MapSnapper: Engineering an Efficient Algorithm for Matching Images of Maps from Mobile Phones

Jonathon S. Hare[a], Paul H. Lewis[a], Layla Gordon[b] and Glen Hart[b]

[a]School of Electronics and Computer Science, University of Southampton, Southampton, SO17 1BJ, UK;
[b]Ordnance Survey, Romsey Road, Southampton, SO16 4GU, UK

ABSTRACT

The MapSnapper project aimed to develop a system for robust matching of low-quality images of a paper map taken from a mobile phone against a high quality digital raster representation of the same map. The paper presents a novel methodology for performing content-based image retrieval and object recognition from query images that have been degraded by noise and subjected to transformations through the imaging system. In addition the paper also provides an insight into the evaluation-driven development process that was used to incrementally improve the matching performance until the design specifications were met.

Keywords: Image Matching, Mobile device, Interest points, Salient regions, Local descriptors, Geometric matching constraints, Robust algorithms

1. INTRODUCTION

The idea of using a mobile device as a platform for information retrieval is not a new one. An example of this is the research on the "physical hyper-link" carried out at HP labs,[1] where a user can 'click' on real world objects as if they were a hyperlink, using a mobile device as the interface. The research at HP did not use machine vision techniques, but instead relied upon iButtons, radio-frequency markers, bar-codes and infrared beacons. More recently, techniques for information retrieval on mobile devices using computer vision have been exploited by a number of research groups including our own.[2–4]

This manuscript aims to describe the design process which was undertaken in order to engineer a successful matching algorithm in the context of the MapSnapper project. In particular, the paper discusses the different approaches that were taken to optimise the algorithm, in terms of both matching accuracy and computational efficiency, in order to meet the overall design goals.

The MapSnapper project aimed to explore how computer vision techniques could be exploited for the matching of low-resolution digital photographs of Ordnance Survey paper map products to digital versions of the same map. In particular, the aim was to ascertain of which part of the map the photograph was taken. The motivation for this comes from a desire to exploit the current ubiquity of mobile information devices that incorporate digital cameras, such as mobile phones and personal digital assistants, and combine these devices with Ordnance Survey paper map products. The vision was of a product that would allow users to query a remote information system based on photos of a paper map taken with the device. The information system could then return useful information to the user via the device. For example, the returned information could include such things as events, facilities, opening times, and accommodation in the geographical region depicted by the query.

Further author information E-mail: jsh2@ecs.soton.ac.uk

2. REQUIREMENTS

The basic requirement of the MapSnapper project was to design an algorithm capable of matching low-resolution images to a map with sufficient accuracy. The algorithms have been designed to return the map coordinates corresponding to the centre of the query image. In addition, the algorithm had to operate in a timely manner.

For purposes of experimentation, sets of test images were created using the cameras of two mobile phones; a Sony-Ericsson T630, and a Sony-Ericsson K750. The T630 represents a first generation camera phone with resolutions of up to CIF size (288×352 pixels). The K750 is a second generation camera phone with a 2 mega-pixel camera. The images used for the test collection from both phones were all shot at the lowest resolution of 120×160 pixels. The K750 images are much sharper because of the better optics on the device. The photographs were captured by holding the camera-phone just above the surface of the map. The distance between the map and camera was approximately 10cm, such that each photograph depicts 1-2 grid squares on the map. No attempt was made to frame a particular grid square. In total, 118 photos were captured from a 1:25000 scale Ordnance Survey Landranger series map, covering an area of 20×60 kilometres. Some exemplar images are shown in Figure 1.

(a)

(b)

Figure 1. Example photos from the T630 (a) and K750 (b)

