2013 IEEE International Conference on Robotics and Automation (ICRA)
Karlsruhe, Germany, May 6-10, 2013

Feature Co-occurrence Maps: Appearance-based Localisation Throughout the Day

Edward Johns and Guang-Zhong Yang
The Hamlyn Centre, Imperial College London

Abstract— **In this paper we present a new method, Feature Co-occurrence Maps, for appearance-based localisation over the course of a day. We show that by quantising local features in both feature and image space, discriminative statistics can be learned on the co-occurrences of features at different times of the day. This allows for matching at any time, without requiring individual images to be stored representing each time of day, and matching is performed efficiently by simultaneously matching to the entire database. We further show how matching along image sequences can be incorporated into the system and adapt existing methods by allowing for non-zero acceleration. Results on a 20km outdoor dataset show improved performance in precision-recall over state of the art.**

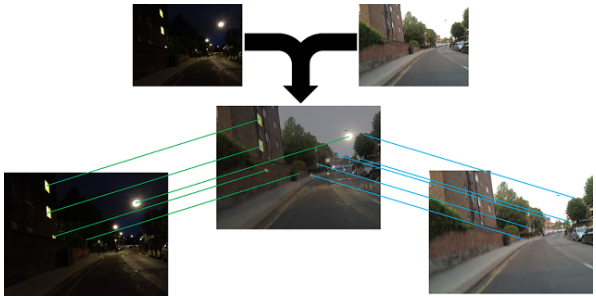
I. INTRODUCTION

A visual nagivation system is one of the core components of modern mobile robots and autonomous vehicles [5], due to the quantity of information available from a single image. One key part of this system is the requirement for a robot to determine when it is positioned, qualitatively, at a location it has previously visited. This has a wide range of applications, including global localisation and the "kidknapped robot" problem [8], loop closure for Simultaneous Localisation and Mapping (SLAM) systems [7], and topological navigation [9].

Typically, this is achieved by storing one or more images per location in the topological map [10], and adopting image retrieval techniques [1], [2], [3] based on local invariant features [18] which can be highly discriminant. For efficiency, the Bag-Of-Words (BOW) model [11] typically preceeds robust geometric feature matching [21], although weak spatial information can also be incorporated into the BOW model [2], [22]. Image retrieval is used to determine the most similar database image to a query image, and hence the most likely location from where the query image was captured. This approach has yielded promising results in large scale environments [10], [24] during the day and indoor scenes with repetitive scene appearances [20]. Due to the illumination invariance of features such as SIFT [18], promising results have also been seen in loop closure at multiple times during the day [23], [16].

However, the problem becomes more challenging when it is necessary to perform localisation during both the daytime and at night. As shown in the top of Figure 1, local features that appear during the day are often not present at the same location at night, due to the dramatic natural illumination change highlighting some features and shadowing others, together with other non-natural changes in scene appearance such as from street lights. Fusion of vision with range sensors can help to overcome this problem [25], but this poses further complications such as sensor calibration.

(a) Local feature correspondences across dramatic illumination changes are rare.



(b) Our system fuses together image data from multiple times of the day into one compact model

Fig. 1: The difficulty in finding local feature matches between day and night images is solved by our proposed solution.

In this paper, we propose a new method, Feature Co-occurrence Maps (Cooc-Map), that exploits the discriminative power of local features, and still allows for location recognition at multiple times of the day. This is achieved by learning the co-occurrence statistics of features in quantised feature space and quantised image space, and fusing together these statistics from different times of the day into one compact model for each location. Query images are then compared to each location model by finding groups of features in the query image that have also co-occurred, with the same spatial relationships, in one of the training images for that location. By training the location models with multiple images over the day and night, each location is represented by several sets of co-occurring features, each representing a different time of day.

A. Related Work

One solution to the problem of scene appearance changes over time, is to store multiple images representing each scene across its full range of appearances [26]. However, whilst methods exist to reduce the number of images into a compact set [28], it is still relatively inefficient in terms of both computational time and memory.

978-1-4673-5643-5/13/$31.00 ©2013 IEEE

Avoiding the use of local features can help to deal with the lack of local feature correspondences across strong changes in illumination. Such image comparison techniques have been proposed by using global image descriptors [12], learning the statistical properties of raw image pixels [13], comparing shape signatures [14] and learning discriminative spatial weighting of a range of image features [15]. However, none of these are discriminative enough to deal with large-scale navigation tasks. Fusing multiple images to represent a single scene model across all illumination conditions has been proposed [17], but this aims to "average out" the scene, rather than learning the time-dependent appearances of a scene.

