Chapter 24

IMAGE BACKGROUND MATCHING FOR IDENTIFYING SUSPECTS

Paul Fogg, Gilbert Peterson and Michael Veth

Abstract Thousands of digital images may exist of a given location, some of which may show a crime in progress. One technique for identifying suspects and witnesses is to collect images of specific crime scenes from computers, cell phones, cameras and other electronic devices, and perform image matching based on image backgrounds. This paper describes an image matching technique that is used in conjunction with feature generation methodologies, such as the Scale Invariant Feature Transform (SIFT) and the Speeded Up Robust Features (SURF) algorithms. The technique identifies keypoints in images of a given location with minor differences in viewpoint and content. After calculating keypoints for the images, the technique stores only the "good" features for each image to minimize space and matching requirements. Test results indicate that matching accuracy exceeding 80% is obtained with the SIFT and SURF algorithms.

Keywords: Image background matching, SIFT, SURF, keypoint reduction

1. Introduction

Electronic matching is commonly performed for fingerprints [5], shoe imprints [1] and facial features [13]. Image feature generation techniques, such as the scale invariant feature transform (SIFT) [7] and speeded up robust features (SURF) [2] algorithms can be used to automate the process of digital image matching. Persons of interest can be identified by grouping and matching multiple images of a crime scene, even when the images are taken from different viewpoints. For example, crime scene images can be used to identify and place suspects and victims at the scene. Alternatively, background details from child pornography images can be used to establish where the pictures were taken.

This paper describes a technique for image matching that is used in conjunction with the scale invariant feature transform (SIFT) and speeded up robust features (SURF) algorithms. The first step involves the generation of keypoints for each algorithm. The next step reduces the number of keypoints to minimize storage requirements and improve matching speeds. The third step performs match comparison, which removes poor quality keypoint matches. The final step analyzes images taken of the same location to identify features and/or persons of interest. Testing indicates that better than 80% matching accuracy is achieved using the SIFT and SURF algorithms.

2. Image Matching Algorithms

This section provides an overview of several image matching algorithms, including the Scale Invariant Feature Transform (SIFT) [7, 8] and Speeded Up Robust Features (SURF) [2] algorithms.

2.1 SIFT Algorithm

The SIFT algorithm [7] performs image recognition by calculating a local image feature vector. The feature vector is used for matching scaled, translated and/or rotated images under low illumination and affine transformations. This technique is inspired by neuronal activities in the inferior temporal cortex of primates, which implement object recognition.

The SIFT algorithm uses four steps to extract image keypoints: scale-space extrema detection, keypoint localization, orientation assignment and keypoint descriptor generation [8].

1. Scale-Space Extrema Detection: In this step, Gaussian kernels of increasing variance are convolved with the image. A total of $s + 3$ images are produced (s is the number of scales); each image has an increased amount of blur. Next, the difference of Gaussians is computed for each pair of blurred images by subtracting each image from the next most blurred image; this produces $s + 2$ differences of Gaussians. Each difference of Gaussians is then bilinearly interpolated to generate the next reduced scale for the total of s scales.

2. Keypoint Localization: Each pixel in a difference of Gaussians is compared with its eight neighbors. A pixel is designated as a keypoint if it is a maximum or minimum at this level and the related pixels at all other scales are also maxima or minima. An improve-

ment to this technique proposed by Lowe [8] fits a 3D quadratic function to the pixels and their neighbors across scales.

3. Orientation Assignment: For each keypoint, the Gaussian blurred image with a value closest to the scale of the keypoint is selected. In this image, the gradient magnitude and orientation of the image are calculated over 36 bins around the keypoint pixel. These 36 vectors, which are weighted by the keypoint scale, identify the orientation of the keypoint.

4. Keypoint Descriptor Generation: The keypoint descriptor is determined by calculating the gradient magnitude and orientation of each pixel in a 16×16 pixel patch around the keypoint. These vectors are weighted by a Gaussian distribution centered at the keypoint and are combined in 4×4 pixel patches. The 16 combined gradients are reduced to eight vectors in each of the cardinal directions. The magnitudes of these vectors become the 128-element keypoint descriptor.