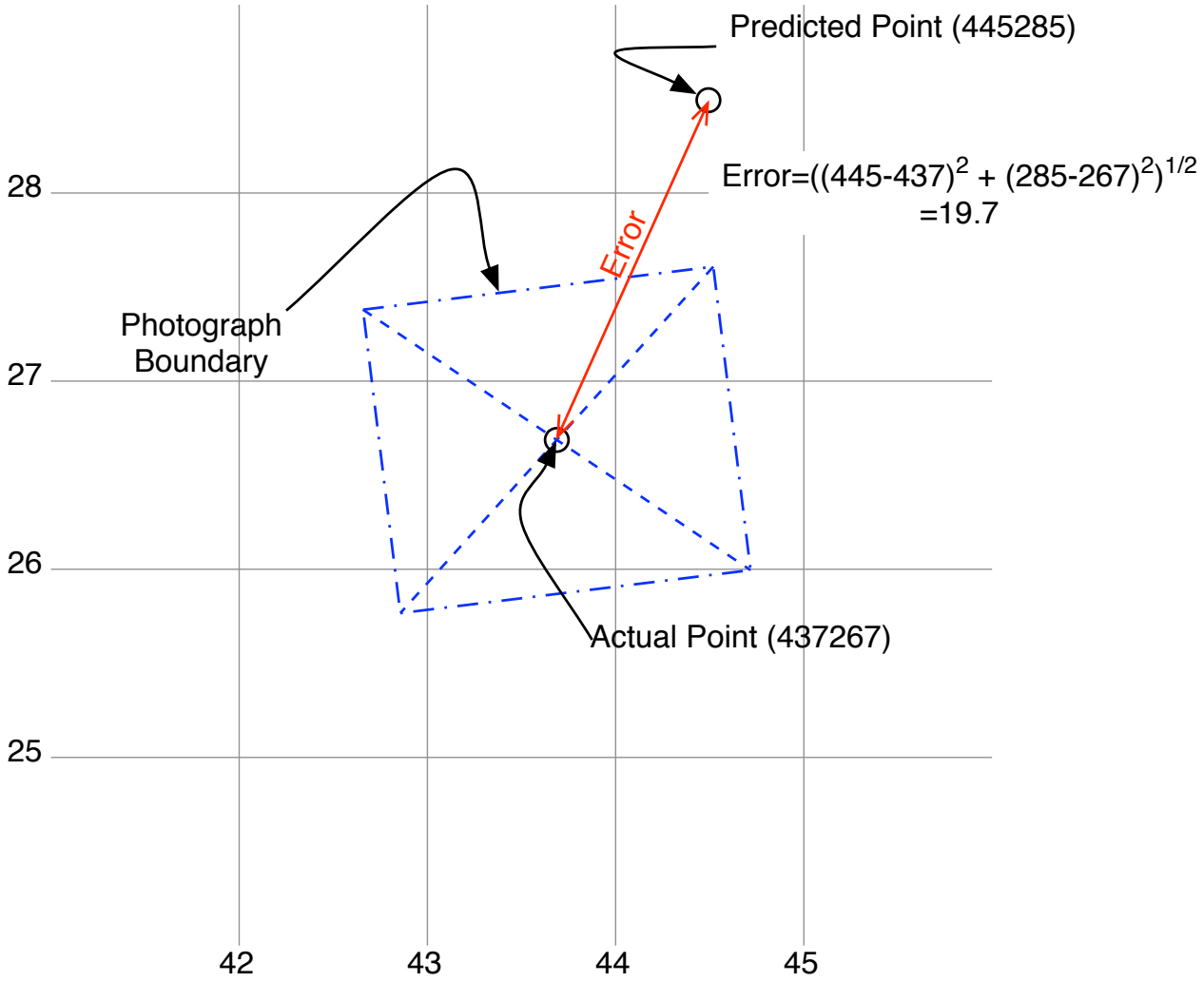

Figure 2. Error measurement

Each of the test images was matched by hand to the map, and the 6-figure grid reference corresponding to the centre of the photo was recorded. This allowed accuracy to be assessed by measuring the distance between the grid reference predicted by the algorithms and the ground truth grid references, as shown in Figure 2.

The error values given by this approach correspond to real-world distances in hundreds of metres. For example an error of 10.0 corresponds to 1km in real terms, or the size of one grid square on the map.

3. ARCHITECTURE

The proposed architecture for the MapSnapper system is based on previous experience in engineering robust retrieval systems. In particular, the use of salient regions for robust image matching and retrieval[5–12] has been exploited. Our own previous research on mobile retrieval within a museum setting[3] was used as a basis for the design of the system architecture.

The MapSnapper matching algorithm architecture consists of four core techniques in computer vision and robust geometry; multiscale interest point (or region) extraction, local descriptor generation, interest point matching, and the estimation of a geometrically consistent model of the matches.