Temporal information and odometry can help with loop closures by ensuring consistent matches across a sequence of images. The use of illumination-invariant local features combined with a filtering system has provided for localisation at varying times of the day [16], but not at night. Strong constraints on image sequence compatibility can avoid the need for local features altogether in SeqSLAM [4], inspired by the RatSLAM method [6], but this lacks the important discriminative power of the local features.

B. Key Contributions

In this paper, we present three key novelties and demonstrate their contribution towards the overall system, together with comparisons against two state-of-the art methods: FAB-MAP [10] and SeqSLAM [4].

- Existing techniques either store multiple images across day and night, or avoid the use of local features altogether. We show how discriminative local features can be used for matching across day and night images by learning feature co-occurrence statistics, without having to store a new frame for each location.
- We present a new geometric feature matching algorithm that ensures global consistency in feature geometric relationships and enables efficient simultaneous matching to all locations.
- The sequence-based matching in SeqSLAM requires the camera path to be similar across two sequences to be matched, and assumes a constant speed of vehicle across the sequence. We extend this by allowing for lateral shifts in camera location and showing how non-zero acceleration can be accounted for.

II. Method

In this section, we describe the theory behind our proposed system and the algorithms used in implementation.

A. Image quantisation

We adopt the Bag-Of-Words model [11] and quantise local features such that each is assigned to the closest *visual word* in feature space, with all visual words belonging to a single *visual dictionary*. We then quantise images spatially in a similar manner to [8], by dividing up images into regular grids of square *spatial words*, each belonging to a single *spatial dictionary*. Each pair of features is thus described by a pair of visual words and a spatial word describing the geometric arrangements of the two features, as in Figure 2.

Whilst loss of information naturally arises during both these quantisation steps, the simplified image representation allows for a powerful statistical model to be developed that would not be possible with raw continuous image data. Furthermore, significant efficiency gains are achieved by avoiding the need for direct feature-to-feature comparisons; finding features with the same visual word, or two features with the same spatial word, can be achieved by use of a simple lookup table.

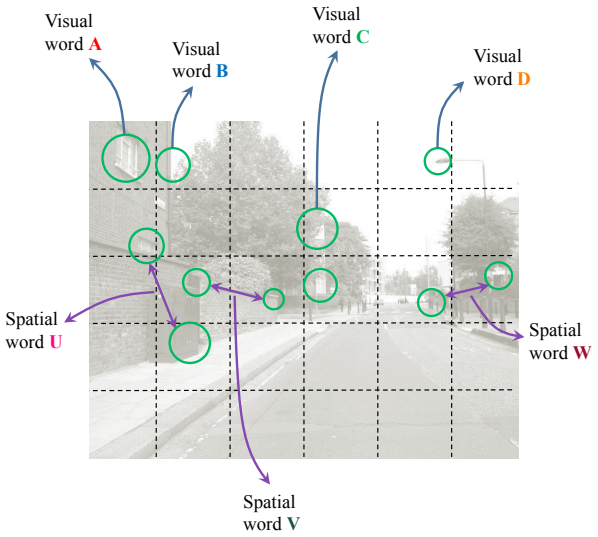


Fig. 2: Quantisation of features in descriptor space and image space. Features are assigned to their closest visual word, and feature pairs are assigned to their closest spatial word.

B. Location models

We consider a topological map within which navigation occurs, consisting of a linear chain of locations such as in Figure 3. We do not perform any quantitative geometric localisation, bundle adjustment or SLAM, and we are concerned only with which location in an existing topological map does a query image represent.

At the i^{th} node in the topology, location $y_i \in Y$ is represented by a *word-co-occurrence matrix* M_i, which stores the co-occurrence statistics of visual and spatial words for the location. Given an image x_i captured at this location, each row in M_i represents a different visual word that has been detected in x_i. The elements along this row then represent the spatial co-occurrence statistics for this visual word and all other visual words.