Lowe [8] identified a marked decrease in matching performance for 112 images as the number of keypoints approaches 100,000 per image. However, the effect of a reduction in the number of keypoints per image on matching performance has not been investigated. This is an important issue because a child pornography case, for example, may have tens of thousands of images; an average of 3,000 keypoints per image results in more than 30,000,000 keypoints. Our strategy is to reduce the number of keypoints per image (which saves time and memory) while achieving satisfactory image matching percentages.

2.2 SURF Algorithm

The SURF algorithm incorporates enhancements to the SIFT algorithm that increase the overall speed [2]. The enhancements are described below in the context of the four steps of the SIFT algorithm.

1. Scale-Space Extrema Detection: SURF uses a 2×2 Hessian matrix, whose components are the convolution of the second-order Gaussian derivative with an area of the image centered at each pixel. To speed this process, a box filter approximation of the second-order Gaussian derivatives is used. The reduction in the scale of the images (to generate multiple scales) is then performed by increasing the size of the box filter approximation [2].

2. Keypoint Localization: SURF uses SIFT's 3D quadratic function to extract localized keypoints [2].

3. Orientation Assignment: Haar wavelet responses in the x and y directions are calculated over a circular neighborhood of radius $6s$ around each keypoint (s is the scale of the image). The Haar responses are weighted with a Gaussian distribution centered at the keypoint and are summed to generate the orientation vector [2].

4. Keypoint Descriptor Generation: The keypoint descriptor is calculated over a $20s$ pixel area around the keypoint oriented according to the orientation assignment. The area is divided into 16 square patches that are evenly spaced over the keypoint descriptor area. In each patch, the Haar wavelet responses in the x and y directions are calculated over a 4×4 pixel square for each pixel in the patch. The response vectors from each pixel in a patch are then combined. The four component vectors from each of the 16 patches give rise to the 64-element keypoint descriptor [2].

The SURF descriptor has similar properties to the SIFT descriptor but is less complex and is, therefore, faster to compute. The times required for keypoint descriptor generation are 354 ms, 391 ms and 1,036 ms for SURF (with a 64-element descriptor), SURF-128 (128-element descriptor) and SIFT, respectively [2]. The average recognition rates or accuracy of detecting repeat locations for SURF, SURF-128 and SIFT are 82.6%, 85.7% and 78.1%, respectively [2].

2.3 Other Image Matching Algorithms

An alternative image matching algorithm is PCA-SIFT [12], which incorporates principal components analysis. PCA-SIFT applies a normalized gradient patch instead of smoothed weighted histograms to generate the keypoint feature vector. This provides users with the ability to specify the size of the feature vector. The default feature vector size in PCA-SIFT is 20 [12]. Experiments show that SIFT runs slightly faster during keypoint generation, 1.59 sec vs. 1.64 sec [12]. However, the experiments also show that PCA-SIFT has a large performance advantage during image matching, 0.58 sec vs. 2.20 sec [12]. This improvement is due to a significant reduction in keypoint feature size (20 vs. 128).

The Shi-Tomasi algorithm [10] selects features that are suitable for tracking between image frames. Keypoints are generated over 7×7 blocks of pixels. The second-order partial derivatives of the intensity of the pixels are calculated for each pixel block. The eigenvalues of the derivatives are identified as an interest point if their minimum exceeds a user-specified threshold. The algorithm is most suitable for small camera position changes, but is not robust enough to handle the large displacements found in our application domain.

3. Keypoint Reduction and Matching

This section presents the methods used to reduce the number of keypoints and to identify a location match given variations in the viewpoint and content.