Throughout the design and implementation of the MapSnapper architecture, it was assumed that the multiscale interest point detection would be performed using the difference-of-Gaussian interest point technique developed by Lowe,[9,11] and that the local descriptor would be Lowe's SIFT descriptor.[11] The difference-of-Gaussian detector works by selecting scale-space peaks detected in a multi-scale difference-of-Gaussian pyramid. Peaks in a difference-of-Gaussian pyramid have been shown to provide the most stable interest regions when compared to a range of other interest point detectors.[7,13,14] Recently, a number of state-of-the-art techniques have been suggested that are able to detect regions that are invariant to affine transforms.[10,13,15–17] However, these approaches are not yet fully affine invariant as they start with initial feature scales and locations selected in a non-affine-invariant manner. Mikolajczyk[13] showed that the performance of his affine invariant detector was below that of the difference-of-Gaussian peaks detection method, until the difference in viewpoint of the two images being matched was very large.

In the case of the MapSnapper project, we don't expect the captured images to exhibit large amounts of out-of-plane rotation because of the proximity of the camera to the map. Our own previous research showed that when compared to a number of state-of-the-art affine invariant detectors, the difference-of-Gaussian detector was the most consistent in performance under the typical imaging conditions that we would expect within a mobile retrieval scenario.[14]

There are a large number of different types of feature descriptors that have been suggested for describing the local image content within a salient region; For example colour moments and Gabor texture descriptors.[18–20] However, many of these descriptors are not robust to poor imaging conditions. A study by Mikolajczyk and Schmid[21] showed that the Scale Invariant Feature Transform (SIFT) descriptor,[11] was superior to other descriptors found in the literature, such as the response of steerable filters or orthogonal filters. The performance of the SIFT descriptor is enhanced because it was designed to be invariant to small shifts in the position of the sampling region, as might happen in the presence of imaging noise.

The SIFT descriptor is a three-dimensional histogram of gradient location and orientation. Lowe, suggests that gradient location be quantised into a 4×4 location grid, and gradient angle be quantised into 8 orientation bins. The resulting descriptor has 128 dimensions. Illumination invariance is obtained by normalising the descriptor by the square root of the sum of the squared components.

Much like many existing retrieval systems, the MapSnapper design calls for two stages. In the *offline* stage, a database of image features is created and indexed. In the *online* stage, the system receives a query image and matches the image against the feature database.

The basic architecture of the offline database creation stage is shown in Figure 3. The process of creating the database is simple and consists of three processing stages:

1. Extraction of multiscale interest regions: Interesting parts of the map image are extracted, and their position and scale is recorded.
2. Generation of local image descriptors for each interest region: A feature-vector representing the pixel-level image content within the region is created and stored for each region.
3. Transformation of position information: The locations of each interest region (in the image coordinate system) are converted to their location in the coordinate system of the map. This consists of multiplying the positions by a scale factor and adding an offset.

The online querying stage of the system, illustrated in Figure 4, contains the same steps as for the offline stage in order to create a list of interest regions together with feature-vectors representing the query image. The remaining stages are as follows:

1. Matching interest regions: Each interest region in the query is matched to the database by comparing the similarity of the feature-vectors. Matching pairs of interest regions are recorded.
2. Geometric Consistency: The pairs of matching interest regions are analysed in order to try and extract a plausible geometric mapping between the query and the map. The geometric mapping has the form of a 3×3 projection matrix.