The elements of M_i are generated as follows. For each pair of features observed in x_i, the corresponding spatial word observed is recorded in the matrix element representing the visual words for the two features. For example, suppose that two features detected in x_i are assigned to visual words V and W respectively, and spatial word S. Then, matrix element M_i at (V, W) is assigned to the value S. Note

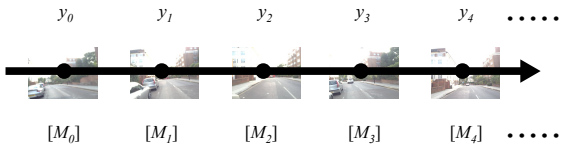


Fig. 3: Each node i in the topological map represents a location y_i, whose appearance is modelled by a word-co-occurrence matrix M_i

that only those visual words that actually appear in x_i are represented in the matrix. This is for memory efficiency, as typically an image will contain only a very small fraction of visual words in the dictionary. A lookup table is used to relate each row of M_i to the respective visual word.

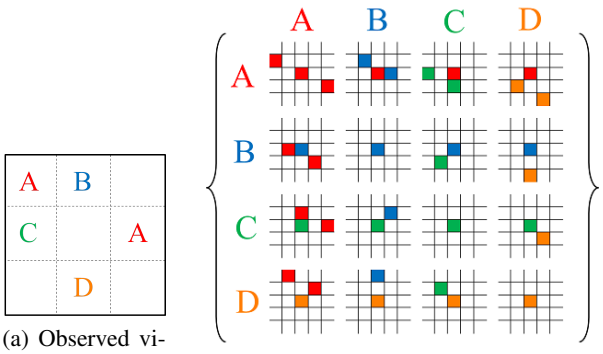
Thus far, M_i only represents the appearance of location y_i at one instance in time. The ability of our system to deal with illumination changes comes into play as the co-occurrence statistics are updated through further images captured of location y_i at different times of the day. Given one of these images, pairs of visual words and their associated spatial words are used to update M_i accordingly. If any observed visual word is not already represented in M_i, it is added along with the corresponding spatial word. For each visual word pair, it is possible to have multiple associated spatial words due to more than one feature in an image being assigned to the same visual word. As such, elements in M_i are not single spatial words, but a set of spatial words that have been assigned to the associated visual word pair.

Figure 4 demonstrates the construction of the word-co-occurrence matrix in a simplified example. In Figure 4(a), extracted features are assigned to visual words A - D, and each pair of features is assigned to a spatial word. In Figure 4(b), the word-co-occurrence matrix is described visually, with each element showing the distribution of spatial words for that visual word pair that has been observed. For example, visual word A has appeared to the left, and also to the bottom-right, of visual word B. Hence, the matrix element at (B, A) shows the respective range of spatial words for these co-occurrences.

The word-co-occurrence matrix thus provides a compact representation of the appearance of a location at many different times during the day. Its power in providing for robust location recognition comes from considering that certain visual words co-occur at different times of the day. During the day, there may be an entirely different set of co-occurring visual words to that of during the night, but these statistics are all maintained in a single compact model of the location, without requiring each individual image to be retained. Recognition of a query image then proceeds by finding groups of co-occurring visual words that have also co-occurred as a group in the training images.

C. Location recognition

Given a query image x_q, local features are extracted and quantised, as before. Using an inverted-file index as is typical in image-retrieval tasks [11], each query visual word points



(a) Observed visual words in quantised image space

(b) Matrix M representing co-occurrence statistics for pairs of visual words

Fig. 4: Construction of the word-co-occurrence matrix M for each location. Each element (i, j) represents the observed distribution of spatial words between the respective visual words represented by row i and column j

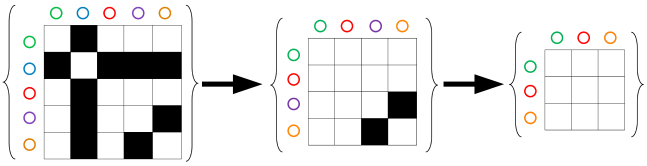
to all database locations which contain that visual word in the word-co-occurrence matrix for that location. Candidate correspondences between features in the query image and features in the database location are then generated for each location.

For each database location y_i, a correspondence-consistency matrix J_i is created, which represents the spatial consistencies between pairs of feature correspondences. With n candidate correspondences between x_q and y_i, the matrix is of size $n \times n$. Each element is of binary value, reflecting whether or not a pair of correspondences is consistent with the data in the word-co-occurrence matrix M_i. For example, if two features in the query image are assigned to visual words A and B, and spatial word S, then the relevant element of M_i is found, relating the spatial relationships between A and B in y_i. If this element contains spatial word C, then the element of K at location (A, B) is set to 1; otherwise, it remains at 0.

