Survey of Feature Points Detection and Matching using SURF, SIFT and PCA-SIFT

[1]Utsav Shah, [2]Darshana Mistry, [3]Yatin Patel

[1]Student, [2]Technical Associate, [3]Assistant Professor
[1]Computer Engineering, Gujarat Technological University, Ahmedabad, India,
[2] Software Department, eInfochips Training and Research Academy (eiTRA), Ahmedabad, India
[3] Faculty of Engineering, SSESGI, Mehsana, India

***Abstract*— This paper summarizes the three robust feature detection and matching methods: Scale Invariant Feature Transform (SIFT), Principal Component Analysis (PCA)–SIFT and Speeded Up Robust Features (SURF). SIFT find its interest points using Difference of Gaussian (DoG). SIFT presents its stability in most situations although it's slow. PCA-SIFT show its advantages in rotation and illumination change and it is faster than SIFT.SURF is the fastest one with good performance as the same as SIFT. 'Fast-Hessian' detector that used in SURF is more than 3 times faster that DOG.**

***Index Terms*— Scale Invariant Feature Transform (SIFT), Principal Component Analysis (PCA)-SIFT, Speeded Up Robust Feature (SURF).**

I. INTRODUCTION

In this modern era, computers and the internet represent the major communication media that connect different parts of the world in one global virtual world. In digital communication, image and videos are also play major role in the world. It is necessary to detect and matching robust features of an image and video (sequence of frames/images). Detect and matching of features points are useful in image registration, object detection, camera calibration, 3D reconstruction, steganography etc.

Feature points are distinctive points as points, lines, edges, corners, blobs, T-Junctions etc. It is necessary that feature points are invariant to rotation, scaling, translation, illumination change. Repeatability is the most valuable property of a feature (interest) point detector and which expresses the reliability of a detector under different viewing conditions for finding the same physical interest points. In this paper, we introduce different approaches as SIFT, PCA-SIFT, SURF which are detected and matching features.

II. SIFT (SCALE INVARIANT FEATURE TRANSFORM)

D. G. Lowe(2004) presented that SIFT algorithm is a method to extract and describe feature points, which is robust to scale, rotation and change in illumination.

There are four steps to implement the SIFT algorithm:

1. Scale-space extrema detection
2. Feature point localization
3. Orientation assignments
4. Feature point descriptor

1) Scale-space Extrema Detection:

The first stage searches over scale space using a Difference of Gaussian (DoG) function to identify potential interest points that are invariant to scale and orientation. The scale space of an image is defined as a function L (x , y , σ) which is produced from the convolution of a variable-scale Gaussian G(x , y , σ) with an input image I(x,y):

$$\mathbf{L(x\,,y\,,\sigma) = G\,(x\,,y\,,\sigma) * I\,(\,x\,,y\,)\,\ldots\ldots\ldots.(1)}$$

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

To efficiently detect stable key-point locations in scale space using scale-space extrema in the difference-of-Gaussian function convolved with the image, D(x , y , σ)which can be computed from the difference of two nearby scales separated by a constant multiplicative factor k :

$$\mathbf{D(x\,,y\,,\sigma) = \big(G(x\,,y\,,k\sigma) - G(x\,,y\,,\sigma)\big) * I(x,y)}$$
$$\mathbf{= L(x\,,y\,,k\sigma) - L(x\,,y\,,\sigma)\;\ldots\ldots\ldots\ldots\ldots\quad(3)}$$

2) Feature Point Localization:

The location and the scale of each candidate point are determined and the feature points are selected based on measures of stability this information allows points to be rejected that have low contrast (and are therefore sensitive to noise) or are poorly localized along an edge.