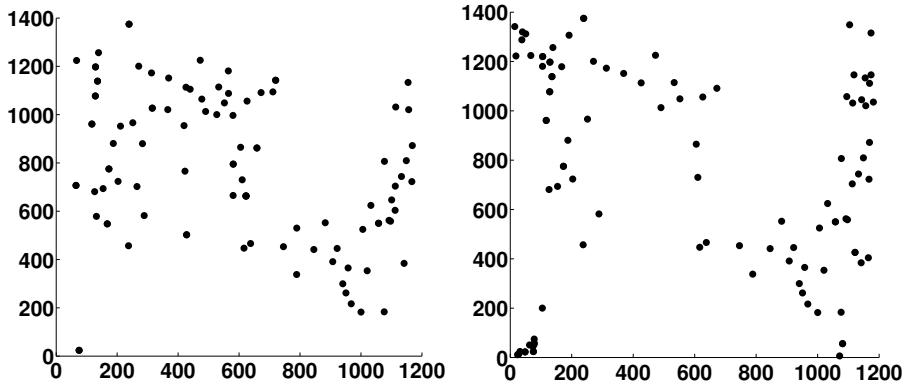
3.1 Keypoint Reduction

The SIFT and SURF algorithms generate an average of 3,000 keypoints per image. Reducing the number of keypoints significantly reduces memory requirements and image matching times but negatively impacts the matching accuracy. This problem can be addressed by choosing "stronger" keypoints that are well distributed in the image. A distance function helps ensure a good keypoint spread, which prevents keypoint clustering and subsequent image occlusion.

Keypoints are selected using an iterative approach. The SIFT algorithm selects the first two points based on the scale of the detected keypoints. For the SURF algorithm, the first two points are selected based on the log of the cardinality of the non zero (Nz) elements of the second moment matrix $\log\left(\frac{1}{\sqrt{|Nz|^2}}\right)$. Consequent keypoints are selected based on a weighted sum of the scale (SIFT) or second moment (SURF) of the keypoint and of the Mahalanobis distance between the keypoint and all previously chosen keypoints [11]. Keypoints are obtained by evaluating each available point (x_i, y_i) using $W_1 D_M(x_i, y_i) + W_2\sigma(x_i, y_i)$ to obtain the largest value. Note that $\sigma(x_i, y_i)$ is the scale/second moment, $D_M(x_i, y_i)$ is the Mahalanobis distance at point (x_i, y_i), W_1 is the weighting on the Mahalanobis distance function, and W_2 is the weighting on the scale/second moment of the keypoint. This process continues until the desired number of keypoints is selected.

The best settings for the distance weighting (W_1) and scale/second moment weighting (W_2) were determined by tests using distance weightings from 0.5 to 100 and a constant scale weighting of 1. The goal was to ensure that the selected keypoints are spread uniformly to prevent partial occlusion but still provide a strong probability of matching. Keypoints tend to cluster when the distance weighting is much greater than the scale/second moment weighting; equal weights generally result in a better distribution of keypoints.

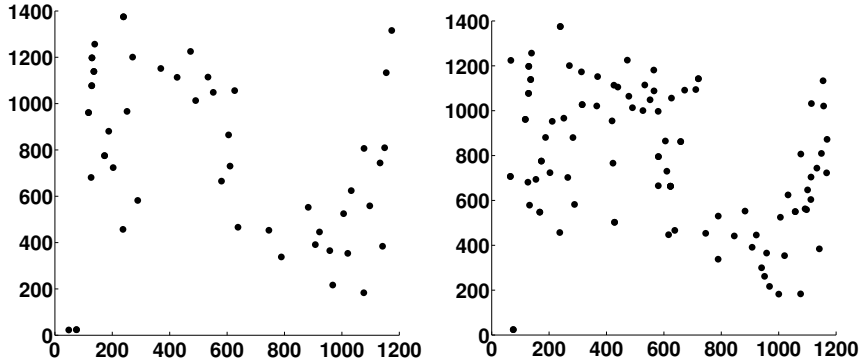
This trend is seen in Figure 1, where the settings of the distance weighting and the scale/second moment weighting of 0:1 (Figure 1(a)) produce a larger spread of keypoints than settings of 5:1 (Figure 1(b)). The figures show feature distributions of 102 keypoints; the figure axes are the x and y coordinates of pixels. The best settings for the distance



(a) Distance to scale ratio: 0:1. (b) Distance to scale ratio: 5:1.

Figure 1. Feature distributions of 102 keypoints.

weighting and the scale/second moment weighting were determined subjectively by overlaying the keypoint distributions and observing the levels of spread and clustering. The setting that results in the greatest spread of keypoints occurs when W_1 and W_2 are both equal to 1.



(a) Image with 52 keypoints. (b) Image with 102 keypoints.

Figure 2. Example keypoint distributions.