The task then becomes to find the subset K of feature correspondences represented in J_i, which contains correspondences that are all spatially consistent with each other. This ensures global spatial consistency over the image - two correspondences in the query image that do not match spatially to the database location cannot both exist in J_i. In this way, we are finding groups of features in the query image that have all co-occurred together in one of the training images for this location.

The subset K_i is generated as follows. Each candidate correspondence c_i is assigned a score equal to the sum of the elements in row i of J_i, i.e. the total number of correspondences in J_i that are spatially consistent with c_i. Then, we iteratively eliminate correspondences in J_i in inverse order of the number of the correspondence score. After every iteration, the score for each correspondence c_i is updated, by subtracting 1 from the existing score if the previously eliminated correspondence was spatially consistent with c_i (and thus previously contributed to the score).

(a) Feature correspondences



(b) Matrix evolution from J to K

Fig. 5: Evolution of the Correspondence-consistency matrix from inital matrix J containing all the correspondences, to final matrix K, containing only correspondences that are spatially consistent with all others.

If two correspondences have an equal score, then one is chosen at random for elimination. This is a rare case, but typically both correspondences will be eliminated eventually and so the random choice is arbitrary. The algorithm stops when all the elements in J_i are 1, and this is our subset K_i. In this way, we eliminate the correspondences in order of how likely they are to be an incorrect correspondence, until we are left with a set K_i representing a group of correspondences which has global spatial consistency. Finally, the similarity score between query image x_q and database location y_i is equal to the size of the matrix J_i, i.e. the number of correspondences in the spatially-consistent group.

Figure 5 demonstrates the evolution of the correspondence-compatibility matrix from J to K. In (a), candidate feature correspondences are found between the query image and the database location. In (b), correspondences in J are iteratively eliminated until the matrix has 1's in every element. Each colour represents a candidate correspondence. Here, the blue correspondence is not spatially consistent with any others, and so is eliminated first. The purple correspondence is spatially consistent with the green and red correspondences, but not with the orange correspondence, and so it is the next to be eliminated. The final matrix J consists only of the red, green and orange correspondences, which form a group that is globally spatially consistent.

D. Simultaneous location matching

The location recognition algorithm can be dramatically sped up by matching the query image to all locations simultaneously. With most existing feature-based approaches to image matching, it is necessary to fully complete the matching algorithm for each image to compute a similarity score. In our method, however, the iterative shrinking of matrix J allows for a calculation of the maximum possible score for a location, and we exploit this to reduce the overall computational load.

For each location y_i, we keep a record of this maximum potential score, which is equivalent to the current number of correspondences in J_i. On each loop of the algorithm, one correspondence is removed from each J. When J consists of only 1's for any location, then the score for this location is recorded as the maximum score thus far. Then, on the next loop, if the maximum potential score for any location is less than this maximum score, we cease to consider this location any further because it cannot grant us the best similarity score.

Most incorrect locations will have many more 0's than 1's in J, and as such would otherwise require a significant number of loops in the algorithm before all the necessary correspondences in J are eliminated. In our experiments, this provided a speed up factor of around 80 compared to the alternative where every location is considered sequentially and independently. Geometric image matching (following a Bag-Of-Words stage, for example), is typically linear in the number of images to consider, whereas we achieve computational time that is substantially sub-linear.

E. Interpolating between different times of day

Learning the location models over multiple tours provides a solid understanding of location appearances at these specific times. The appearance of locations between these times is, however, not explicit, but we can estimate these appearances by interpolating between those images we do have. This helps to allow for performance at any time during the day, and not just around the times at which the training tours were captured.

We achieve this by merging tours that are adjacent in time, and updating the word-co-occurrence matrix M based on the merged set of features for each location. Consider two training images $x_i^{t_0}$ and $x_i^{t_2}$ captured at location i at times t_0 and t_2 respectively. Due to the smooth nature of natural illumination changes, we can assume that any feature that occurs in both $x_i^{t_0}$ and $x_i^{t_2}$ is also likely to appear in the hypothetical image $x_i^{t_1}$ captured at time t_1. A feature that occurs in $x_i^{t_0}$ but not in $x_i^{t_2}$ may also still appear in $x_i^{t_1}$, up until a time just before t_2. Merging the co-occurrence statistics at t_0 and t_2 in this way maximises the recall of feature correspondences at all times between t_0 and t_2.

