SIFT and SURF Performance Evaluation Against Various Image Deformations on Benchmark Dataset

Nabeel Younus Khan
Computer Science Department
Otago University
Dunedin, New Zealand
Email: nabeel@cs.otago.ac.nz

Brendan McCane
Computer Science Department
Otago University
Dunedin, New Zealand
Email: mccane@cs.otago.ac.nz

Geoff Wyvill
Computer Science Department
Otago University
Dunedin, New Zealand
Email:geoff@cs.otago.ac.nz

***Abstract*—Scene classification in indoor and outdoor environments is a fundamental problem to the vision and robotics community. Scene classification benefits from image features which are invariant to image transformations such as rotation, illumination, scale, viewpoint, noise etc. Selecting suitable features that exhibit such invariances plays a key part in classification performance.**

This paper summarizes the performance of two robust feature detection algorithms namely Scale Invariant Feature Transform (SIFT) and Speeded up Robust Features (SURF) on several classification datasets. In this paper, we have proposed three shorter SIFT descriptors. Results show that the proposed 64D and 96D SIFT descriptors perform as well as traditional 128D SIFT descriptors for image matching at a significantly reduced computational cost. SURF has also been observed to give good classification results on different datasets.

***Keywords*-SIFT; SURF; Viewpoint; Scale; Kd trees.**

I. Introduction

Images of one scene may be taken from different viewpoints or may suffer transformations such as rotation, noise etc. So it is likely that two images of the same scene will be different. The task of finding similarity correspondences between two images of the same scene or object has thus become a challenging problem in a number of vision applications. Such applications range from image registration, camera calibration, object recognition, scene localization in navigation systems, image retrieval based search engines etc.

For image matching, extraction of such information (i.e. features) is required from the images which can provide reliable matching between different viewpoints of the same image. Feature detection occurs within an image and seeks to describe only those parts of that image where we can get unique information or signatures (i.e. feature descriptors). During training, feature descriptors are extracted from sample images and stored. In classification, feature descriptors of a query image are then matched with all trained image features and the trained image giving maximum correspondence is considered the best match. Feature descriptor matching can be based on distances such as Euclidean, Mahalanobis or distance ratios.

The search for distinctive features from images is divided into two main phases. First, "keypoints" are extracted from distinctive locations from the images such as edges, blobs etc. Keypoint detectors should be highly repeatable. Next, neighborhood regions are picked around every keypoint and distinctive feature descriptors are computed from each region.

A variety of feature detection algorithms have been proposed in the literature to compute reliable descriptors for image matching [1], [2], [3], [4], [5], [6], [7], [8]. SIFT and SURF descriptors are the most promising due to good performance and have now been used in many applications.

A thorough comparison of many feature descriptors was reported in [12] which concluded that overall SIFT outperforms other detectors. However, SURF was not included in the comparisons and although it has been claimed to be superior to SIFT by the proposers of SURF [6], this has not been independently verified by other researchers on large datasets, although it has been done on several small datasets [13], [14].

In this paper, we offer a substantive evaluation of SIFT and SURF on several large sets of images and further test each algorithm on typical image transformations such as rotation, scale, blurring and brightness variance. We also compare three shorter SIFT descriptors on these datasets.

In section 2, we briefly discuss the working mechanism of SIFT and SURF followed by discussion of our proposed shorter SIFT descriptors. In section 3, we throughly compare matching performance of all descriptors on standard benchmark datasets. We also evaluate all descriptors performance against possible image transformations. The article is concluded in section 4.

II. Methods

A. SIFT Detector

The SIFT detector has four main stages namely, scale-space extrema detection, keypoint localization, orientation computation and keypoint descriptor extraction [5].