Limited testing was conducted to identify the best number of keypoints to select from an image. The tests compared the image keypoint distribution between selecting 52 keypoints (Figure 2(a)) versus 102 keypoints (Figure 2(b)). Both distributions were generated using distance

(W_1) and scale (W_2) weights of 1. The larger number of keypoints (102) provides a more uniform distribution along both axes.

The more uniform the distribution of points, the better the matching opportunities. Using a large number of keypoints was considered to address background occlusion. However, the computational cost of keypoint reduction is high, so a decision was made to limit the number of keypoints in subsequent tests to 102. More research is required to identify the optimal number of keypoints.

3.2 Background Matching Using SIFT

Image background matching with the SIFT algorithm involves an extension of Hess' SIFT implementation [6]. Each image is processed using the SIFT keypoint generation algorithm to produce 102 keypoints as described in Section 3.1. The image keypoints are stored in a database that is used for match comparisons. Next, the keypoints corresponding to each pair of images are compared. The best candidate match is found by calculating the nearest neighbor using a minimum Euclidean distance for the descriptor vector. The distance from the second-closest neighbor is used to define the distance ratio such that 90% of the bad matches are pruned with a distance ratio greater than 0.8 [8]. The Best Bin First algorithm is used to implement the nearest neighbor search; the Hough transform is used to identify clusters of features that help enhance the recognition of small or occluded objects [8].

Two quality checks are performed to eliminate poor matches; both checks use the same initial framework. First, each pair of match points are converted into lines calculated as if the two images are stacked on top of each other (see Figure 3 in Section 4.1). The intersection points for each line are then computed; these intersection points are used to identify poor matches. The first quality check removes a match if it produces intersection points within the frame of the match image. The second check calculates the mean and standard deviation of the intersection points; a line is a poor match when 90% or more of its intersection points lie outside one standard deviation from the mean.

3.3 Background Matching Using SURF

SURF image background matching is similar to that of SIFT with the exception that the MATLAB® keypoint generation software created by Alvaro and Guerrero [3] is employed. However, the quality checks developed for SIFT do not perform as well as those for SURF. The reason is that SIFT generates a significantly larger number of false matches; most matches are accepted because the standard deviation of the intersection

points is quite large. An additional check is incorporated prior to match filtering to improve the quality of matching. This check tests the slopes of the match lines against a threshold of 0.4; a match line is eliminated when its slope exceeds the threshold.

4. Experimental Results

A Fuji FinePix E550 was used to acquire the 125 images used to test the image background matching algorithms. The images were taken at six locations (home office, guest bedroom office, stairwell, living room, home exterior and computer laboratory). 119 images were taken at 1,600×1,200 resolution and six were taken at 640×480 resolution.

The images were taken from various vantage points with different points of view (POV). The camera distance for the indoor images varied between 2.75 feet and 11 feet; the rotation varied approximately ±15 degrees and the camera angle variation was more than ±50 degrees. The home office was the only location where images were taken at two resolutions (1,600×1,200 and 640×480). The outdoor images had much larger variations; the distance varied 50 feet and the rotation and camera angle varied ±10 degrees and more than ±180 degrees, respectively.

The images were divided into seven groups for testing. Images taken at each of the six locations were placed in a separate group, except for those taken at the home office, which were placed into two groups because the camera viewpoint for these images differed by 180 degrees.

The 125 images were converted to gray scale prior to matching. This is because the two matching algorithms use the intensity of each pixel $I(x, y)$ in keypoint calculations. It is possible to create keypoints in color images using each of the three color channels (red, green, blue) as separate intensity values, but the matching performance for both algorithms degrades.

The first step in the matching technique involved the extraction of the keypoints for each image using the SIFT and SURF algorithms. Next, keypoint reduction was performed using the method described in Section 3.1; the reduced keypoints were stored in a data file to facilitate matching. After matching, the keypoint comparison technique presented in Section 3.2 was performed on the matched keypoint lines in an effort to prune "bad" matches.

To verify the accuracy of the technique, each of 125 images was compared with every other image, resulting in a total of image 7,750 comparisons for each of the algorithms. However, before the algorithms were applied, a human who had not seen any of the image locations was asked to group the images based on location. The individual placed the images

into 24 groups using prominent reference points to distinguish image locations. Six of the 24 groups contained just one image. The accuracy of identification was 55% mainly due to the creation of extra groups.