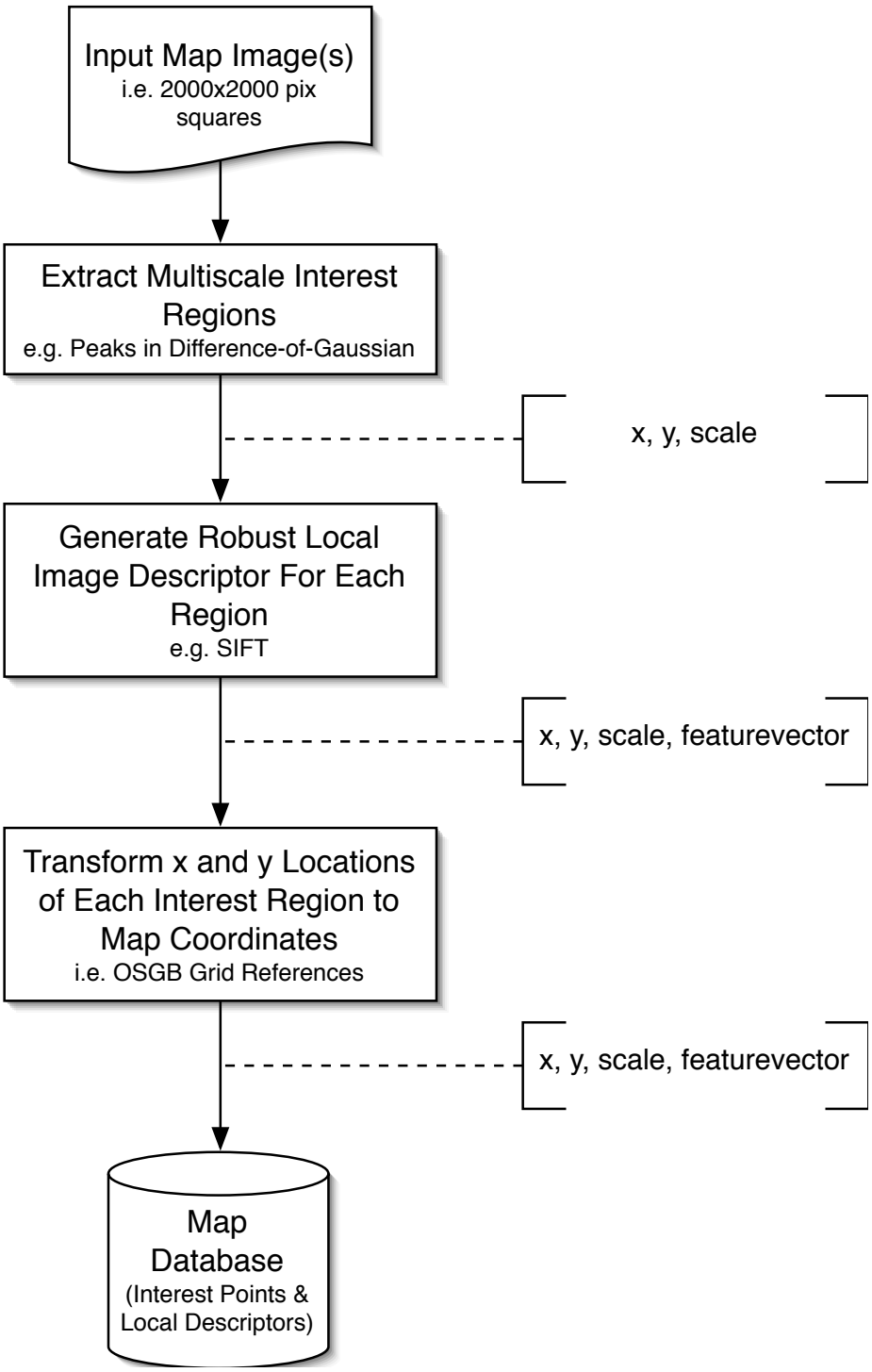

Figure 3. Offline Database Creation

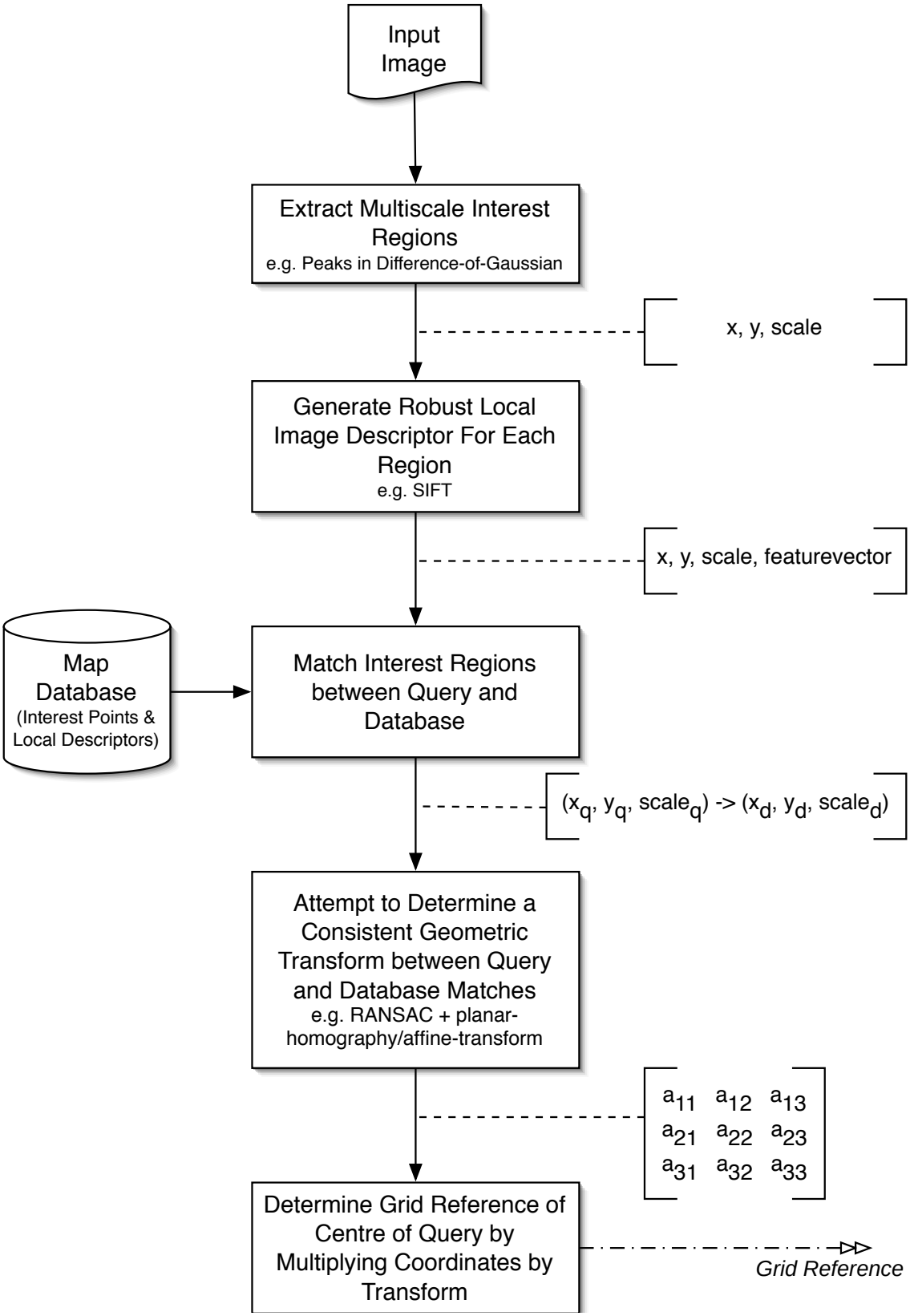

Figure 4. Online Querying

3. Determination of location of query in map coordinates: The geometric mapping is used to project the coordinates of the centre of the query image (i.e. half the pixel width and height) to the map coordinate system. This is achieved by multiplying the projection matrix with the coordinate vector.

3.1 First Prototype

The initial MapSnapper prototype was not very robust, however it did serve to demonstrate the promise of the proposed architecture. As previously mentioned, the interest points and local descriptors were extracted using Lowe's difference-of-Gaussian and SIFT techniques. The map database was stored in memory as a linked list of structures representing interest point location and scale, together with their descriptors.

3.1.1 Interest Point Matcher

The obvious way of matching an interest point between a query image and the database would be to find the minimum *distance* (e.g. Euclidean distance) between the descriptor of the query point and all of the points in the database. However, as noted by Lowe,[11] this doesn't work so well if the descriptors are not evenly distributed in feature-space.

Instead of taking this simple approach, the technique proposed by Lowe has been adopted. Matches are identified by finding the 2 nearest neighbours of each interest point from the query image among those in the map database, and only accepting a match if the distance to the closest neighbor is less than 0.8 of that to the second closest neighbour. It should be noted that this matching approach is brute-force; that is every interest point in the query image is compared to every interest point from the digitised map.

3.1.2 Geometric Consistency

Geometric consistency within the prototype was handled by using a robust estimation of a planar homography between the query image and the map. The planar homography is a non-singular linear relationship between points on two planes, that is, it maps points within the query image to points on the map. The homography has 8 degrees of freedom and is unique up to a scale factor.

Robust estimation of the planar homography was performed using the Random Sample Consensus algorithm[22] (Algorithm 1) coupled with estimation of the homography parameters using the standard Singular Value Decomposition (SVD) method.[23] Estimation of the homography parameters requires 4 pairs of interest point correspondences.