The first stage uses Difference of Gaussians (DoG) to identify the potential keypoints. Several Gaussian blurred

978-0-7695-4588-2/11 $26.00 © 2011 IEEE
DOI 10.1109/DICTA.2011.90

images at different scales are produced from the input image and DoGs are computed from neighbours in scale space. In the second stage, candidate keypoints are located by finding extrema in the DoG images that are locally extremal in space and scale. Spatially unstable keypoints are eliminated by thresholding against the ratio of eigenvalues of the Hessian matrix (unstable edge keypoints have a high ratio, and stable corner keypoints have a low ratio), low contrast keypoints are eliminated and the remaining keypoints are localised by interpolating across the DoG images. The third stage assigns a principal orientation to each keypoint.

The final phase computes a highly distinctive descriptor for each keypoint. In order to achieve orientation invariance, the descriptor coordinates and gradient orientations are rotated relative to the key point orientation. For every keypoint, a set of orientation histograms are created on 4x4 pixel neighborhoods with 8 bins each (using magnitudes and orientation of samples in 16 x 16 region around the keypoint). The resulting feature descriptor will be a vector of 128 elements that is then normalized to unit length to handle illumination differences. Descriptor size can be varied, however best results are reported with 128D SIFT descriptors [5]. SIFT descriptors are invariant to rotation, scale, contrast and partially invariant to other transformations.

The SIFT descriptor size is controlled by its width i.e. the array of orientation histograms (n x n) and number of orientation bins in each histogram (r). The size of resulting SIFT descriptor is rn^2 [5]. The value of n affects the window size around the keypoint as we use 4 x 4 region to capture pattern information e.g. for $n = 3$, we will use a window of size 12 x 12 around the keypoint. Various sizes were analyzed in [5] and it was reported that 128D SIFT is superior in terms of matching precision, i.e. $n = 4$ and $r = 8$. Most other work has used standard 128D SIFT features while very few have tried smaller SIFT descriptors for small scale works e.g. 36D SIFT features from 3 x 3 subregions, each with 4 orientation bins, with few target images are used in [18].

Smaller sized descriptors use less memory and result in faster classification but precision rates may be affected. No research article has investigated the classification performance of SIFT descriptors of size other then 128.

B. Proposed changes in existing SIFT

We have implemented our own SIFT algorithm and have proposed not to double the input image in the first stage as it results in generation of a huge number of trained features. In traditional 128D SIFT, a 16 x 16 window is used around the detected keypoint. That window is divided into sixteen 4 x 4 regions. From every region, we compute the gradient information and summarize that into 8 bin orientation histograms. Gradients far away from the keypoint are given less weight compared to near ones. Magnitudes are Gaussian weighted based on distance before being adding to the corresponding orientation bins. Finally we get a 4 x 4 array of orientation histograms each one having 8 values resulting in 128D SIFT descriptor as shown in Fig. 1.

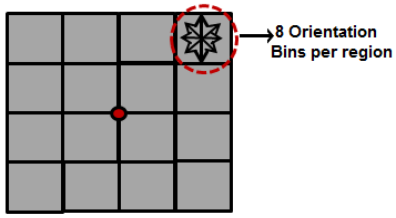

Figure 1. 4 x 4 computed orientation histogram arrays in 128D SIFT.

Reduced size SIFT descriptors can be generated by skipping orientation values from some regions of the 4 x 4 array. Different square arrays of orientation histograms have been tested in [5], but other choices are possible. We have tested three other choices all shown in Figure 2:

1) 96D SIFT: Ignore corner regions.
2) 64D SIFT: Ignore corner regions and average neighboring outside regions.
3) 32D SIFT: Use central 2 x 2 block.

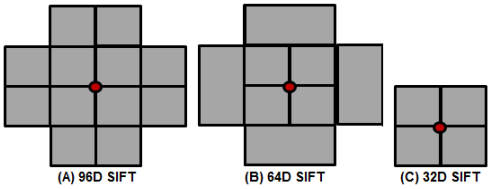

Figure 2. Customized 4 x 4 orientation histogram array configurations used to generate shorter SIFT descriptors.