The performance of the human could not be compared with that of SIFT and SURF because he grouped images individually instead of performing 7,750 comparisons (like the algorithms). Nevertheless, the experiment demonstrates the difficulty involved in matching images.

Reducing the number of the keypoints saved for each image conserves storage space. We demonstrate that this technique reduces storage as well as the time required for matching image locations. Specifically, we compare the storage and time requirements for our image matching technique with those for the SIFT and SURF algorithms. The tests were conducted using a dual core Xeon 3 GHz workstation with 3 GB RAM.

Table 1. Storage required by the SIFT and SURF algorithms.

Algorithm	**Size On Disk**	**Percent Reduction**
SIFT Files	197 MB	
Reduced SIFT Files	4.88 MB	97.5%
SURF Files	290 MB	
Reduced SURF Files	16.1 MB	94.4%

Table 1 shows the storage required by the SIFT and SURF algorithms before and after keypoint reduction. The storage requirements are for the 125 SIFT/SURF keypoint files generated from the 125 images used in the experiment. Keypoint reduction yields a 97.5% reduction in the storage requirements for SIFT. Similar results are obtained for the SURF algorithm (94.4% reduction).

Table 2. Execution time for the SIFT algorithm.

SIFT Algorithm	**Approximate Execution Time**	**Percent Reduction**
Match	24 hours 39 minutes	N/A
Reduced Match	6 hours 23 minutes	74.1%
Keypoint Reduction	3 hours 27 minutes	86.0%
Reduced Match and Keypoint Reduction	9 hours 50 minutes	60.1%

Using 102 well-selected keypoints per image instead of several thousand keypoints (which would otherwise be used) significantly reduces the time required to perform image matching. Table 2 presents the time required to run a complete matching experiment for the SIFT algorithm.

SIFT matching of the 125 images takes more than 24 hours whereas the time required for keypoint reduction and subsequent matching requires just 9 hours and 50 minutes, a 60.1% reduction.

Table 3. Execution time for the SURF algorithm.

SURF Algorithm	Approximate Execution Time	Percent Reduction
Match	12 hours 19 minutes	N/A
Reduced Match	2 hours 16 minutes	81.6%
Keypoint Reduction	1 hours 39 minutes	86.6%
Reduced Match and Keypoint Reduction	3 hours 55 minutes	68.2%

Table 3 shows that similar reductions in computational time are obtained for the SURF algorithm. SURF requires 12 hours and 19 minutes to perform a full match on the 125 test images. On the other hand, keypoint reduction and match requires only 3 hours and 55 minutes, a 68.2% reduction. Below we show that the storage and time savings come without significant loss of image matching accuracy.

4.1 SIFT Algorithm Results

Figure 3 shows that the SIFT match algorithm deals well with occlusion. A total of six matches were found in the two images in Figure 3. One of them – the one on the individual's arm – is an incorrect match. This incorrect match is pruned by both SIFT quality check methods.

Figures 4 and 5 indicate that relatively few images are incorrectly matched – this occurs when images of different locations are identified as being of the same location. Figure 4 shows that the Type I error (false positives) drops dramatically until a threshold of 4. As shown in Figure 5, 81.0% accuracy is obtained using a threshold (η) of 5. However, lower resolution images matched poorly with an accuracy of 72.5%.

The highest accuracy (81.1%) for the SIFT algorithm is obtained using a threshold of 6. In fact, correct matches were obtained even for a large threshold of 98 (not shown in Figure 5). However, using a threshold of 102 incorrectly drops some image matches; this is because the matching algorithm uses a nearest neighbor algorithm to identify keypoint matches and some of the neighbors are pruned during keypoint reduction [8].

There was no difference in the maximum accuracy obtained for the two quality checks. Note that the data in Figures 4 and 5 were computed using only the intersection standard deviation quality check.

Figure 3. SIFT image showing reduced keypoint matches with occlusion.

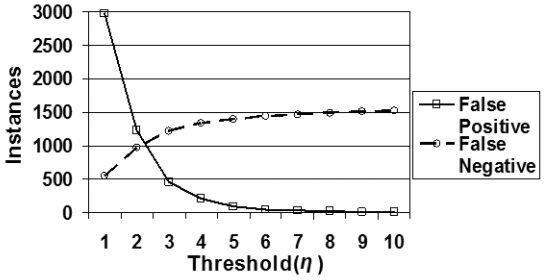


Figure 4. SIFT error with reduced keypoints.

