Automated Identification and Reporting System by Face Recognition Using SIFT Algorithm

M. V. Rajarao, M. S. Raviteja, C. V. S. R. Phani kiran, and K. L Nrao Om Srivathsa
Veramasa, Hyderabad, India

***Abstract*—Face recognition is an integral part of biometrics. In biometrics basic traits of human is matched to the existing data and depending on result of matching identification of a human being is traced. Facial features are extracted and implemented through algorithms which are efficient and some modifications are done to improve the existing algorithm models. A face recognition system using the SIFT (Space invariant feature transformation) algorithm was implemented. The algorithm is based on Image features approach which represents a SIFT method in which a small set of significant features are used to describe the variation between face images. Experimental results for different numbers of faces are shown to verify the viability of the proposed method.**

***Index Terms*—face recognition, facial features, SIFT**

I. Introduction

Facial recognition was the source of motivation behind the creation of eigenfaces. For this use, eigenfaces have advantages over other techniques available, such as the system's speed and efficiency. Using eigenfaces is very fast, and able to functionally operate on lots of faces in very little time. Unfortunately, this type of facial recognition does have a drawback to consider: trouble recognizing faces when they are viewed with different levels of light or angles. For the system to work well, the faces need to be seen from a frontal view under similar lighting. Face recognition using eigenfaces has been shown to be quite accurate. By experimenting with the system to test it under variations of certain conditions, the following correct recognitions were found: an average of 96% with light variation, 85% with orientation variation, and 64% with size variation. To complement eigenfaces, another approach has been developed called Eigen features. This combines facial metrics (measuring distance between facial features) with the Eigenface approach. Another method, which is competing with the Eigenface technique, uses 'fisher faces'. This method for facial recognition is less sensitive to variation in lighting and pose of the face than the method using eigenfaces.

A more modern alternative to eigenfaces and fisher faces is the active appearance model, which decouples the face's shape from its texture: it does an Eigenface decomposition of the face after warping it to mean shape. This allows it to perform better on different projections of the face, and when the face is tilted.

A. Face Recognition based Attendance Marking System

The system consists of a camera that must be positioned in the office room to take snap shots of the room. These images are then sent to an enhancement module where Histogram Normalization is used for the contrast enhancement of the image, Median Filter is used for removing noise from the image.

To avoid false detection skin classification technique is used. This process first classifies the skin and then retains only the skin pixels and other the other pixels are set to black. The enhanced image is then sent to a face detection and recognition module. This requires MATLAB software version 7.6. Two databases are maintained, the first one is the Face database to store the face images and extracted features at the time of enrolment process and the second attendance database contains the information about the employees and is also used to mark attendance.

II. Scale Invariant Feature Transform (SIFT)

Image matching is a fundamental aspect of many problems in computer vision, including object or scene recognition, solving for 3D structure from multiple images, stereo correspondence, and motion tracking. This paper describes image features that have many properties that make them suitable for matching differing images of an object or scene. The features are invariant to image scaling and rotation, and partially invariant to change in illumination and 3D camera viewpoint. They are well localized in both the spatial and frequency domains, reducing the probability of disruption by occlusion, clutter, or noise. Large numbers of features can be extracted from typical images with efficient algorithms. In addition, the features are highly distinctive, which allows a single feature to be correctly matched with high probability against a large database of features, providing a basis for object and scene recognition.

The cost of extracting these features is minimized by taking a cascade filtering approach, in which the more expensive operations are applied only at locations that pass an initial test.

A. Stages of Computation used to Generate the Set of Image Features

This approach has been named the Scale Invariant Feature Transform (SIFT), [1] as it transforms image data into scale-invariant coordinates relative to local features. An important aspect of this approach is that it generates

Manuscript received July 10, 2017; revised October 9, 2017.

doi: 10.18178/ijeetc.6.4.12-16

large numbers of features that densely cover the image over the full range of scales and locations.

A typical image of size 500x500 pixels will give rise to about 2000 stable features (although this number depends on both image content and choices for various parameters).

The quantity of features is particularly important for object recognition, where the ability to detect small objects in cluttered backgrounds requires that at least 3 features be correctly matched from each object for reliable identification. For image matching and recognition, SIFT features are first extracted from a set of reference images and stored in a database. A new image is matched by individually comparing each feature from the new image to this previous database and finding candidate matching features based on Euclidean distance of their feature vectors. This paper will discuss fast nearest-neighbor algorithms that can perform this computation rapidly against large databases

The key point descriptors are highly distinctive, which allows a single feature to find its correct match with good probability in a large database of features. However, in a cluttered image, many features from the background will not have any correct match in the database, giving rise to many false matches in addition to the correct ones. The correct matches can be filtered from the full set of matches by identifying subsets of key points that agree on the object and its location, scale, and orientation in the new image.