Matrix M is then created by merging co-occurrence statistics between all images captured in adjacent tours. For example, the statistics from the tour at 4pm are merged with the tour at 2pm to create one set of "virtual" co-occurrence statistics, and also with the tour at 6pm to create another set. Finally, each virtual set of statistics then combined in M to provide statistics representing all illumination conditions throughout the day and night.

F. Matching to sequences

Rather than matching a query image to a single location in the map, the SeqSLAM method [4] matches a sequence of query images to a sequence of database locations and ensures consistency in appearance across the sequence. To achieve this, they create a dissimilarity matrix which stores

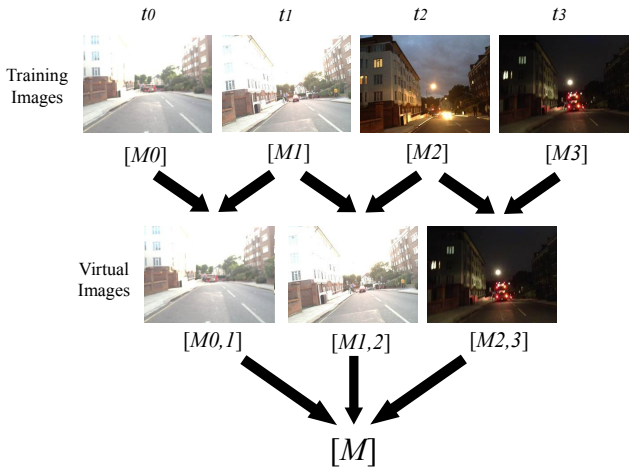


Fig. 6: Interpolating between training images by fusing word co-occurrence matrices allows for an estimation of scene appearance in the intermediate time.

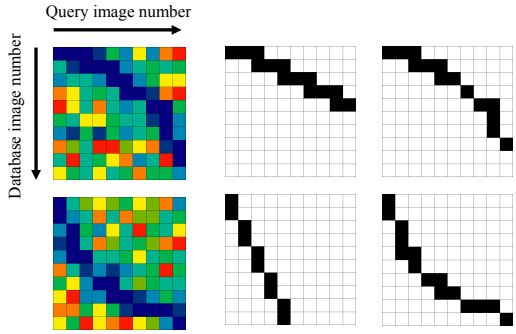
dissimilarity scores between each image in a query sequence and each in the entire database sequence. The method proceeds by finding the best path through this matrix, with a path representing pairs of image matches. The best path is the one that has the smallest sum of dissimilarity scores across all the image match pairs in the sequence.

One of the assumptions of the SeqSLAM method is that there is zero acceleration across the sequence, with only a linear path allowed through the dissimilarity matrix. However, non-zero accelerations are common, particularly in busy city centres where traffic lights and junctions disrupt the smooth flow of traffic. As such, we consider all forward-moving paths through the matrix, such that the database locations may be skipped or repeated if the query image sequence is accelerating or decelerating, respectively, relative to the database sequence. This is achieved algorithmically by creating a path that chooses the image pair with the lowest dissimilarity at each step, with no other constraints on the path shape other than that it must not reverse. This simple method almost always provides the global optimum path, without the computational overhead of an optimisation step.

Figure 7 demonstrates the difference between our approach to sequence matching and that of SeqSLAM. By allowing for a non-linear path through the dissimilarity matrix, a better fit between the query sequence and the best-matching database sequence. Other than allowing for non-zero acceleration, the same parameters as in SeqSLAM were used to match image sequences.

III. Experiments

In this section, we describe the datasets and implementation details for testing our system and comparing to the state-of-the-art.



(a) The image dissimilarity matrix. Blue represents high similarity.

(b) SeqSLAM - zero acceleration

(c) Adapted for non-zero acceleratio

Fig. 7: In the top row, the query sequence is accelerating relative to the database sequence, and in the bottom row, it is decelerating. By allowing the path through the dissimilarity matrix to take the optimum non-linear route, a more accurate correlation across the sequences can be achieved.

Fig. 8: Sample images from our dataset showing the same location observed at different times of day.