3) Orientation Assignment:

One or more orientations are assigned to each feature point location based on local imagegradient directions. For each image sample at this scale L(x, y), the gradient magnitude m(x, y) and orientation $\theta(x, y)$are precomputed using pixel differences:

$$\mathbf{m(\,x\,,y\,) = \sqrt{(L(\,x+1,y) - L(\,x-1,y))^2 + (L(\,x,y+1) - L(\,x,y-1))^2}}$$

$$\boldsymbol{\theta(x,y) = \tan^{-1}\left(\frac{(L(\,x,y+1) - L(\,x,y-1))}{(L(\,x+1,y) - L(\,x-1,y))}\right)} \quad \textbf{.......(4)}$$

4) Feature Point Descriptor:

A feature descriptor is created by first computing the gradient magnitude and orientation at each image sample point in a region around the feature point location, as shown on the left of Fig. 1.

These are weighted by a Gaussian window, indicated by the overlaid circle. These samples are then accumulated into orientation histograms summarizing the contents over 4x4 sub-regions, with 8 orientation bins. So each feature point has a 128-element feature as shown on the right, with the length of each arrow corresponding to the sum of the gradient magnitudes near that direction within the region.

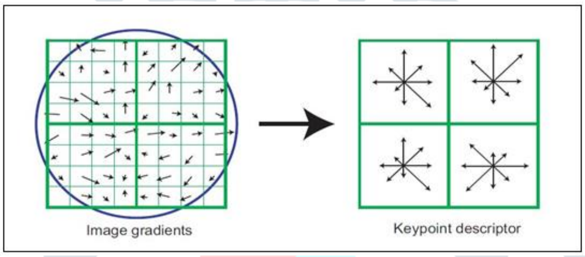

Fig. 1: Feature descriptor creation

III. PCA-SIFT

Y. Ke and R. Sukthankar (2004) introduced PCA (Principle component analysis)-SIFT which accepts the same input as the standard SIFT descriptor: the sub-pixel location, scale, and dominant orientations of the key point. We extract a 41×41 patch at the given scale, centered over the keypoint, and rotated to align its dominant orientation to a canonical direction. PCA-SIFT can be summarized in the following steps: (1) pre-compute an Eigen space to express the gradient images of local patches; (2) given a patch, compute its local image gradient; (3) project the gradient image vector using the Eigen space to derive a compact feature vector. This feature vector is significantly smaller than the standard SIFT feature vector, and can be used with the same matching algorithms. The Euclidean distance between two feature vectors is used to determine whether the two vectors correspond to the same key point in different images. Principal Component Analysis is a standard technique for dimensionality reduction and has been applied to a broad class of computer vision problems, including feature selection object recognition and face recognition. While PCA suffers from a number of shortcomings such as its implicit assumption of Gaussian distributions and its restriction to orthogonal linear combinations, it remains popular due to its simplicity. The idea of applying PCA to image patches is not novel.

3.1 Offline computation of patch Eigen space

PCA enables us to linearly-project high-dimensional samples onto a low-dimensional feature space. For our application, this projection (encoded by the patch Eigen space) can be pre-computed once and stored. As discussed above, the input vector is created by concatenating the horizontal and vertical gradient maps for the 41×41 patch centered at the key point. Thus, the input vector has 2×39×39=3042 elements. We then normalize this vector to unit magnitude to minimize the impact of variations in illumination. It is important to note that the 41×41 patch does not span the entire space of pixel values, nor the smaller manifold of patches drawn from natural images; it consists of the highly-restricted set of patches that passed through the first three stages of SIFT. More precisely, each of the patches satisfies the following properties: (1) it is centered on a local maximum in scale-space; (2) it has been rotated so that (one of its) dominant gradient orientations is aligned to be vertical; (3) it only contains information for the scale appropriate to this key point –i.e.,the 41×41 patch may have been created from a much larger region from the original image. The remaining variations in the input vector are mainly due to the “identity” of the keypoint (i.e.,the 3-D scene corresponding to this location) or to un-modeled distortions (such as perspective effects caused by changing camera viewpoint). It is not unreasonable to believe that these remaining variations can be reasonably modeled by low-dimensional Gaussian distributions, enabling PCA to accurately represent them with a compact feature representation. More importantly, projecting the gradient patch onto the low-dimensional space appears to retain the identity related variation while discarding the distortions induced by other effects.