- Scale-space extreme detection: The first stage of computation searches over all scales and image locations. It is implemented efficiently by using a difference-of-Gaussian function to identify potential interest points that are invariant to scale and orientation

  The first stage of keypoint detection is to identify locations and scales that can be repeatably assigned under differing views of the same object. Detecting locations that are invariant to scale change of the image can be accomplished by searching for stable features across all possible scales, using a continuous function of scale known as scale space (Witkin, 1983). It has been shown by Koenderink (1984) and Lindeberg (1994) that under a variety of reasonable assumptions the only possible scale-space kernel is the Gaussian function. Therefore, the scale space of an image is defined as a function, D(x,y, σ), that is produced from the convolution of a variable-scale Gaussian, , with an input image, I(x, y):

  $$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y),$$

  where $*$ is the convolution operation in x and y, and

  $$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2}.$$

To efficiently detect stable keypoint locations in scale space, we have proposed (Lowe, 1999) using scale-space extrema in the difference-of-Gaussian function convolved with the image, which can be computed from the difference of two nearby scales separated by a constant multiplicative factor k:

$$\begin{aligned} D(x, y, \sigma) &= (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) \\ &= L(x, y, k\sigma) - L(x, y, \sigma). \end{aligned} \quad (1)$$

There are a number of reasons for choosing this function. First, it is a particularly efficient function to compute, as the smoothed images, L, need to be computed in any case for scale space feature description, and D can therefore be computed by simple image subtraction.

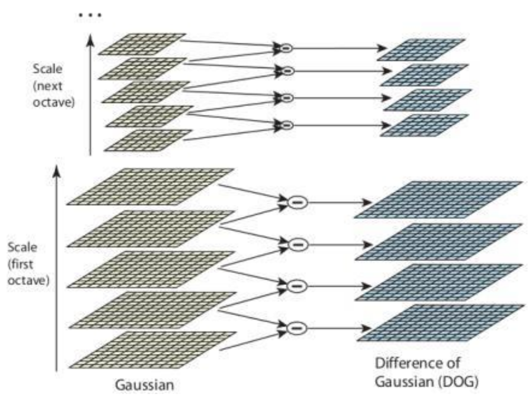


Figure 1.

- Key point localization: At each candidate location, a detailed model is fit to determine location and scale. Key points are selected based on measures of their stability.

  In addition, the difference-of-Gaussian function provides a close approximation to the scale-normalized Laplacian of Gaussian, as studied by Lindeberg (1994).[2][1][3] Lindeberg showed that the normalization of the Laplacian with the factor is required for true scale invariance. In detailed experimental comparisons, Mikolajczyk (2002) found that the maxima and minima of produce the most stable image features compared to a range of other possible image functions, such as the gradient, Hessian, or Harris corner function.

  The relationship between D and can be understood from the heat diffusion equation (parameterized in terms of rather than the more usual) From this, we see that can be computed from the finite difference approximation to using the difference of nearby scales at and therefore, This shows that when the difference-of-Gaussian function has scales differing by a constant factor it already incorporates the scale normalization required for the scale-invariant

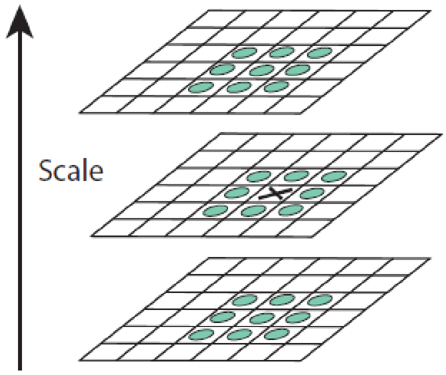


Figure 2.

Maxima and minima of the difference-of-Gaussian images are detected by comparing a pixel (marked with X) to its 26 neighbors in 3x3 regions at the current and adjacent scales (marked with circles).Laplacian. The factor (k − 1) in the equation is a constant over all scales and therefore does not influence extreme location. The approximation error will go to zero as k goes to 1, but in practice we have found that the approximation has almost no impact on the stability of extrema detection or localization for even significant differences in scale, such as $k = \sqrt{2}$. An efficient approach to construction of $D(x, y, \sigma)$ is shown in Fig. 2. The initial image is incrementally convolved with Gaussians to produce images separated by a constant factor k in scale space, shown stacked in the left column. We choose to divide each octave of scale space (i.e., doubling of σ). into an integer number, s, of intervals, so $k = 2^{1/s}$. We must produce s + 3 images in the stack of blurred images for each octave, so that final extreme detection covers a complete octave. Adjacent image scales are subtracted to produce the difference-of-Gaussian images shown on the right. Once a complete octave has been processed, we resample the Gaussian image that has twice the initial value of σ (it will be 2 images from the top of the stack) by taking every second pixel in each row and column. The accuracy of sampling relative to σ is no different than for the start of the previous octave, while computation is greatly reduced.