Algorithm 1: The RANSAC (Random Sample Consensus) algorithm

Given a fitting problem with parameters $\vec{x}$, estimate the parameters.
Assumptions:

- The parameters $\vec{x}$ can be estimated from N data items.
- There are M data items in total.
- The maximum number of allowed iterations is L. L can be estimated from knowing the probability of a randomly selected data item belonging to a good model, p_g and the probability the algorithm will exit without finding a good model if one exists, p_{fail}.

```
begin
    for l ← 0 to L do
        s⃗ ← Select N data items at random
        x⃗ ← Estimate model from s⃗
        K ← Determine how many data items (of M) fit the model with parameter vector x⃗ within a user
        given tolerance
        if K is big enough then
            accept fit and return success.
    return failure
end
```

3.1.3 Performance

The design of the system means that it is possible for the algorithm to fail before it can estimate a matching grid reference. Failures can occur at two points in the architecture; firstly, the matcher may fail to find sufficient matches (at least 4 are required for the geometric consistency stage), and secondly, the geometric consistency stage may itself fail if the RANSAC algorithm fails to find a consistent model. It is also possible, but very unlikely, that the system could fail at the interest-point detection stage by detecting no interest points within the query.

Table 1 shows a breakdown of the performance of the prototype system with the data-set. Failure rates are reported separately to the error rates. Error rates were calculated by counting the number of queries with an error over a given threshold. In addition, the percentage error rates are calculated only over the cases where a grid reference was returned by the algorithm (i.e. failures were ignored).

Table 1. Performance of the Initial Prototype

	Failures	Errors					
		> 2	> 5	> 10	> 20	> 30	> 50
Rate	53%	47%	47%	45%	34%	29%	18%

The results in Table 1 show that the system completely failed to find a match about 50% of the time, and had an error of more than two grid squares distance over 30% of the time. Obviously this performance is not satisfactory. It is interesting to look at the breakdown of results between the two differing camera phones. The breakdown of results is shown in Table 2. The results show that the performance in terms of number of failures is much better for the newer K750 phone, but localisation performance is worse (at least compared to the few T630 images that did not result in failure).

Table 2. Breakdown of the Performance of the Initial Prototype

	Failures	Errors					
		> 2	> 5	> 10	> 20	> 30	> 50
Rate (K750)	18%	82%	82%	76%	53%	41%	20%
Rate (T630)	79%	21%	21%	21%	19%	19%	16%

Overall, the average time taken to produce a non-failure result was around 48 seconds, whilst the time for a failure was a little less, at 34 seconds. The K750 photos took longer to process in both failure and non-failure cases. This is because the higher quality of imagery causes more interest points to be located, in turn causing more matches to be sought.

4. IMPROVEMENTS

The improvements to the algorithm in order to satisfy the original requirements were performed in two stages; Firstly, improvements were made to the retrieval robustness of the algorithm. Secondly, attempts were made to improve the computational performance of the algorithm, thus improving processing speed.

4.1 Robustness

Analysis of the prototype algorithm showed that most of the robustness problems were being caused by the inability of the software to find a good fit of the matched interest points with the geometric model. In order to improve the robustness of the algorithm, a number of different approaches were tried, but the final one basically involved changing the geometric model from a planar homography to an affine transform. Whilst the planar homography had been used in previous applications,[3] the affine transform proved to be a much better choice in the MapSnapper case due to the almost insignificant amount of out-of-plane rotation between the photographs and map.

Additionally, the RANSAC algorithm was modified slightly by adding an additional step. If the RANSAC algorithm found a good model, then the model was tested against all of the data in order to determine which data points were inliers and which were outliers. All of the inliers were then used to determine a better model using a least-squares technique.

Performance figures for the robustness-improved system are shown in Tables 3. As can be seem from the results, the newer K750 photos still have better performance, but both phones have a much better overall performance. It should be noted that the images that fell into the failure category actually proved very difficult to locate by hand in order to generate the ground truth.

Table 3. Breakdown of the Performance of the Robust Algorithm

	Failures	Errors					
		> 2	> 5	> 10	> 20	> 30	> 50
Rate (Overall)	13%	8%	8%	8%	7%	7%	6%
Rate (K750)	4%	4%	4%	4%	4%	4%	2%
Rate (T630)	16%	12%	10%	10%	9%	9%	9%

Computationally, the robust algorithm is actually more efficient than the prototype algorithm, with overall average run times of just under 40 seconds for both failure and non failure cases. The K750 photographs took on average about 50 seconds for non-failure and 30 seconds for failure. The longest K750 processing run took 73 seconds, but did result in an exact match. The reason for the improved performance is that with the improved geometric model, the RANSAC algorithm takes many fewer iterations and hence fitting attempts in order to locate a good match.