C. SURF Detector

SURF, also known as approximate SIFT, employs integral images and efficient scale space construction to generate keypoints and descriptors very efficiently. SURF uses two stages namely keypoint detection and keypoint description [6]. In the first stage, rather than using DoGs as in SIFT, integral images allow the fast computation of approximate Laplacian of Gaussian images using a box filter. The computational cost of applying the box filter is independent of the size of the filter because of the integral image representation. Determinants of the Hessian matrix are then used to detect the keypoints. So SURF builds its scale space by keeping the image size the same and varies the filter size only.

The first stage results in invariance to scale and location. In the final stage, each detected keypoint is first assigned a reproducible orientation. For orientation, Haar wavelet responses in x and y directions are calculated for a set of pixels within a radius of 6σ where σ refers to the detected keypoint scale. The SURF descriptor is then computed by constructing a square window centered around the keypoint

and oriented along the orientation obtained before. This window is divided into 4 x 4 regular sub-regions and Haar wavelets of size 2σ are calculated within each sub-region. Each sub-region contributes 4 values thus resulting in 64D descriptor vectors which are then normalized to unit length. The resulting SURF descriptor is invariant to rotation, scale, contrast and partially invariant to other transformations. Shorter SURF descriptors can also be computed however best results are reported with 64D SURF descriptors [6].

We have used the OpenSURF implementation [20] and use k-d trees to speed up nearest neighbor matching. 64D SURF feature descriptors are extracted and classification is performed in the same way as we do in SIFT. The default threshold used in the supplied code is $(d_1/d_2 < 0.65)$ to check the correspondences where d_1 and d_2 refer to query and trained image vectors. The threshold is kept the same in all experiments.

III. Results

A. Evaluation Method

Performance is measured using an image matching task on a number of datasets using a naive matching algorithm. Keypoints and feature descriptors are extracted from each of the images in the dataset and all descriptors are inserted into a k-d tree [21]. To perform image classification, all nearest neighbors of the features in the query image are found in the image collection. If the nearest neighbor is within a threshold in feature space, then a feature correspondence is recorded. The matched image is selected as that image with the most feature correspondences from the collection. We have identified distance thresholds for all SIFT descriptors by observing the similarity variance between different feature descriptors. We have used 170, 160, 150 and 90 distance thresholds respectively for our 128D, 96D, 64D and 32D SIFT descriptors during image matching. These thresholds work well in the classification of images.

B. Data Sets

All features and descriptors have been evaluated for image matching on different benchmark datasets [19], [22], [23], [24]. Samples from each datasets are shown in Figure 3.

1) David Nister Dataset: The dataset described in [19] contains 4 different images of 2500 objects i.e. 10,000 images in total. The dataset contains a variety of indoor and outdoor environmental images and is used as a standard to test the robustness of classification algorithms. To test the classification performance, we have selected the first 500 object images from the dataset (i.e. first 2000 images are picked). The first image of every object is used for testing while the remaining three images have been used for training (i.e. 500 test and 1500 trained images). Approximately 0.47M SIFT and 0.62M SURF trained features are extracted from the training set.

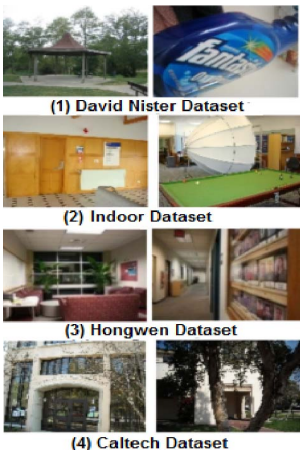

Figure 3. Sample images from standard Benchmark Datasets

2) Indoor Dataset: The dataset described in [22] contains indoor images taken from the Computer Science building at Anonymous University. It is a standard office-type building with some classroom size computer laboratories — many different locations within the building look very similar. The dataset covers about 30 indoor locations and contains about 700 images. The dataset is challenging because it contains many similar looking images. We have used 630 and 70 images for training and testing with about 0.17M SIFT and 0.42M SURF extracted trained features. At least two query images are picked from every indoor location resulting in a total of about 70 images.