In order to detect the local maxima and minima of $D(x, y, \sigma)$, each sample point is compared to its eight neighbors in the current image and nine neighbors in the scale above and below (see Fig. 2). It is selected only if it is larger than all of these neighbors or smaller than all of them. The cost of this check is reasonably low due to the fact that most sample points will be eliminated following the first few checks.

An important issue is to determine the frequency of sampling in the image and scale domains that are needed to reliably detect the extreme. Unfortunately, it turns out that there is no minimum spacing of samples that will detect all extreme, as the extreme can be arbitrarily close together. This can be seen by considering a white circle on a black background, which will have a single scale space maximum where the circular positive central region of the difference-of-Gaussian function matches the size and location of the circle. For a very elongated ellipse, there will be two maxima near each end of the ellipse. As the locations of maxima are a continuous function of the image, for some ellipse with intermediate elongation there will be a transition from a single maximum to two, with the maxima arbitrarily close to

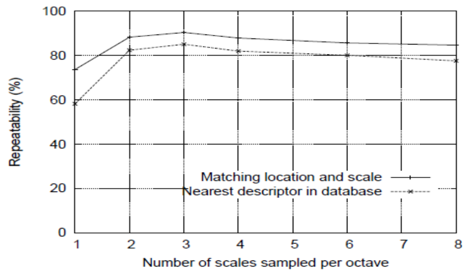


Figure 3.

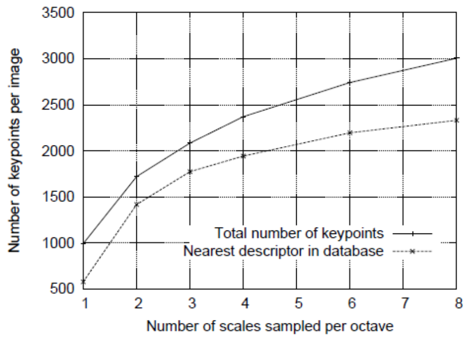
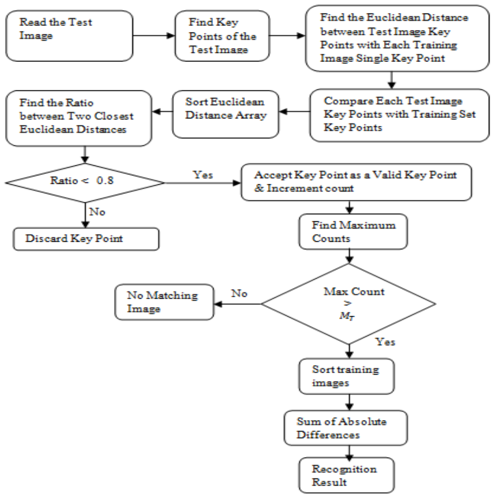


Figure 4.

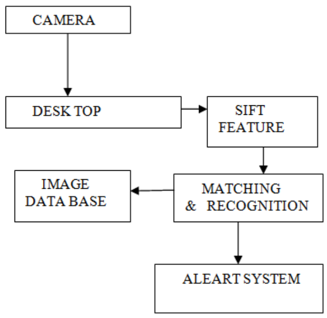
The top line of the first graph shows the percent of keypoints that are repeatable detected at the same location and scale in a transformed image as a function of the number of scales sampled per octave. The lower line shows the percent of keypoints that have their descriptors correctly matched to a large database. The second graph shows the total number of key points detected in a typical image as a function of the number of scale samples each other near the transition. Therefore, we must settle for a solution that trades off efficiency with completeness. In fact, as might be expected and is confirmed by our experiments, extreme that are close together are quite unstable to small perturbations of the image. We can determine the best choices experimentally by studying a range of sampling frequencies and using those that provide the most reliable results under a realistic simulation of the matching task.

III. SIFT Flow Chart



With the above flow chart we can understand features extraction of a run and stored data base image.

B. *SIFT Block Diagram*



IV. Results and Discussion

Two benchmark databases are employed for comparison purposes. The first is AT&T face database [5], containing 400 images for 40 persons with 10 images/person. There are different orientations and facial expressions for each subject. The image size is 112×92 pixels. There is an average of 70 SIFT features extracted from each image. Fig. 1 shows a sample of images for one subject. The second database is Yale face database [1]. It contains 165 images for 15 subjects, with 11 images/person. The images contain different facial expressions and illumination conditions for each subject. The image size is 243 × 320 pixels, and an average of 230 SIFT features are extracted for each image. The raw faces were used without any kind of preprocessing (cropping, normalization, histogram equalization, etc.) to assess the robustness of the algorithms in the comparison.