The matching performance obtained with the keypoint reduction technique compares well against that obtained when using the full unreduced

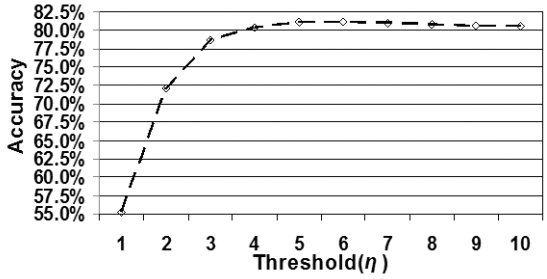


Figure 5. SIFT accuracy with reduced keypoints.

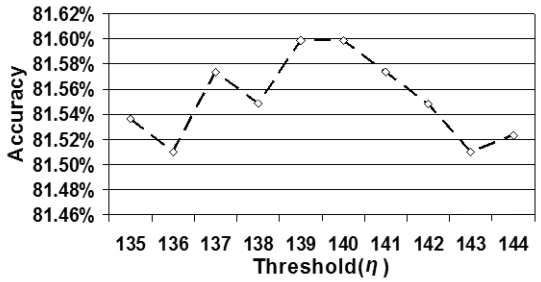


Figure 6. SIFT accuracy with unreduced features.

set of SIFT keypoints. Figure 6 shows that the maximum accuracy of 81.6% is achieved at thresholds of 139 and 140 for the SIFT algorithm without keypoint reduction. This accuracy (81.6%) is marginally better than that obtained for SIFT matching using keypoint reduction (81.1%).

4.2 SURF Algorithm Results

The SURF algorithm produces a larger number of matches than the SIFT algorithm, but the percentage of incorrect matches is much higher.

Figure 7 shows the SURF match image, which has a total of 44 matches. This image has many more incorrect matches than the corresponding SIFT image (Figure 3).

Figure 8 shows that the Type I error (false positives) and Type II error (false negatives) for the SURF algorithm with reduced keypoints are comparable to those for SIFT (Figure 4).

Figure 9 shows that the maximum accuracy of 79.6% for the SURF algorithm occurs at a threshold of 57, where the unreduced SURF accuracy is 78.3%. However, by adding the slope threshold of 0.4, the accuracy is improved to 80.7%.

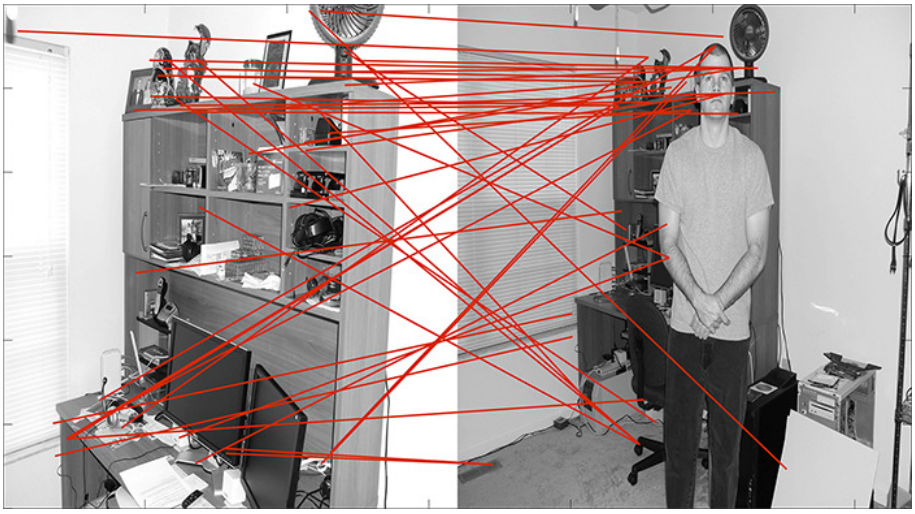


Figure 7. SURF image showing reduced keypoint matches with occlusion.