3) Hongwen Dataset: The dataset reported in [23] contains 8000 indoor images covering one floor of a building taken over a period of time. We have used 2250 and 100 images for training and testing with 12M SIFT and 14M SURF extracted trained features. 100 test images have been provided separately in the used dataset.

4) Caltech Dataset: The dataset reported in [24] contains images for 50 building exteriors around the Caltech campus (California Institute of Technology). Five different images are taken for each building from different angles and distances resulting in 250 images. The first image of every building is used for testing while the remaining four are used for training with about 0.08M SIFT and 0.2M SURF trained features.

C. Image Transformations

We have evaluated matching performance of all proposed descriptors against different image transformations on the David Nister dataset [19]. We have used 500 images of different scenes to test the invariance characteristics of features.

1) Rotation Invariance: To test rotation invariance, the first 500 object images from the dataset are picked and used for testing. These test images are rotated at different

angles in a clockwise direction to generate 500 trained images. $imrotate$ function of MATLAB with 'bilinear' interpolation is used for performing rotations. Results have been summarized in Figure 4 and show that SIFT (32D, 64D, 96D and 128D) and SURF are rotation invariant.

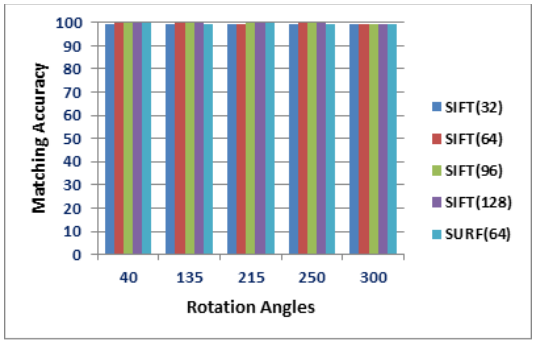

Figure 4. SIFT vs SURF Performance on Rotated Images

2) Illumination Invariance: To test illumination invariance, the same first 500 object images have been used from the dataset. These test images are used to generate trained images for classification.

To test the brightness, 500 test images are brightened by adding or subtracting a number of pixel value offsets (i.e. every pixel's red, green and blue channel intensities are incremented equally). The sum is clamped to be within 0-255. The offsets tested are: $50, 70, 100, 120, -30, -50, -70, -90$. Results are summarized in Figure 5 and Figure 6 and show that SIFT (64D, 96D and 128D) and SURF are invariant to illumination and perform well. 32D SIFT does not perform well on severely darkened images but does well on brightened images.

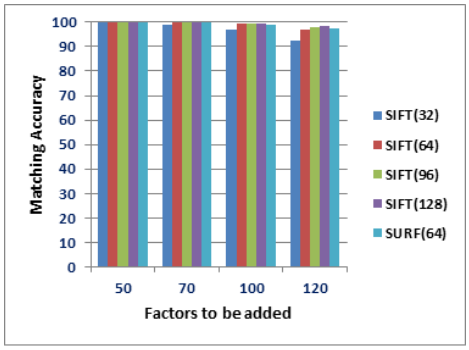

Figure 5. SIFT vs SURF Performance on Brightened Images

3) Noise Invariance: To test noise invariance, we have used the $imnoise$ function of MATLAB and have applied three types of noise to the test images: Gaussian, salt and pepper and speckle to generate the trained data. Image pixels are first scaled to 0-1 before applying noise. We have added Gaussian white noise with $\sigma^2 = 0.1$ and $\sigma^2 = 0.2$, salt and pepper noise with density of 15% and 35%, and multiplicative white noise with mean 0 and $\sigma^2 = 0.04$. Results in Figure 7 show that detectors degrade most noticably with salt and pepper noise and this effects smaller sized detectors more severely than larger detectors.