Two more experiments were carried out to check the performance with different training set sizes. The first was run using training set of size 20% and test set of 80%, while the second using 80% training and 20% testing. In all the experiments, 10 independent trials were performed with randomly chosen training and test sets. Table II shows the results. As expected, the performance degrades with smaller training set size and increases with larger training set. It is also clear that the SIFT approach is better than the others. The performance is significantly better in Yale database using the smaller training set (90.1% for SIFT vs. 73.3% for Eigenfaces and 83.5% for Fisherfaces).

TABLE I. Baseline Accuracy Results. The best Results are in Boldface

	Eigenfaces					
	Nearest Neighbor			Nearest Cluster Center		
	Euclidean	City-block	Cosine	Euclidean	City-block	Cosine
AT&T	89.3	92.9	89.0	74.7	87.1	73.7
Yale	68.4	72.0	68.0	57.7	72.1	59.4
	Fisherfaces					
	Nearest Neighbor			Nearest Cluster Center		
	Euclidean	City-block	Cosine	Euclidean	City-block	Cosine
AT&T	91.3	90.8	93.8	91.4	91.1	93.7
Yale	83.4	86.8	86.4	83.8	86.9	84.6
	SIFT					
	Cosine			Angle		
AT&T	93.7			96.3		
Yale	85.8			91.7		

TABLE II. Training Set Size Results

	Eigenfaces					
	Nearest Neighbor			Nearest Cluster Center		
	Euclidean	City-block	Cosine	Euclidean	City-block	Cosine
AT&T 20%	76.0	80.1	76.1	71.6	79.2	70.0
Yale 20%	69.5	73.3	72.0	58.9	69.9	62.1
AT&T 80%	96.0	97.2	95.5	78.6	91.3	76.5
Yale 80%	81.3	83.0	81.0	70.0	78.6	76.3

	Fisherfaces					
	Nearest Neighbor			Nearest Cluster Center		
	Euclidean	City-block	Cosine	Euclidean	City-block	Cosine
AT&T 20%	76.8	74.7	84.6	79.0	77.4	85.0
Yale 20%	83.4	82.3	82.3	83.5	82.5	82.0
AT&T 80%	95.2	94.1	96.0	95.6	94.6	96.2
Yale 80%	87.0	89.6	89.3	87.0	89.6	89.3

	SIFT	
	Cosine	Angle
AT&T 20%	79.6	85.7
Yale 20%	84.7	90.1
AT&T 80%	99.0	99.7
Yale 80%	92.0	95.6

V. Summary

This paper presents a new approach for face recognition, based on matching SIFT fea-tures. The new approach is compared to Eigenfaces and Fisherfaces, and proved supe-rior to both of them in all experiments, specially with smaller training set sizes. Upon investigating the effective number of SIFT features required for reliable matching, the experiments reveal that we need only 30% of the features, which saves 91% of the time needed to match all the extracted features. In addition, the SIFT features approach continues to provide superior performance for up to 50% reduction in resolution.

References

[1] M. A. Turk and A. P. Pentland, "Face Recognition Using Eigenfaces," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pp. 586–591. 1991.

[2] A. J. Goldstein, L. D. Harmon, and A. B. Lesk, "Identification of human faces," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, vol. 59, pp. 748–760, May 1971.

[3] A. Elschlager, "The representation and Matching of pictorial structures," *IEEE Transactions on Computer*, vol. C-22, pp. 67-92, 1973.

[4] S. S. R. Abibi, "Simulating evolution: Connectionist metaphors for studying human cognitive behaviour," in *Proc. TENCON 2000*, vol. 1, pp. 167-173, 2000.

[5] Y. Cui, J. S. Jin, S. Luo, M. Park, and S. S. L. Au, "Automated Pattern Recognition and Defect Inspection System," in *Proc. 5th International Conference on Computer Vision and Graphical Image*, vol. 59, pp. 768–773, May 1992.

[6] Y. Zhang and C. Liu, "Face recognition using kemel principal component analysis and genetic algorithms," in *Proc. IEEE Workshop on Neural Networks for Signal Processing*, Sept. 2002, pp. 4-6.

[7] J. Zhu and Y. L. Yu, "Face recognition with eigenfaces," in *Proc. IEEE International Conference on Industrial Technology*, pp. 434 -438, Dec. 1994.

[8] M. H. Yang, N. Ahuja, and D. Kriegmao, "Face recognition nusing kernel eigenfaces," in *Proc. IEEE International Conference on Image Processing*, vol. 1, pp. 10-13, Sept. 2000.

[9] S. Se, D. G. Lowe, and J. Little, "Global localization using distinctive visual features," in *Proc. International Conference on Intelligent Robots and Systems, Lausanne*, Switzerland, pp. 226-231, 2002.

Raviteja M. S.
Full postal address. Plot no 42 Samathapuri colony Hyd