3.2 Feature representation

To find the feature vector for a given image patch, we simply create its 3042-element normalized image gradient vector and project it into our feature space using the stored Eigen space. The standard SIFT representation employs 128-element vectors; using PCA-SIFT results in significant space benefits. As discussed above, we use the Euclidean distance between two feature vectors to determine whether the two vectors belong to the same keypoint in different images. This distance generates a binary decision, and adjusting this threshold enables one to select the appropriate trade-off between false positives and false negatives.

IV. Speeded-Up Robust Feature (SURF)

The search for discrete image point correspondences can be divided into three main steps.

First, `interest points' such as corners, blobs, and T-junctions are selected at distinctive locations in the image.

Next, using feature vector the neighborhood of every interest point is represented. This descriptor has to be distinctive and at the same time robust to noise, geometric and photometric deformations and detection displacements.

Finally, between deferent images the descriptor vectors are matched based on a distance between the vectors, e.g. the Euclidean distance.

According to H. Bay(2006 and 2008), SURF's detector and descriptor are not only faster, but the former is also more repeatable and the latter more distinctive. Bay,Ess and Tinne(2006 and 2008)had concluded that Hessian-based detectors are more stable and repeatable than their Harris based counterparts and observed that approximations like the DoG can bring speed at a low cost in terms of lost accuracy.

4.1 . Interest Point Detection

4.1.1. Integral Images

Integral Image or summed area tables is an intermediate representation of the image. It contains the sum of intensity values of all pixels in input image *I* within rectangular region formed by origin O=(0,0) and any point X=(x ,y). It provides fast computation of box type convolution filters.

$$\boldsymbol{I_{\Sigma}(X)} = \sum_{i=0}^{i\leq x}\sum_{j=0}^{j\leq y} \boldsymbol{I(i,j)} \ldots\ldots\ldots\ldots\ldots..\boldsymbol{(5)}$$

After computing integral image, only three operations (addition or subtraction) require for calculating sum of intensity values of pixels over any upright rectangular area.

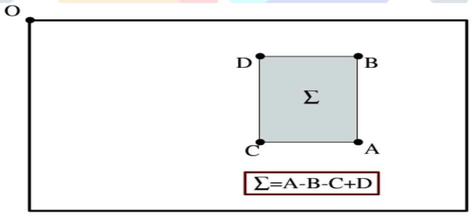

Fig. 2: Using integral images, it takes only three additions and four memory accesses to calculate the sum of intensities inside a rectangular region of any size.

4.1.2. Hessian Matrix Based Interest Points

The Hessian matrix $\mathcal{H}(\mathrm{X},\sigma)$ in X for a point X=(x,y) of an image I, at scale σ is defined as follows:

$$\boldsymbol{\mathcal{H}(X,\sigma)} = \begin{bmatrix} \mathbf{L_{xx}(X,\sigma)} & \mathbf{L_{xy}(X,\sigma)} \\ \mathbf{L_{xy}(X,\sigma)} & \mathbf{L_{yy}(X,\sigma)} \end{bmatrix} \ldots\boldsymbol{(6)}$$

Where $\mathrm{L_{xx}}(\mathrm{X},\sigma)$ is the convolution of the Gaussian second order derivative $\frac{\partial^2}{\partial x^2}g(\sigma)$ with the image I in point X and similarly for $\mathrm{L_{xy}}(\mathrm{X},\sigma)$ and $\mathrm{L_{yy}}(\mathrm{X},\sigma)$.

For scale-space analysis Gaussians are optimal, but in practice they have to be discretised and cropped. Under image rotations around odd multiples of $\frac{\pi}{4}$ this leads to a loss in repeatability. But in Hasian matrix due to the square shape of the filter repeatability is maximum around multiples of $\frac{\pi}{2}$.

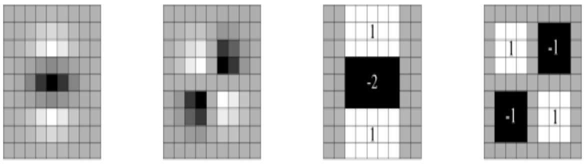

Fig. 3: Left to right: the (discretised and cropped) Gaussian second order partial derivative in y-(Lyy) and xy-direction (Lxy), respectively; our approximation for the second order Gaussian partial derivative in y- (Dyy) and xy-direction (Dxy). The grey regions are equal to zero.

