SenseCam Image Localisation using Hierarchical SURF Trees

Ciarán Ó Conaire, Michael Blighe, Noel E. O'Connor

Clarity Centre, Centre for Digital Video Processing, Dublin City University, Ireland
oconaire@eeng.dcu.ie,
http://www.cdvp.dcu.ie

Abstract. The SenseCam is a wearable camera that automatically takes photos of the wearer's activities, generating thousands of images per day. Automatically organising these images for efficient search and retrieval is a challenging task, but can be simplified by providing semantic information with each photo, such as the wearer's location during capture time. We propose a method for automatically determining the wearer's location using an annotated image database, described using SURF interest point descriptors. We show that SURF out-performs SIFT in matching SenseCam images and that matching can be done efficiently using hierarchical trees of SURF descriptors. Additionally, by re-ranking the top images using bi-directional SURF matches, location matching performance is improved further.

Keywords: Image matching, SenseCam, localisation, SURF.

1 Introduction

The SenseCam is a wearable camera that automatically captures images of a user's activities (figure 1). In order to organise the thousands of images generated per day, the physical location of the user for each image captured could potentially be very useful as additional metadata. For example, if you had lost your briefcase, your SenseCam could help you automatically retrace your steps. Similarly, at a conference it could be used to establish what posters you had visited, providing the basis for determining where you spent your time (and thus by implication what you found most interesting).

Of course, a number of technological solutions are available for localisation, such as GPS [1], GSM [2] and RF-based localisation using base-station signal strength [3]. It is not always desirable or even possible to use these technologies. For example, GPS localisation is unreliable indoors; GSM or RF-based localisation may not support localisation to a fine granularity, required for some applications. The SenseCam could, for example, replace the traditional audio-guide used in many museums [4]. This would require no additional overhead in terms of infrastructure and would allow wearers to follow a more natural path

Fig. 1. The SenseCam: a prototype wearable camera. The bottom two rows show some typical SenseCam images.

through the museum, rather than obeying the predefined audio-guide route. In applications, such as the monitoring of people with age-dementia [5], a wearable visual monitor is a useful diagnostic tool, and knowledge of the wearer's location could facilitate more efficient browsing of a large repository of captured images or video.

In this paper, we address the problem of estimating a person's trajectory through a space, using only their SenseCam images. In our proposed approach, we begin by creating an image database of known positions. SURF image features [6] are extracted from the database images and clustered into a hierarchical-tree. This data structure allows fast matching of a query image with its most similar database images. Finally, since SenseCam images have a temporal order, we can use this to impose some constraints on the user's path through the space. Three strategies for path optimisation are evaluated.

This paper is organised as follows: In section 2, we provide a brief background literature review to contextualise our work. The recently proposed SURF descriptor is compared to the standard SIFT method in section 3. Section 4 de-

scribes our experimental setup. We present results in section 5, demonstrating the localisation accuracy of the proposed approach and give our conclusions and directions for future work in section 6.

2 Related work

Many different approaches to automatic user localisation have been proposed in the literature. In [3], Bahl and Padmanabhan present an RF-based system for location estimation that uses signal-strength information from wireless network base-stations at known locations. Using a similar approach, GSM has also shown potential for providing good localisation [2]. Additionally, GPS has been used in many systems, but it does not work indoors and recent studies have shown that GPS coverage is only available for 4.5% of the time a user carries a device over a typical day [1]. Image based localisation provides an alternative and complementary approach to these technologies.

A great deal of work has been done in the autonomous vehicle community on Simultaneous Localisation and Mapping (SLAM), whereby a mobile robot builds a map of its environment and at the same time uses this map to compute its own location [7]. In this paper, we tackle a different problem whereby we cannot control the movements of the camera wearer and the image capture rate is significantly lower than for video.

Kosecka and Yang [8] use SIFT features [9] for user localisation by creating an image database of known locations and matching query images to their database. The SIFT descriptor is a gradient orientation histogram robust to illumination and viewpoint changes. In a similar vein to SIFT and other interest point descriptors (many of which are evaluated in [10]), the recently proposed SURF method [6] locates interest points and extracts an invariant descriptor for each point. However, SURF achieves greater computational efficiency by using integral images. In order to efficiently locate relevant images in a large database, Nistér and Stewnius [11] propose the use of a *vocabulary tree* of SIFT descriptors.