4.2 Speed

Most of the processing time in the robust algorithm is spent in determining the matches. A sizable amount of time is also spent in the interest point detection and local descriptor calculation, however, because off-the-shelf algorithms are being used, the only way to optimise these would be to optimise at source code-level, which was beyond the remit of the project.

4.2.1 Smaller Matching Sample

The first way to speed-up the algorithm is based on the statistics of the matching process. Empirically, the percentage of good matches by the nearest-neighbour matching approach discussed in Section 3.1.1 is over 70%. Using this knowledge, it is possible to attempt to find a good geometric model before all the matches have been located. In our implementation, the RANSAC geometry estimation is run as soon as 10 interest points have been matched. The RANSAC estimation will then attempt to build a geometric model that fits at least 7 of the matches. It should be noted that the ordering of the interest points in the query image is such that interest points with larger scales are matched first — these interest points tend to have more unique and thus better matching descriptors.

In the test database, this approach does not affect the robustness of the matching. However, because only a subset of the possible interest point matches are considered it does mean that the algorithm could be more easily tricked into generating a geometry where one does not exist. For example, if the system is queried with an image that comes from a different map to the one that has been indexed. Although not included in the current implemention, this situation could be remedied by adding constraints and sanity checking to look at the geometric transform and assess whether it is reasonable.

4.2.2 Scale-space Search

In the prototype and robust algorithm, the matching process is brute-force. This approach works, but, it does not scale particularly well, especially if the number of map-tiles is increased. An alternative approach is to employ a scale-space search to reduce the search space. Basically, the scale-space search works by finding an

initial seed point by brute-force matching of the interest point from the photograph with the largest scale against the entirety of the digital map, and then reducing the search area for subsequent matching to a smaller area around the initial seed match. There is of course no need to search scales bigger than the point in the digitised map either. In order to efficiently index the interest points in three dimensions (x-position, y-position, scale) a B-Tree data structure is used. The B-Tree enables the rapid extraction of a volume of scale-space.

In our implementation, the scale-space search is a little crude; basically a 200×200 region of map (corresponding to 20×20 grid squares) is extracted around the seed match and this is used for subsequent matching. Obviously, this is unlikely to be the most optimal approach, however it does work well with the test data.

4.2.3 Performance

The matching performance of the sped-up robust algorithm is shown in Table 4. Interestingly, the performance is actually better than the robust version. This is because the scale-space search limits outliers from far away from the matching region. All of the failure cases were either due to the algorithm being unable to find a seed match (i.e. there were no matches between the image and database interest points), or the matching algorithm finding less than the three matches required for affine transform estimation. The performance with K750 is particularly impressive; all but 2% of the errors occured with a distance less than 2 units. This is actually easily less than the error to be expected within the ground truth data itself.

Table 4. Breakdown of the Performance of the Sped-Up Robust Algorithm

	Failures	Errors					
		> 2	> 5	> 10	> 20	> 30	> 50
Rate (Overall)	10%	4%	3%	3%	2%	1%	1%
Rate (K750)	2%	2%	0%	0%	0%	0%	0%
Rate (T630)	16%	6%	6%	6%	3%	1%	1%

In terms of computational speed, the results are equally impressive. On average, the algorithm took about 5 seconds per image when failure did not occur. Failure cases took 17 seconds on average. The longest non-failure run for the K750 photographs was just over 14 seconds. On the whole, with the exception of a few outliers the non-failure K750 runs are all very close to 5 seconds, and the T630 runs are all close to 6 seconds. This is quite a difference from the 50 second or so runs with the robust version of the algorithm.

5. CONCLUSIONS

This paper has described the systematic approach taken to design a suitable algorithm for matching poor quality query images taken from a mobile phone against a high quality digital representation of a map. The design methodology was heavily evaluation-driven and involved many stages of incremental improvements in order to reach the final design. The outcome of this research is a fast, robust algorithm that meets the design criteria.

The matching algorithm combines a number of computer vision techniques, including interest point extraction and local descriptor generation with multidimensional indexing. Geometric constraints were applied to ascertain whether the interest-point matches are consistent.

ACKNOWLEDGMENTS

We would like to thank Ordnance Survey for providing funding and data for this work.

REFERENCES

1. J. Barton and T. Kindberg, "The challenges and opportunities of integrating the physical world and networked systems," Tech. Rep. HPL-2001-18, HP Labs, 2001.
2. K. Martinez, "Visual Linking." Multimedia Understanding — Fest Group 2001, 2001.