The 9 x 9 box filters shown in figure 3, are approximations of a Gaussian with $\sigma = 1.2$ and represent the lowest scale. They will denoted by Dxx,Dyy and Dxy.

The Weights applied to the rectangular regions are kept simple for computational efficiency.

$$\mathbf{det}(\mathcal{H}_{approx}) = \boldsymbol{Dxx} * \boldsymbol{Dyy} - (\boldsymbol{wDxy})^{2} \text{ ...}\mathbf{(7)}$$

For the Hessian's determinant the, relative weight w of the filter responses is used to balance the expression.

Interest points need to be found at different scales because the search of correspondences often requires their comparison in images where they are seen at different scales. As an image pyramid, scale spaces are usually implemented. The images are smoothed with a Gaussian repeatedly and then sub-sampled for achieving a higher level of the pyramid. These pyramid layers are subtracted for getting the DoG (Difference of Gaussians) images where blobs and edges can be found.

Because use of box filters and integral images, there is no need to iteratively apply the same filter to the output of a previously filtered layer, but instead can apply box filters of any size directly on the original image at exactly the same speed and parallel.

Therefore, rather than iteratively reducing the image size the scale space is analyzed by up-scaling the filter size, figure 3.3. The output of the 9 x 9 filter is considered as the initial scale layer, to which we will refer as scale s = 1.2 (approximating Gaussian derivatives with$\sigma = 1.2$). The following layers are obtained by filtering the image with gradually bigger masks, taking into account the discrete nature of integral images and the specific structure of our filters.

Computational efficiency is main motivation for this type of sampling. Furthermore, there is no aliasing because we do not have to down sample the image. On the downside, box filters contains high-frequency components and in zoomed-out variants of the same scene that can get lost, which can limit scale-invariance.

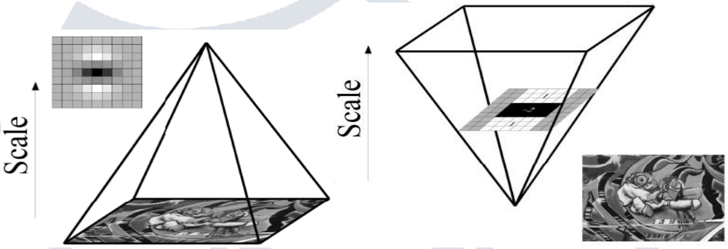

Fig. 4: Instead of iteratively reducing the image size (left), the use of integral images allows the up-scaling of the filter at constant cost (right).

The scale space is divided into levels of octaves. Each octave is obtained by convolving the same input image with a filter of increasing size and it represents a series of filter response maps. In total, an octave uses a scaling factor of 2, which implies that one needs to more than double the filter size. The scale space construction starts with the 9 x 9 filter, which calculates the blob response of the image for the smallest scale. Then, filters with sizes 15 x 15, 21 x 21, and 27 x 27 are applied, by which a scale change of more than 2 has been achieved. But this is needed, as a 3D non-maximum suppression is applied both spatially and over the neighboring scales. Since maxima are used for reasons of comparison only, the first and last Hessian response maps in the stack cannot contain such maxima themselves.

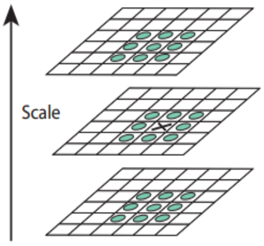

Fig. 5: Maxima and minima of images are detected by comparing a pixel (marked with X) to its 26 neighbors in 3x3 regions at the current and adjacent scales (marked with circles).

Similar considerations hold for the other octaves. For each new octave, the filter size is doubled (going from 6 to 12 to 24 to 48). At the same time, the sampling intervals for the extraction of the interest points can be doubled as well for every new octave. Due to this the computation time is reduced and the loss in accuracy is comparable to the image sub-sampling of the traditional approaches. The second octave is computed with filter sizes 15, 27, 39, 51. Similarly, a third octave is computed with the filter sizes 27, 51, 75, 99. The scale space analysis is performed for a fourth octave, using the filter sizes 51, 99, 147, and 195, if the original image size is still larger than the corresponding filter sizes.