A. Dataset

Whilst there are existing datasets available consisting of image sequences captured at different times of the day [4], these typically contain only two sets of sequences, one during the day and one at night. Our system requires several tours of the same route at multiple times of the day to capture the full range of illumination conditions. As such, we acquired a new dataset consisting of a 20km path along roads in and around an urban neighbourhood, with one image captured every second, for tours of around 2200 images each. Images were captured from a standard colour camera mounted on a car dashboard, at a rate of one image per second, and with a resolution of 640×480.

A total of 6 tours were completed of the route during the same day, with starting times of 2pm, 4pm, 6pm, 7pm, 8pm and 10pm. A greater density of tours was captured around 7pm as this was the time of sunset when scene appearances change dramatically. Figure 8 shows example images of the dataset at the various times that tours were completed.

B. Implementation Details

SIFT features [18] were chosen as our local features, although many other examples exist that would be suitable and can be substituted into the system with ease. A visual dictionary of 10K words was constructed using an approximate k-means method [1], with a separate set of urban images acquired to train the dictionary. Images were spatially quantised into $12 \times 8 = 96$ squares in image space. Features can appear both above and below each other, and to the right and left of each other, and as such the size of the spatial dictionary is $96 \times 4 = 384$ spatial words.

C. Experimental Procedure

To evaluate the Cooc-Map method's ability to perform loop closure and global localisation at different times of the day, we tested the recognition system by using one of the tours for testing and all the others for training. The tours for 4pm, 6pm, 7pm and 8pm were used as queries, leaving out 2pm and 10pm because the system requires training tours to be completed before and after the time of the query tour, in order to benefit from the interpolation stage.

We tested our system against two benchmarks. First, the FAB-MAP [10] method, a probabilistic appearance-based approach that approximates the joint probabilistic distribution of visual word observations. Second, the SeqSLAM method [4], which directly uses image intensities and exploits sequence matching to allow for recognition of the flow of scene structure at different times of the day. We perform global localisation based on the map described by the training images by ranking locations in order of their respective scores.

Our method was tested both with and without the sequence-matching stage. In the former, the location with the highest score to the query image is returned as the best match. In order to generate the precision-recall curves, we adjusted the minimum location similarity score required for the location to be considered a match, i.e. the minimum number of feature correspondences in the spatially-consistent groups of matrix J (see Section II C). A true positive match was recorded if the best matching database location is within 20 metres of the location of the query image.

IV. Results

Figure 9 shows qualitative results for an example query image where our method succeeds, but both SeqSLAM and FAB-MAP fail. FAB-MAP uses discriminant local features as in our work, but lacks the ability to learn location appearances for multiple times of day, without the need to accumulate several images for each location which is highly inefficient. SeqSLAM allows for matching across strong illumination variations, but lacks the discriminant local features necessary to deal with recognition in large-scale environments.

Figure 10 shows the performance of the Cooc-Map system when compared to the competitors. Each graph shows the precision-recall properties for loop closure at four different times of the day. For Cooc-Map, each of the other five tours