Our previous work on the SenseCam includes [12] and [13], where we have focused on developing SenseCam-image clustering techniques to assist user browsing by grouping the thousands of images generated by the device into *events*. We have also examined matching events in a person's life [14]. We believe that knowledge of the user's location will be a valuable additional aid to efficient organisation of SenseCam images.

3 Feature Comparison

3.1 Data capture

In order to perform a straight-forward comparison between the SIFT and SURF descriptors, SenseCam images of static scenes are captured by fixing the camera in place and taking one image every 5-10 minutes. The interest-point descriptors found in each image of a single static scene should have unique corresponding

Fig. 2. Some of the images use to compare SURF and SIFT. The first two columns differ only by ambient lighting, camera noise and compression effects. The third column shows images that contain a severe change in lighting.

points in other images of the scene. SenseCam images have high noise, due to the lack of a flash, and a high number of compression artifacts, since a JPEG compression quality of 60% is used. These static scene comparisons test each method's robustness to these distortions. We also test the more difficult scenario of a severe lighting change. This is simulated by turning on another set of lights in the room. Previous works, notably [10] and [6], have more thoroughly tested other interesting distortions not considered in this work, such as camera rotation, scaling and blur. In total, 7 static scenes and 2 lighting-change scenes were captured, with approximately 9 images each. In each scene, all pairs of images are compared.

3.2 Evaluation: SIFT vs. SURF

In our tests, the implementation used for SIFT was the original version by Lowe [9]. Similarly, the code provided online for SURF by Bay et al. was used [6]. Due to the camera noise, SIFT returned many interest points at small scales. These were found to be very unstable. Therefore, to improve SIFT performance, points with scales less than 2 were removed. The standard SURF descriptor is rotationally invariant, but since the images captured with a SenseCam are almost always upright, the non-rotationally invariant USURF is appropriate for our application. We compared these two variants, as well as there *extended* versions that are twice as long but increase matching accuracy. In total, the four variants of SURF we tested were: SURF_{64}, SURF_{128}, USURF_{64} and USURF_{128}.

To measure the stability (or repeatability) of the interest point detectors of the methods, we computed (for a pair of images) the fraction of interest points in the 1^{st} image that had a match in the 2^{nd} image. Since each point is described

	$SIFT$	$SURF_{64}$	$SURF_{128}$	$USURF_{64}$	$USURF_{128}$
Stability	0.7190	**0.8094**	**0.8094**	**0.8094**	**0.8094**
Average Precision	0.9965	0.9960	0.9952	**0.9974**	0.9970

Table 1. Static scene comparison of SIFT and SURF. Rows show figures for Stability and Average Precision. Columns show SIFT and 4 SURF variants. the SIFT descriptor is a 128-dimensional vector. The number beside each SURF descriptor indicates the size of the descriptor and USURF is a non-rotationally-invariant version of SURF. Since all SURF variants use the same detector, their stability scores are equal. USURF_{64} performs best in these tests overall.

by a (circular) region, a match was declared if for two regions, the intersection area divided by the union area was greater than 0.5, as proposed in [10]. The stability value shown in the tables is an average of all image comparisons.

To measure the performance of the descriptors, we used the distance ratio test [9]. To examine whether a point from the 1^{st} image has a match in the 2^{nd}, its two most similar descriptors in the 2^{nd} image are found. If the ratio of the nearest distance to the second nearest distance is less than α, a match is declared. By varying this α parameter, the precision-recall curves in figures 3 and 4 were generated. For each descriptor, the average precision was computed as the area under the curve.

Table 1 shows the results when the testing images only differ by slight changes in ambient lighting, camera noise and compression effects. Examples are shown in the first two columns of figure 2. To examine more drastic changes, table 2 shows the results when the testing images suffer a severe lighting change (as well as camera noise and compression effects). For example, compare the image in the first two columns of figure 2 to the images in the third column.

Both sets of results indicated that the USURF_{64} descriptor provided high stability and matching performance and therefore this was adopted for our experiments in image matching. USURF_{128} is marginally better for severe lighting changes, but the trade-off is that the descriptor is twice as large, so we retained the smaller descriptor. SIFT was adversely affected by the severe lighting change, as can be seen in figure 4.