3. J. S. Hare and P. H. Lewis, "Content-based image retrieval using a mobile device as a novel interface," in *Proceedings of Storage and Retrieval Methods and Applications for Multimedia 2005*, R. W. Lienhart, N. Babaguchi, and E. Y. Chang, eds., pp. 64–75, SPIE, (San Jose, California, USA), January 2005.
4. M. Jia, X. Fan, X. Xie, M. Li, and W.-Y. Ma, "Photo-to-search: Using camera phones to inquire of the surrounding world," in *MDM '06: Proceedings of the 7th International Conference on Mobile Data Management (MDM'06)*, p. 46, IEEE Computer Society, (Washington, DC, USA), 2006.
5. C. Schmid and R. Mohr, "Local grayvalue invariants for image retrieval," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **19**(5), pp. 530–535, 1997.
6. J. S. Hare and P. H. Lewis, "Scale saliency: Applications in visual matching, tracking and view-based object recognition," in *Distributed Multimedia Systems 2003 / Visual Information Systems 2003*, pp. 436–440, Knowledge Systems Institute, (Florida International University, Miami, Florida), September 2003.
7. J. S. Hare and P. H. Lewis, "Salient regions for query by image content.," in *CIVR*, P. G. B. Enser, Y. Kompatsiaris, N. E. O'Connor, A. F. Smeaton, and A. W. M. Smeulders, eds., *Lecture Notes in Computer Science* **3115**, pp. 317–325, Springer, 2004.
8. J. S. Hare and P. H. Lewis, "On image retrieval using salient regions with vector-spaces and latent semantics.," in *CIVR*, W. K. Leow, M. S. Lew, T.-S. Chua, W.-Y. Ma, L. Chaisorn, and E. M. Bakker, eds., *Lecture Notes in Computer Science* **3568**, pp. 540–549, Springer, 2005.
9. D. Lowe, "Object recognition from local scale-invariant features," in *Proc. of the International Conference on Computer Vision ICCV*, pp. 1150–1157, (Corfu), 1999.
10. T. Tuytelaars and L. V. Gool, "Content-based image retrieval based on local affinely invariant regions," in *Third International Conference on Visual Information Systems*, pp. 493–500, 1999.
11. D. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision* **60**, pp. 91–110, January 2004.
12. K. Mikolajczyk and C. Schmid, "Indexing based on scale invariant interest points," in *International Conference on Computer Vision*, pp. 525–531, (Canada), July 2001.
13. K. Mikolajczyk, *Detection of local features invariant to affine transformations*. PhD thesis, Institut National Polytechnique de Grenoble, France, 2002.
14. J. S. Hare, *Saliency for Image Description and Retrieval*. PhD thesis, University of Southampton, 2005.
15. S. Obdrzálek and J. Matas, "Local affine frames for image retrieval," in *CIVR*, M. S. Lew, N. Sebe, and J. P. Eakins, eds., *Lecture Notes in Computer Science* **2383**, pp. 318–327, Springer, 2002.
16. J. Matas, O. Chum, M. Urban, and T. Pajdla, "Robust wide baseline stereo from maximally stable extremal regions.," in *BMVC*, P. L. Rosin and A. D. Marshall, eds., British Machine Vision Association, 2002.
17. K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. V. Gool, "A comparison of affine region detectors," *Accepted in International Journal of Computer Vision* , 2005.
18. N. Sebe, Q. Tian, E. Loupias, M. Lew, and T. Huang, "Evaluation of salient point techniques," *Image and Vision Computing* **21**, pp. 1087–1095, 2003.
19. M. A. Stricker and M. Orengo, "Similarity of color images," in *Storage and Retrieval for Image and Video Databases (SPIE)*, pp. 381–392, 1995.
20. W. Y. Ma and B. S. Manjunath, "A comparison of wavelet transform features for texture image annotation," in *ICIP '95: Proceedings of the 1995 International Conference on Image Processing (Vol.2)-Volume 2*, p. 2256, IEEE Computer Society, (Washington, DC, USA), 1995.
21. K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," in *International Conference on Computer Vision & Pattern Recognition*, **2**, pp. 257–263, June 2003.
22. M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," pp. 726–740, 1987.
23. E. Vincent and R. Laganière, "Detecting planar homographies in an image pair," in *Proceedings of the 2nd International Symposium on Image and Signal Processing and Analysis*, pp. 182–187, (Pula, Croatia), June 2001.