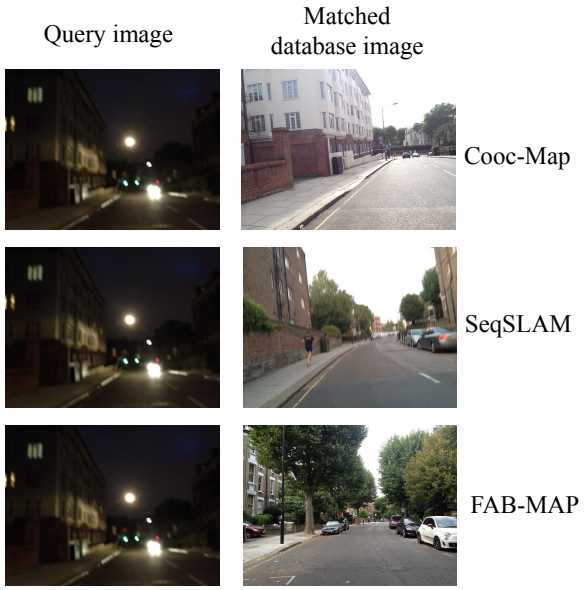


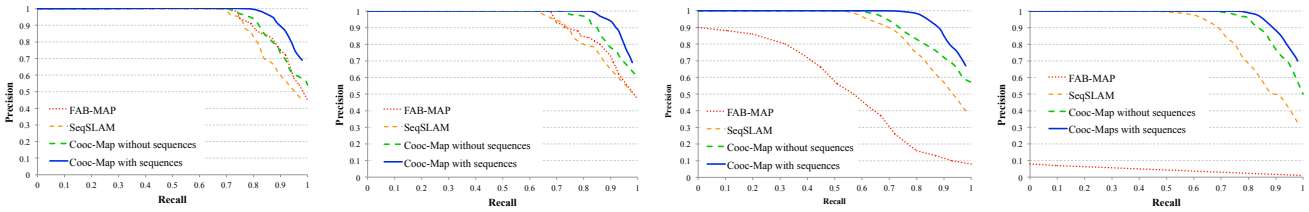
Fig. 9: Cooc-Map often provides correct location matches when SeqSLAM and FAB-MAP fail. This is due to the use of discriminative local features lacking in SeqSLAM, whilst learning a compact model across all illumination conditions which FAB-MAP lacks.

was used to train the system. For FAB-MAP and SeqSLAM, training is only carried out from one tour, and so we use the tour at 2pm as the database tour to which matches are attempted.

Our new method outperforms both FAB-MAP and SeqSLAM, even when sequences are not incorporated, and loop closure is performed with only a single image. This is especially promising because the requirement of matching to sequences is prone to complications when objects moving in the scene (for example, other vehicles), significantly affect the overall optical flow in a sequence of images. The performance of FAB-MAP naturally degrades as time progresses, particularly after sunset, which occurred at around 7pm. SeqSLAM also drops in performance marginally as the time between the database tour and query tour increases. However, Cooc-Map does not degrade after dark. The presence of the word co-occurrence statistics at nighttime are already present in the location model and so location recognition can be performed at high recall and precision.

The use of local features in Cooc-Map, rather than the raw image intensities as in SeqSLAM, means that lateral shifts along the vehicle's path do not cause problems. Cooc-Map considers only the spatial relationships between feature pairs - not their absolute position on an image. As such, deviations from a previous path, such as when driving along a motorway, can still be successfully localised in our method, but provides difficulties for SeqSLAM, where overlapping the two images shows little consistency. This is demonstrated in Figure 11.

Finally, the Cooc-Map method we have presented per-



(a) 4pm (b) 6pm (c) 7pm (d) 8pm

Fig. 10: Precision-recall performance of our system and competitors.

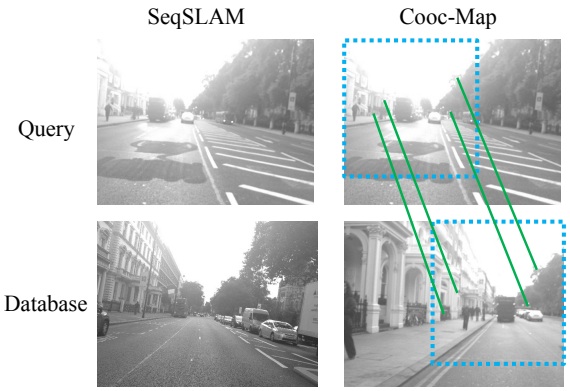


Fig. 11: Lateral shift in vehicle location causes SeqSLAM to fail, but Cooc-Map to succeed.

forms favourably at an overall rate of 82ms per localisation on the 20km path (excluding feature extraction, including sequence matching). This efficiency arises due to the compact location models that discretise image features, allowing for rapid image comparisons, and also the ability to perform image comparisons simultaneously over the entire tour.

V. Conclusions

In this paper, we have presented a new method, Feature Co-occurrence Models, for appearance-based localisation across different times of the day. We have shown that by quantising local features in both feature and image space, discriminative statistics can be generated on the co-occurrences of features at different times of the day. This allows for matching at any time, without requiring individual images to be stored representing each time of day. Furthermore, we have shown that allowing for non-zero acceleration in a sequence matching stage enables matching to sequences when a vehicle exhibits significantly different motion to when the training tour was captured.

