242 (2007)
5. Jain, G.: Vision-based hand gesture pose estimation for mobile devices. University of Toronto, (2009)
6. Pavlovic. V., Sharma, R., Huang, T.S.: Visual interpretation of hand gestures for human-computer interaction: A review. In: IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI), 7(19):pp. 677-695 (1997)
7. Marcel, S., Bernier, O., Viallet, J.E., Collobert, D.: Hand Gesture Recognition using Input-Output Hidden Markov Models. In: Proc. of the FG'2000 Conference on Automatic Face and Gesture Recognition. (2000)
8. Aran, O., Ari, I., Benoit, F., Campr, A., Carrillo, A.H., Fanard, Akarun, L., Caplier, A., Rombaut, M., Sankuru, B.: Sign Language Tutoring Tool. eNTERFACE 2006, The Summer Workshop on Multimodal Interfaces, Croatia. (2006)
9. Lowe, David G.: Object recognition from local scale-invariant features. In: Proceedings of the International Conference on Computer Vision, pp. 11501157 (1999)
10. Funayama, R., Yanagihara, H., Gool, L.V., Tuytelaars, T., Bay, H.: Robust interest point detector and descriptor. published 2009-09-24 (2009)
11. Veksler, O.: Dense features for semi-dense stereo correspondence. In: IJCV, 47(1-3):pp. 247260 (2002)
12. Rosten, E., Drummond, T.: Fusing points and lines for high performance tracking. In: IEEE International Conference on Computer Vision 2: 15081511 (2005)
13. Matas, J., Chum, O., Urban, M., Pajdla, T.: Robust wide baseline stereo from maximally stable extremal regions. In: Proceedings of British Machine Vision Conference, pp. 384-396 (2002)
14. Cortes, C., Vapnik, V.: Support-vector networks. Machine Learning 20 (3): 273 (1995)
15. Zafar, Z., Berns, K.: Recognizing Hand Gestures for Human-Robot Interaction. In:Proceedings of the 9th International Conference on Advances in Computer-Human Interactions (ACHI), pp. 333-338 (2016)