In terms of choosing a threshold for matching, Lowe suggests using a value of $\alpha = 0.6$ [9]. We found that by optimising various performance measures (such as the F_1 measure), larger values of α were preferred for SenseCam image matching, due to the higher noise. In the rest of this work we use $\alpha = 0.7$.

4 Experimental setup

In our experiments, we wished to trace the path of an individual through our labs, comprising of two large office spaces and a corridor (see map in figure 10). A database of 156 images was created by capturing SenseCam images uniformly over the area. The median distance from a point in the space to the position of

	$SIFT$	$SURF_{64}$	$SURF_{128}$	$USURF_{64}$	$USURF_{128}$
Stability	0.3081	**0.4706**	**0.4706**	**0.4706**	**0.4706**
Average Precision	0.6600	0.8099	0.8054	0.8379	**0.8392**

Table 2. Severe lighting changes: Comparison of SIFT and SURF for feature matching. See table 1 for details of table abbreviations. USURF_{128} performs marginally better than USURF_{64} in these tests.

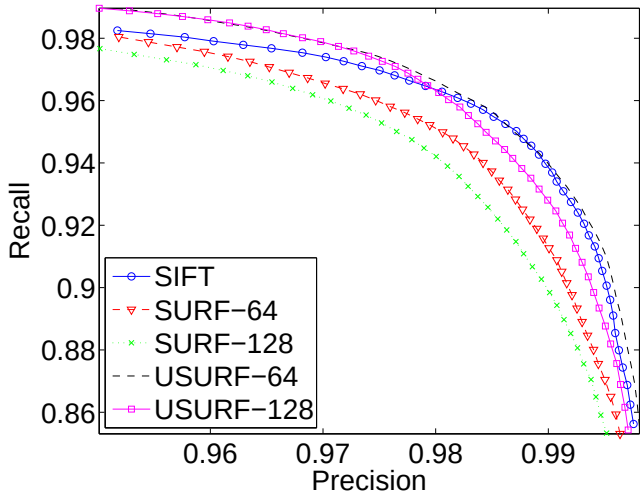

Fig. 3. Robustness to camera noise, JPEG compression and ambient lighting changes: Precision-Recall curve for the evaluated descriptors.

the nearest database image is 1.19m. The position of each image was manually annotated using an interactive map-based tool. Testing sequences were similarly annotated to act as a ground truth for our experiments. Image distortion due to the fish-eye lens was removed using the Camera Calibration Toolbox for Matlab [15]. Examples of out database and testing images are shown in figures 7 and 8.

To measure how likely it is that 2 images are of the same location, Kosecka and Yang [8] simply counted the number of SIFT feature matches between the query and database (DB) image. We experimented with this measure and various others, such as reverse matching (number of matches from DB-to-query), summing both counts (sum of query-to-DB and DB-to-query) and RANSAC inliers (estimating the fundamental matrix using all matches, and counting the inliers). We found the best strategy was to perform matching in both directions (query-to-DB and DB-to-query) and to count the number of correspondences that were found in both directions (e.g. point A matches to B in query-to-DB matching, and point B matches to point A in DB-to-query matching). We refer to these as *bi-directional matches*. To compare the 5 strategies, we used all 273 images in

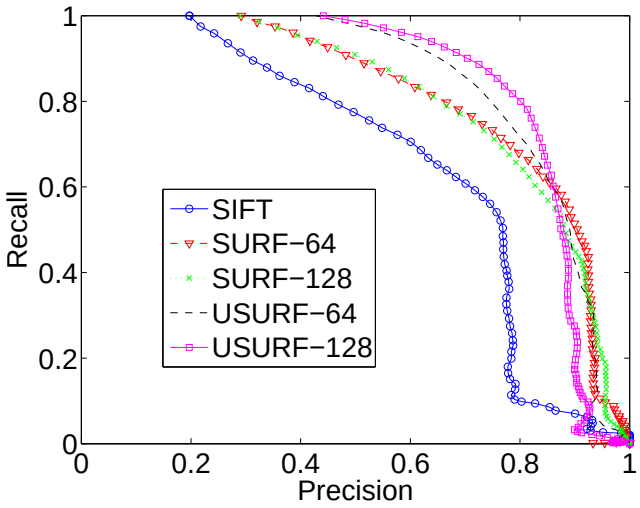

Fig. 4. Robustness to severe lighting changes (as well as camera noise, JPEG compression and ambient lighting changes): Precision-Recall curve for the evaluated descriptors. The SIFT descriptor fares quite poorly here compared to the SURF descriptors.