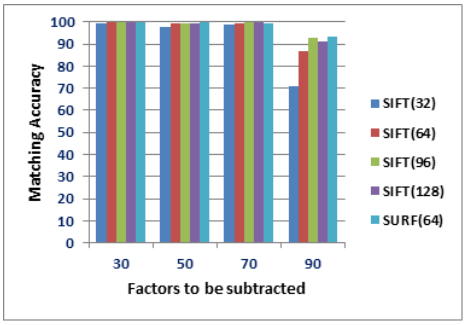

Figure 6. SIFT vs SURF Performance on Darkened Images

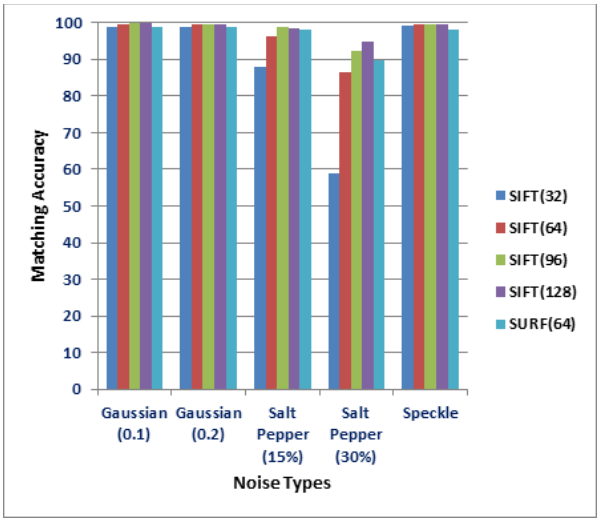

Figure 7. SIFT vs SURF Performance on Noisy Images

D. Blurring

The $fspecial$ function of MATLAB is used to generate a Gaussian filter (20 x 20) and filtering of the image is then done using $imfilter$ function. Experiments have tested various values of Gaussian blur i.e. $\sigma = 5$, $\sigma = 10$ and $\sigma = 20$. Figure 8 clearly shows that all SIFT and SURF handle blurring well. However for a very large amount of blurring i.e. $\sigma = 20$, trained images become too blurry. In such cases, 32D SIFT, and SURF perform worst, and it is not surprising that larger detector sizes are less affected by high blurring factors.

1) Scale: To test scaling, we selected 500 objects from the dataset for which there were two images available at different scales (i.e. a total of 1000 images). We did not

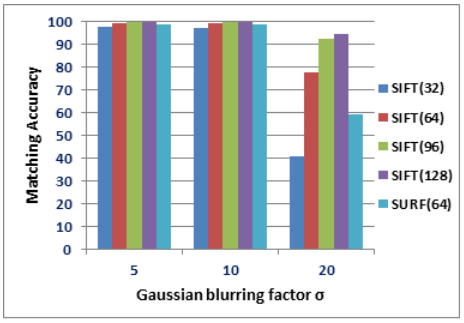

Figure 8. SIFT vs SURF Performance on Blurred Images

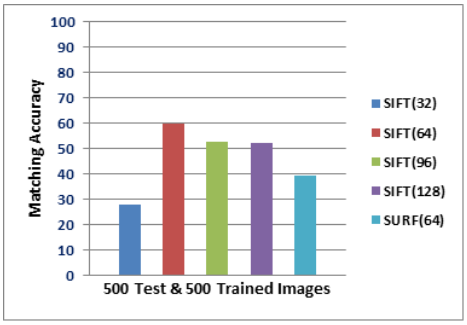

Figure 10. SIFT vs SURF Performance on images of different viewpoints

select the first 500 object images as there were no scaled images available in most of the cases. All selected object images have a scale transformation while some have additional smaller viewpoint transformations as well. The first image has been used for training while the other is used for testing in both SIFT and SURF. Results in Figure 9 show that for image scaling, 64D, 96D and 128D SIFT outperform SURF. 32D SIFT underperforms but still performs comparable to SURF which indicates that SIFT has excellent invariance to scaling even with smaller descriptors.

Figure 9. SIFT vs SURF Performance on Scaled Images

2) Viewpoint: To test viewpoint, we selected 500 objects from the dataset for which there were two images available in the dataset taken from different viewpoints i.e. a total of 1000 images. The viewpoint angles vary for every object. The first object image is used for training while the other is used for testing. Results in Figure 10 show that all descriptors perform worse than in other experiments, but 64D, 96D and 128D SIFT still outperform SURF. Interestingly, 64D SIFT produces the best performance.