References

[1] J. Philbin, O. Chum, M. Isard, J. Sivic and A. Zisserman: "Lost in quantization: Improving particular object retrieval in large scale image databases", in Proc. CVPR 2008

[2] G. Tolias and Y. Avrithis: "Speeded-up, relaxed spatial matchin," in Proc. ICCV 2011

[3] R. Raguram, C. Wu, J. M. Frahm and S. Lazebnik: "Modeling and recognition of landmark image collections using iconic scene graphs", in Trans. IJCV 2011

[4] M. J. Milford and G. F. Wyeth, "SeqSLAM: Visual Route-Based Navigation for Sunny Summer Days and Stormy Winter Nights," in Proc. ICRA 2012

[5] A. Geiger, P. Lenz and R. Urtasun: "Are we ready for Autonomous Driving? The KITTI Visual Benchmark Suite", in Proc. CVPR 2012

[6] M. Milford and G. Wyeth: "Persistent Navigation and Mapping using a Biologically Inspired SLAM System", in Trans. IJRR 2010

[7] A. J. Davison, I. Reid, N. Molton and O. Stasse: "Real-Time Single Camera SLAM", in Trans. PAMI 2007

[8] E. Johns and G.-Z. Yang: "Global Localisation in a Dense Continuous Topological Map", in Proc. ICRA 2011

[9] D. Filliat: "Interactive learning of visual topological navigation", in Proc. IROS 2008

[10] M. Cummins and P. Newman: "FAB-MAP: Probabilistic localization and mapping in the space of appearance", in Trans. IJRR 2008

[11] J. Sivic and A. Zisserman: "Video google: a text retrieval approach to object matching in videos", in Proc. ICCV 2003

[12] A. Oliva and A. Torralba: "Modeling the Shape of the Scene: A Holistic Representation of the Spatial Envelope", in Trans. IJCV 2001

[13] K. Ni, A. Kannan, A. Criminisi and J. Winn: "Epitomic Location Recognition", in Proc. CVPR 2008

[14] E. Shechiman and M. Irani: "Matching Local Self-Similarities across Images and Videos", in Proc. CVPR 2007

[15] A. Shrivastava, T. Malisiewicz, A, Gupta and A. A. Efros: "Data-drive Visual Similarity for Cross-domain Image Matching", in Proc. SIGGRAPH Asia 2011

[16] A. Glover, W. Maddern, M. Milford and G. Wyeth: "FAB-MAP + RatSLAM: Apperance-based SLAM for Multiple Times of Day", in Proc. ICRA 2010

[17] E. Johns and G.-Z. Yang: "From Images to Scenes: Compressing an Image Cluster into a Single Scene Model for Place Recognition", in Proc. ICCV 2011

[18] D. Lowe: "Distinctive Image Features from Scale-Invariant Keypoints", in Trans. IJCV 2004

[19] A. Kawewong, N. Tongprasit, S. Tangruamsub and O. Hasegawa: "Online and Incremental Apperance-based SLAM in Highly Dynamic Environments", in Trans. IJRR 2011

[20] A. Angeli, S. Doncieux, J.-A. Meyer and D. Filliat: "Real-Time Visual Loop-Closure Detection", in Proc. ICRA 2008

[21] J. Philbin, O. Chum, M. Isard, J. Sivic and A. Zisserman: "Object retreival with large vocabularies and fast spatial matching", in Proc. CVPR 2007

[22] Y. Zhang, Z. Jia and T. Chen: "Image Retrieval with Geometry-Preserving Visual Phrases", in Proc. CVPR 2011

[23] C. Valgren and A. Lilienthal, "Sift, surf, and seasons: Long-term outdoor localization using local features", in Proc. European Conference on Mobile Robots 2007

[24] G. Schindler, M. Brown and R. Szeliski: "City-Scale Location Recognition", in Proc. CVPR 2007

[25] J. A. Castellanos, J. Neira and J. D. Tardós: "Multisensor fusion for simultaneous localization and map building", in Trans. Robotics and Automation, 17(6), 2002

[26] Y.-T Zheng *et al*:" Tour the World: building a web-scale landmark recognition engine", in Proc. CVPR 2009

[27] R. Raguram, C. Wu, J.-M. Frahm and S. Lazebnik: "Modeling and Recognition of Landmark Image Collections Using Iconic Scene Graphs", in Trans. IJCV, 95(3), 2011

[28] E. Johns and G.-Z. Yang: "Dynamic Scene Models for Incremental, Long-Term, Appearance-Based Localisation", in Proc. ICRA 2013