9 training sequences and compared each one to all 156 database images. Figure 5 illustrates the robustness of each measure in image matching. By varying a threshold on the measure, we can trade-off the recall of the correct location with the average position error. Bi-directional matches significantly out-perform the other measures in this regard.

Ideally, we would rank database images by their bi-directional matches with the query image, but this is computationally expensive, so inspired by [11], a fast voting method was developed to narrow the search to a small subset of the database. All $42,595$ SURF descriptors were extracted from the 156 database images. They were split into two groups (+1/-1) based on the sign of the Laplacian (which is included in the SURF descriptor). For each group, we then applied the K-means clustering algorithm recursively. First, clustering the data-points into K clusters, then clustering the points in each cluster into K sub-clusters and continuing recursively until a cluster contains less than K data-points. This created two hierarchical-SURF-trees which allowed rapid descriptor matching. Given a SURF descriptor, it can be compared to the K root nodes of the tree matching its Laplacian sign. The closest node can then be chosen and its subtree traversed recursively. The matched leaf node is then the approximate nearest-neighbour of the query descriptor. Unlike [11] where inverse files are used, we label each leaf-node with the DB image from which it is derived. We adopt a simple voting strategy to find potential DB images. Each SURF point in the query image is matched to its approximate-NN by the SURF-tree, and then casts one vote for the corresponding DB image. Our MATLAB

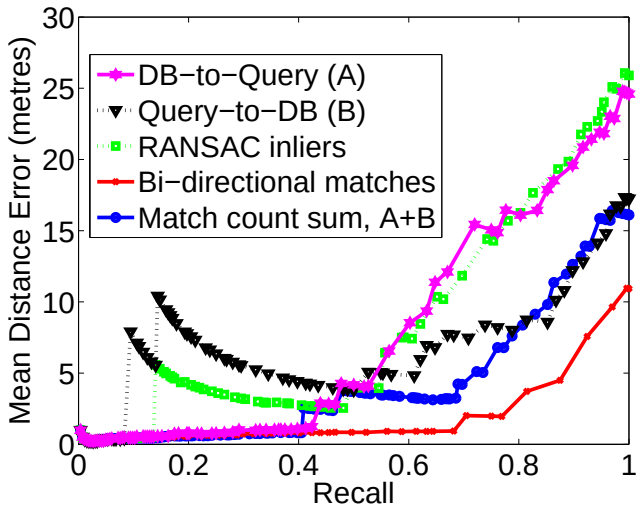

Fig. 5. Evaluating the methods of evaluating if two images are a location *match*: using the 9 training sequences (273 images). Bi-directional matches clearly out-perform the other matching measures.

code for K-means-hierarchical-tree generation and matching is available online at: `http://elm.eeng.dcu.ie/~oconaire/source/`

In order to estimate how well this voting strategy approximates bi-directional match ranking, we measured how frequently the best database match (highest *score* according to bi-directional matching) appeared in the top N results. Figure 6 shows the results of this test, and also examines the effects of the branching factor K. As in [11], we found that larger values of K perform better, though decreasing speed slightly. In this work, we used $K = 16$, making the feature matching more than 600 times faster than an exhaustive search. Query images with a low score probably do not have a good match in the database. When we ignore images that have less than 4 bi-directional-matches, then the performance can be seen to be even better.

The task of image matching in our chosen environment was quite challenging. Both the database and testing data contain distracting objects, such as people. The database images were taken 3 months before the testing images, so objects had been moved, Christmas decorations introduced, etc. Additionally, both offices spaces are structurally almost identical.