4.2 Interest Point Description and Matching

4.2.1 Orientation Assignment

In order to be invariant to image rotation, we identify a reproducible orientation for the interest points. Due to this, we first calculate the Haar wavelet responses in x and y direction within a circular neighborhood of radius 6s around the interest point, with scale s at which the interest point was detected. The sampling step s is scale dependent. In keeping with the rest, also the size of the wavelets are scale dependent and set to a side length of 4s. Therefore, we can use integral images for fast filtering again. To compute the response in x or y direction at any scale only six operations are needed.

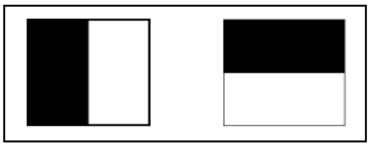

Fig. 6: Haar wavelet filters to compute the responses in x (left) and y direction (right). The dark parts have the weight -1 and the light parts +1

Once the wavelet responses are calculated and weighted with a Gaussian $\sigma = 2s$ centered at the interest point, the responses are represented as points in a space with the horizontal response strength along the abscissa and the vertical response strength along the ordinate. We calculate the sum of all responses within a sliding orientation window of size$\frac{\pi}{3}$ for estimating dominant orientation, see figure 6. The horizontal and vertical responses within the window are summed. The two summed responses then yield a local orientation vector. The orientation of the interest point can be defined by finding the longest such vector over all windows. The size of the sliding window is a parameter which had to be chosen carefully. Small sizes fire on single dominating gradients; large sizes tend to yield maxima in vector length that are not outspoken. Both of them result in a miss orientation of the interest point.

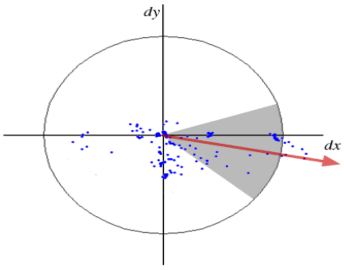

Fig. 7: Orientation assignment: A sliding orientation window of size $\frac{\pi}{3}$detects the dominant orientation of the Gaussian weighted Haar wavelet responses at every sample point within a circular neighborhood around the interest point.

4.2.2 Descriptor based on Sum of Haar Wavelet Responses

For the extraction of the descriptor, square region cantered around the interest point constructed and oriented along the orientation selected in the previous section. The size of this window is 20s. Examples of such square regions are illustrated in fig. 8.

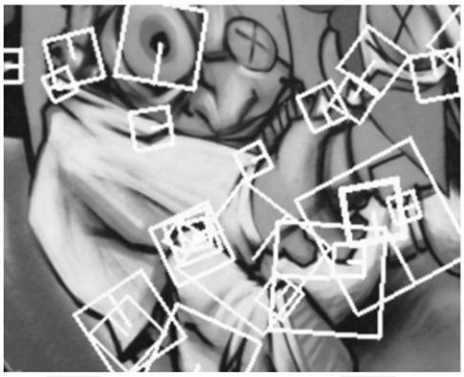

Fig. 8: Detail of the Graffiti scene showing the size of the oriented descriptor window at different scales.

The region is split up into smaller 4 x 4 square sub-regions regularly. This preserves important spatial information. For each sub-region, we compute Haar wavelet responses at 5 x 5 regularly spaced sample points. For reasons of simplicity, we call dx and dy, the Haar wavelet response in horizontal direction and dy the Haar wavelet response in vertical direction respectively (filter size 2s), see figure 9 again.