E. General Matching

In the final experiment, we have evaluated the classification performance of all descriptors using the four real world datasets described in Section III-B. In all cases, the test and training images are independent.

We have used 1500 images for training and 500 images for testing in David Nister dataset (0.47M SIFT and 0.62M SURF extracted trained features), 630 training and 70 test images for Indoor dataset (0.17M SIFT and 0.42M SURF features), 2250 training and 100 test images for Hongwen Dataset (12M SIFT and 14M SURF features) and 200 training and 50 test images for CALTECH dataset (0.08M SIFT and 0.2M SURF features). Classification results are shown in Figure 11 which show that all SIFT and SURF descriptors perform well on all datasets except for 32D SIFT.

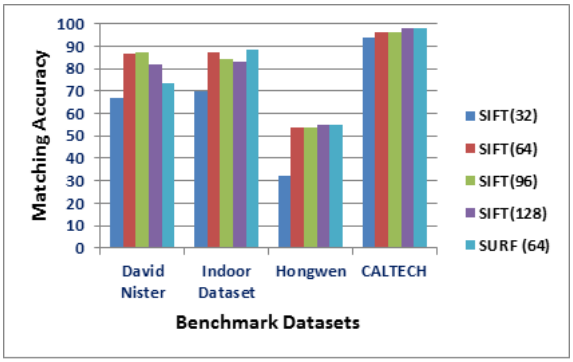

Figure 11. SIFT vs SURF Classification Performance on Benchmark Datasets

Execution time for each of the methods is shown in Table I. As expected, using smaller descriptors in SIFT methods results in significant speedups which can be important for some applications. Average matching time is surprisingly high for SURF. Partly because SURF produces many more features and partly because of the implementation [20].

Table I
AVERAGE MATCHING TIME REQUIRED TO MATCH ONE QUERY IMAGE BY ALL DESCRIPTORS.

	128D SIFT	96D SIFT	64D SIFT	32D SIFT	SURF
Time	59 sec	33 sec	18 sec	11 sec	80 sec

IV. DISCUSSION AND CONCLUSION

In this paper we have reported a thorough analysis. Most descriptors performed reasonably well except for 32D SIFT which has underperformed. The reason is that 32D SIFT

descriptors are very small and thus fail to capture sufficient pattern information. This leads to larger wrong matches during classification. The important findings of this paper are as follows:

1) 128D, 64D and 96D SIFT are similar in matching accuracy across most tests.
2) 64D SIFT is superior for matching images with different viewpoints and this is likely due to the smaller window size which minimizes occlusion effects.
3) SURF is as good as SIFT on most tests except for scaling, large blur and viewpoint invariance.
4) On real image datasets there is little to separate the different SIFTs (excluding 32D SIFT) and SURF except for efficiency.
5) The proposed 64D SIFT should be preferred in most future applications as it is as accurate as 128D SIFT, but it also offers almost three times faster image matching and half the memory requirements.

REFERENCES

[1] T. Linderberg, *Feature Detection with automatic scale selection*, International journal of Computer Vision, Vol. 30, pp. 79-116, 1998.

[2] K. Mikolajczyk and C. Schmid, *An affine invariant interest point detector*, Proc. European Conference on Computer Vision, pp. 128-142, 2002.

[3] T. Tuytelaars and L. Van Gool, *Wide baseline stereo based on local, affinely invariant regions*, Proc. British Machine Vision Conference, pp. 412-425, 2000.

[4] J. Matas, O. Chum, M. Urban and T. Padjla, *Robust wide baseline stereo from maximally stable extermal regions*, Proc. British Machine Vision Conference, Vol. 1, pp. 384-393, 2002.

[5] D. Lowe, *Distinctive Image features from scale invariant keypoints*, International journal of Computer Vision, Vol. 60, pp. 91-110, 2004.