Our user localisation approach works as follows. For a query image Q_i, we extract SURF descriptors and use the hierarchical-SURF-trees to vote for DB images. The 7 highest scoring DB images are then processed to find bi-directional matches. The DB image with the most bi-directional matches is determined to be the best match. Let B_i denote the (x, y) ground position of the best

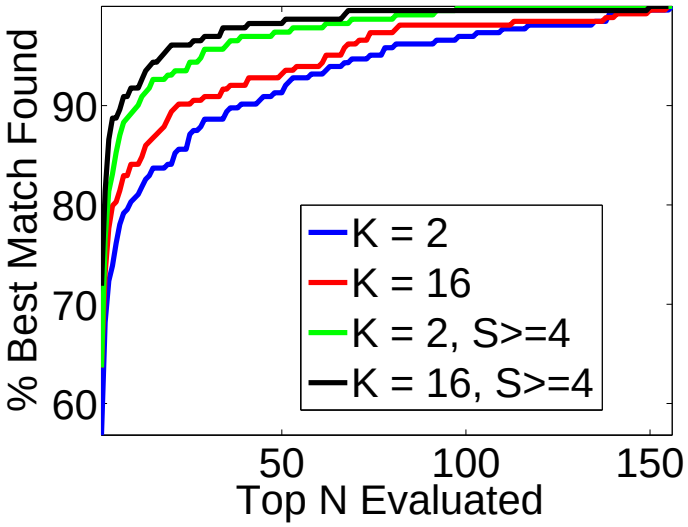

Fig. 6. Evaluating the voting performance: This graphs shows how well the voting performs in quickly finding the best matching image. K is the branching factor and S is the number of bi-directional matches.

Fig. 7. Examples of annotated database images of the environment.

match to query image Q_i, obtaining S_i bi-directional matches. Let W_i denote the estimated (x, y) position of the user when image Q_i was captured.

We evaluated 3 strategies in determining a user's motion through the space. Method 1 is simply to use the best match: $W_i = B_i$. Method 2 is to set a confidence threshold, T, and use it to determine if we should use the best match. If $S_i \geq T$ then a reliable match is declared and $W_i = B_i$, otherwise it's position would be interpolated linearly using other reliable matches. Using 9 training sequences, we found the minimum average distance error occurred at $T = 5$. Method 3 is the same as method 2, but the interpolation is performed using the geodesic distance, which takes into account the positions of walls and doors. Additionally, if the distance between the estimated positions of two consecutive images was greater than 10m, the position of the image with the lower score was interpolated.

Fig. 8. Examples of testing images.

5 Experimental Results

Figure 9 shows some examples of the matching. In column 1, the images are matched despite occlusion and one image captured during the day, the other at night. Similarly, columns 2, 3 and 4 indicate the robustness of the matching to the introduction of new objects and scale changes. An example of an estimated trajectory, along with the ground truth is shown in figure 10. Table 3 shows the overall localisation accuracy of the three methods over 9 training and 9 testing sequences. From the improvement of method 2 over method 1, it is clear that the use of the image temporal ordering is important for accurate path estimation. The use of the geodesic distance and outlier elimination reduces the error further by over 12%.

Fig. 9. (top row) query images, (2^{nd} row) their best matching database images

Sequence	*Method* 1	*Method* 2	*Method* 3
Training Data	2.26m	1.34m	1.10m
Testing Data	3.11m	1.77m	1.55m

Table 3. Average distance error in metres.

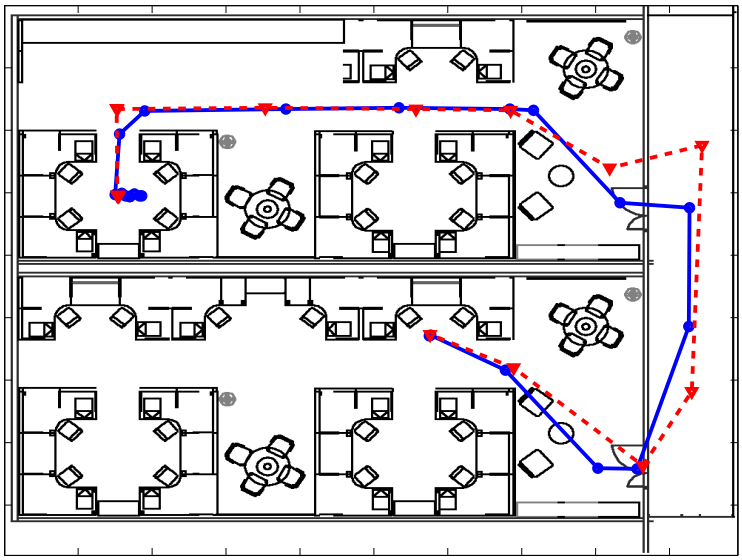

Fig. 10. Example trajectory: ground truth (solid blue line) and estimated trajectory (dashed red line).