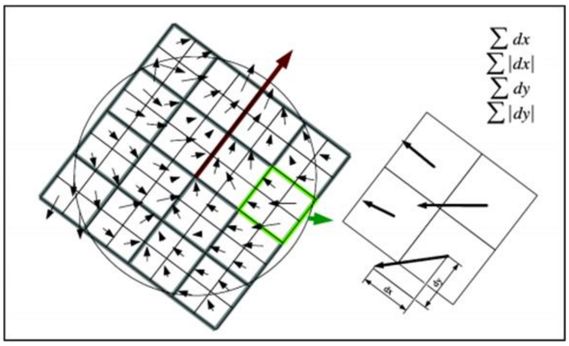

Fig. 9: To build the descriptor, an oriented quadratic grid with 4 x 4 square sub-regions is laid over the interest point (left). For each square, the wavelet responses are computed. The 2 x 2 sub-divisions of each square correspond to the actual fields of the descriptor. These are the sums dx, |dx|, dy and |dy| computed relatively to the orientation of the grid (right).

“Horizontal" and“Vertical" here is defined in relation to the selected interest point orientation (see figure 9).To increase the robustness towards geometric deformations and localization errors, the responses dx and dy are first weighted with a Gaussian ($\sigma = 3.3s$) centered at the interest point.

Then, the wavelet responses dx and dy are summed up over each sub-region and form a first set of entries in the feature vector. In order to bring in information about the polarity of the intensity changes, we also extract the sum of the absolute values of the responses, |dx| and |dy|. Hence, each sub-region has a four-dimensional descriptor vector V for its underlying intensity structure $V = (\sum d_x, \sum d_y, \sum|d_x|, \sum|d_x|)$. Concatenating this for all 4 x 4 sub-regions, this results in a descriptor vector of length 64. The wavelet responses are invariant to a bias in illumination (offset). Invariance to contrast (a scale factor) is achieved by turning the descriptor into a unit vector.

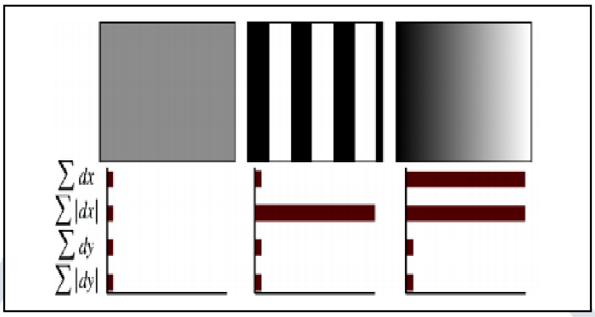

Fig. 10. The descriptor entries of a sub-region represent the nature of the underlying intensity pattern. Left: In case of a homogeneous region, all values are relatively low. Middle: In presence of frequencies in x direction, the value of $\sum|d_x|$ is high, but all others remain low. If the intensity is gradually increasing in x direction, both values $\sum d_x$ and $\sum|d_x|$ are high.

Figure 10.shows the properties of the descriptor for three distinctively different image intensity patterns within a sub-region. One can imagine combinations of such local intensity patterns, resulting in a distinctive descriptor.

4.3 Color SURF

J. fu; X. jing; S. Sun (2013) introduced Color SURF algorithm not only keeps the advantages of the pure gray-based geometric description but also adds color information to improve the distinctiveness of the featureset. P. fan; A. men; M. Chen (2013) found color descriptor with using of local kernel color histogram.

The detailed procedures of this method are as follows:

1) Extracting the scale space feature of interest points by using the Fast-Hessian matrix
2) Finding the location as well as scale of the interest points.
3) Assigning Orientation.
4) Adding color information to SURF descriptor, so that the descriptor describes not only the distribution of Harr-wavelet responses but also the color information in RGB format.

The first three stages are the same as SURF. The last stage is described as follows:

For the extraction of the descriptor, firstly, constructing a square region centered on the interest point, the size of this window is 20*s*. This region is split up regularly into smaller 4×4 square sub-regions. Secondly, it is oriented along the orientation selected in the previous section. Thirdly, for each sub-region, the factors calculated in the SURF descriptor are also calculated here. What's more, we calculate three factors namely $\sum r\,(x,y)$, $\sum g(x,y)$, $\sum b(x,y)$ for each sub-region.