[6] H. Bay, T. Tuytelaars and L. Van Gool, *SURF: Speeded Up Robust Features*, Proc. European Conference on Computer Vision, Vol. 110, pp. 407-417, 2006.

[7] G. Carneiro and A.D. Jepson, *Multi scale phase based local features*, Proc. International Conference on Computer Vision and Pattern Recognition, Vol. 1, pp. 736-743, 2003.

[8] F. Schaffalitzky and A. Zisserman, *Multi view image matching for unordered image sets*, Proc. European Conference on Computer Vision, Vol. 1,pp. 414-431, 2002.

[9] C. Harris and M. Stepehens, *A combined corner and edge detector*, Proc. Alvey Vision Conference, pp. 147-151, 1998.

[10] E. Rosten, R. Porter and T. Drummond, *FASTER and better: A machine learning approach to corner detection*, IEEE Trans. Pattern Analysis and Machine Intelligence, Vol. 32, pp.105-119, 2010.

[11] T. Kadir and M. Bardy, *Scale, saliency and image description*, International journal of Computer Vision, Vol. 45, pp.83-105, 2001.

[12] K. Mikolajczyk, T. Tuytelaars, C Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir and L. Van Gool, *A Comparison of Affine Region Detectors*, International journal of Computer Vision, Vol. 65, pp. 43-72, 2005.

[13] L. Juan and O. Gwun, *A Comparison of SIFT , PCA-SIFT and SURF*, International Journal of Image Processing, Vol. 65, pp. 143-152, 2009.

[14] J. Bauer, N. Sunderhauf and P. Protzel, *Comparing several implementations of two recently published feature detectors*, Proc. International Conference on Intelligent and Autonomous Systems, 2007.

[15] Y. Ke and R. Sukthankar, *PCA-SIFT a more distinctive representation for local image descriptors*, Proc. International Conference on Computer Vision and Pattern Recognition, Vol. 2, pp. 506-513, 2004.

[16] P.A. Viola and M.J. Jones, *Rapid Object Detection using a boosted cascade of simple features*, Proc. International Conference on Computer Vision and Pattern Recognition, Vol. 1, pp. 511-518, 2001.

[17] Y. Zhann-Long and G. Baco-Long,*Image based mosaic based on SIFT*, Proc. International Conf. on Intelligent Information Hiding/ Multimedia Signal Processing, pp. 1422-1425, 2008.

[18] W. Daniel, R. Gerhard, M. Alessandro, D. Tom and S. Dieter, *Pose Tracking from Natural Features on Mobile Phones*, Proc. International Symposium on Mixed and Augmented Reality, pp. 125-134, 2008.

[19] D. Nister and H. Stewenius, *Scalable recognition with a vocabulary tree*, Proc. International Conference on Computer Vision and Pattern Recognition,Vol. 2, pp.2161-2168, 2006. Available from http://www.vis.uky.edu/~stewe/ukbench/.

[20] C. Evans, *Notes on the OpenSURF Library*, University of Bristol (UK), 2009. Available from http://www.chrisevansdev.com.

[21] A. Moore, *An introductory tutorial on kd trees*, Technical Report No. 209, University of Cambridge, 1991.

[22] N. Khan, *Indoor Environmental Images (Computer Science Department)*, Otago University (NZ), 2010. Available from http://www.cs.otago.ac.nz/pgdweb/nabeel/Downloads/Dataset.zip.

[23] H. Kang, A. Efros, M. Hebert and T. Kanade, *Image Matching in Large Scale Indoor Environment*, Proc. International Conference on Computer Vision and Pattern Recognition Workshop, pp. 33-40, 2009. Available from http://www.hwkang.com/.

[24] M. Aly, P. Welinder, M. Munich and P. Perona, *Towards Automated Large Scale Discovery of Image Families*, Proc. International Conference on Computer Vision and Pattern Recognition Workshop, 2011. Available from http://www.vision.caltech.edu/malaa/datasets/caltech-buildings/