6 Conclusion and Future Work

In this paper we presented our approach to user localisation using SenseCam images. Firstly, the newer SURF descriptor was compared to the commonly used SIFT descriptor and was found to have superior stability and matching performance in SenseCam imagery. Secondly, inspired by [11], we developed a fast method of finding potential query matches in our annotated location database. Thirdly, we demonstrated that SURF *bi-directional matching* out-performs other measures for image matching. Finally, we evaluated three strategies for estimating the path a user walked through the space and showed that accurate user localisation is possible, despite challenging image data. Future work will investigate fusing image information with complementary RF-based localisation using multiple wireless network base-stations.

References

1. LaMarca, A., et al.: Place lab: Device positioning using radio beacons in the wild. In: Proceedings of the Third International Conference on Pervasive Computing. (May 2005)
2. Varshavsky, A., et al.: Are gsm phones the solution for localization? In: 7th IEEE Workshop on Mobile Computing Systems and Applications. (2006)
3. Bahl, P., Padmanabhan, V.N.: Radar: An in-building rf-based user location and tracking system. In: Proceedings of the IEEE Infocom. (March 2000)

4. Fasel, B., Gool, L.V.: Interactive museum guide: Accurate retrieval of object descriptions. Adaptive Multimedia Retrieval: User, Context, and Feedback **4398** (2007)
5. Megret, R., Szolgay, D., Benois-Pineau, J., Joly, P., Pinquier, J., Dartigues, J.F., Helmer, C.: Wearable video monitoring of people with age dementia: Video indexing at the service of healthcare. In: International Workshop on Content-Based Multimedia Indexing (CBMI). (2008)
6. Bay, H., Tuytelaars, T., Gool, L.V.: Surf: Speeded up robust features. In: 9th European Conference on Computer Vision (ECCV'06). (May 2006)
7. Durrant-whyte, H., Bailey, T.: Simultaneous localisation and mapping (*slam*): Part 1, the essential algorithms. Robotics and Automation Magazine (2006)
8. Kosecka, J., Yang, X.: Global localization and relative positioning based on scale-invariant keypoints. In: 17th International Conference on Pattern Recognition. Volume 4. (Aug 2004) 319–322
9. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision **60**(2) (2004) 91–110
10. Mikolajczyk, K., Schmid, C.: A performance evaluation of local descriptors. IEEE Transactions on Pattern Analysis and Machine Intelligence **27**(10) (2005) 1615–1630
11. Nistér, D., Stewnius, H.: Scalable recognition with a vocabulary tree. In: IEEE Conference on Computer Vision and Pattern Recognition. (June 2006) 2161–2168
12. Ó Conaire, C., O'Connor, N.E., Smeaton, A., Jones, G.J.F.: Organising a daily visual diary using multi-feature clustering. In: Proc. of 19th annual Symposium on Electronic Imaging. (2007)
13. Blighe, M., Borgne, H.L., O'Connor, N., Smeaton, A.F., Jones, G.: Exploiting context information to aid landmark detection in sensecam images. In: International Workshop on Exploiting Context Histories in Smart Environments (ECHISE 2006) - Infrastructures and Design, 8th International Conference of Ubiquitous Computing (Ubicomp 2006). (Sept 2006)
14. Doherty, A.R., Ó Conaire, C., Blighe, M., Smeaton, A.F., O'Connor, N.E.: Combining image descriptors to effectively retrieve events from visual lifelogs (under review). In: Multimedia Information Retrieval (MIR). (2008)
15. Bouguet, J.Y.: Camera calibration toolbox for matlab. http://www.vision.caltech.edu/bouguetj/calib_doc/index.html