For each interest point *P*(*x*, *y*) color value is represented by *r*(*x*, *y*),*g*(*x*, *y*), *b*(*x*, *y*). Then, in every sub-region, *r* (*x*, *y*),*g* (*x*, *y*), *b*(*x*, *y*) of every pixel are summed up to forma first set of entries to the feature vector.$\sum r\,(x,y), \sum g(x,y), \sum b(x,y)$ are weighted with three different factors, α, β, γ. They have different values when the matching images are of different rotation angles, and are set according to a large number of tests. The wavelet responses are invariant to a bias in illumination. Finally, the descriptor vector for each sub-region can be described as:

$$\boldsymbol{V_{sub}} = (\sum \boldsymbol{dx}, \sum \boldsymbol{dy}, \sum|\boldsymbol{dx}|, \sum|\boldsymbol{dy}|, \sum\frac{\boldsymbol{r}}{\boldsymbol{\alpha}}, \sum\frac{\boldsymbol{g}}{\boldsymbol{\beta}}, \sum\frac{\boldsymbol{b}}{\boldsymbol{\gamma}})\ldots\mathbf{(8)}$$

V. DISCUSSION

From survey of all papers, we found that SIFT‘s matching success attributes to that its feature representation has been carefully designed to be robust to localization error. SIFT shows its stability in scale, rotation and blur.

PCA is known to be sensitive to registration error. Using a small number of dimensions provides significant benefits in storage space and matching speed. PCA-SIFT need to improve blur and scale performances.

SURF shows its stability and fast speed in the experiments. It is known that ‘Fast-Hessian’ detector that used in SURF is more than 3 times faster that DOG (which was used in SIFT) and 5 times faster than Hessian-Laplace. SURF looks fast and good in most situations, but when the rotation is large, it also needs to improve this performance.

VI. ACKNOWLEDGEMENTS

I would like to thank my guides, Ms. Darshana Mistry (CSE) and Yatin Patel (CE), for their leadership and valuable advice throughout my research work.

REFERENCES

[1] D. G. Lowe(2004), “Distinctive image features from scale-invariant key points”, International journal of Computer Vision, Vol-60, Issue-2, pp.91-110.

[2] H. Bay, T. Tuytelaars, and L. V. Gool (2006), “SURF: Speeded Up Robust Features”, Computer Vision–ECCV.

[3] H. Bay , A. Ess , T. Tuytelaars , and L. V. Gool(2008), “Speeded-Up Robust Features”, Vol. 110, No. 3, pp. 346--359, June 2008.

[4] L. Juan, O. Gwun(2009), “A comparison of SIFT, PCA-SIFT and SURF”, International Journal of Image Processing (IJIP) Volume(3), Issue(4), pp. 143-152.

[5] N.NagaRaju, T.Satyasavitri, Ch.A.Swamy, “Image Registration Using Scale Invariant Feature Transform”, International Journal of Scientific Engineering and Technology, Volume No.2, Issue No.7, pp : 675-680 , July 2013

[6] K. MIKOLAJCZYK AND C. SCHMID, “Scale & Affine Invariant Interest Point Detectors”, International Journal of Computer Vision 60(1), 63–86, 2004.

[7] T. LINDEBERG, “Feature Detection with Automatic Scale Selection”, International Journal of Computer Vision 30(2), 79–116 (1998).

[8] N. Hamid, A. Yahya, R. Ahmad, O. Al-Qershi(2012), “A Comparison between Using SIFT and SURF for CharacteristicRegion Based Image Steganography”, IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 3, No 3, May 2012

[9] J. fu; X. jing; S. Sun (2013), *“C-SURF: Colored Speeded up Robust Features”*,International Standard Conference Trustworthy Computing and Services 2012, pp. 203–210, 2013.

[10] P. fan; A. men; M. Chen (2013),“Color –SIRF: A SURF Descriptor with Local Kernel Color Histogram”,IEEE,2009,pp.726-730.

[11] Y. Ke and R. Sukthankar(2004). “PCA-SIFT: A More Distinctive Representation for Local Image Descriptors”,Proc. Conf. Computer Vision and Pattern Recognition,pp. 511-517.