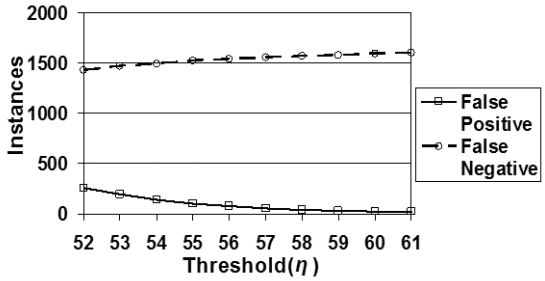


Figure 8. SURF error with reduced keypoints.

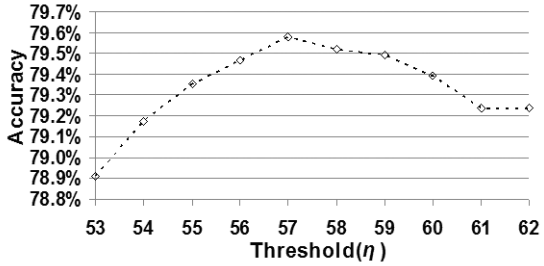


Figure 9. SURF accuracy with reduced keypoints.

5. Conclusions

Automating image background matching for the task of grouping images based on location is, indeed, feasible. Good results are obtained using the SIFT algorithm augmented with keypoint reduction. Specifically,

the SIFT algorithm provides a maximum accuracy of 81.1% whereas the SURF algorithm has a maximum accuracy is 79.6%. Significant space and time savings are obtained using keypoint reduction. The storage reduction for the SIFT and SURF algorithms are 97.5% and 94.4%, respectively. The corresponding savings in computational time for SIFT and SURF are 60.1% and 68.2%, respectively.

Additional work is needed to enhance image background matching with reduced keypoints. This includes analyzing match points to improve matching accuracy and identifying optimal threshold values for the SIFT and SURF quality check methods. Furthermore, tests need to be run on large databases of images with varying content, size and quality.

References

[1] W. Ashley, What shoe was that? The use of a computerized image database to assist in identification, *Forensic Science International*, vol. 82(1), pp. 7–20, 1996.

[2] H. Bay, L. Van Gool and T. Tuytelaars, SURF: Speeded up robust features, *Proceedings of the Ninth European Conference on Computer Vision*, pp. 404–417, 2006.

[3] H. Bay, L. Van Gool and T. Tuytelaars, SURF: Speeded Up Robust Features Software (www.vision.ee.ethz.ch/∼surf/index.html).

[4] S. Birchfield, KLT: An Implementation of the Kanade-Lucas-Tomasi Feature Tracker (www.ces.clemson.edu/∼stb/klt).

[5] J. Gonzalez-Rodriguez, J. Fierrez-Aguilar, D. Ramos-Castro and J. Ortega-Garcia, Bayesian analysis of fingerprint, face and signature evidence with automatic biometric systems, *Forensic Science International*, vol. 155(2-3), pp. 126–140, 2005.

[6] R. Hess, SIFT Software (web.engr.oregonstate.edu/∼hess).

[7] D. Lowe, Object recognition from local scale-invariant features, *Proceedings of the International Conference on Computer Vision*, pp. 1150–1157, 1999.

[8] D. Lowe, Distinctive image features from scale-invariant keypoints, *International Journal of Computer Vision*, vol. 60(2), pp. 91–110, 2004.

[9] F. Murtagh, Z. Geradts, J. Bijhold and R. Hermsen. Image matching algorithms for breech face marks and firing pins in a database of spent cartridge cases of firearms, *Forensic Science International*, vol. 119(1), pp. 97–106, 2001.

[10] J. Shi and C. Tomasi, Good features to track, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 593–600, 1994.

[11] M. Veth and J. Raquet, Fusion of low-cost imaging and inertial sensors for navigation, *Proceedings of the Institute of Navigation Global Navigation Satellite System Conference*, 2006.

[12] S. Zickler and A. Efros, Detection of multiple deformable objects using PCA-SIFT, *Proceedings of the Twenty-Second National Conference on Artificial Intelligence*, pp. 1127–1132, 2007.

[13] W. Zhao, R. Chellappa, P. Phillips and A. Rosenfeld, Face recognition: A literature survey, *ACM Computing Surveys*, vol. 35(4), pp. 399–458, 2003.