Towards illumination invariance for visual localization

Ananth Ranganathan
Honda Research Institute USA, Inc
Mountain View, CA 94043
www.ananth.in

Shohei Matsumoto
Honda R&D Co. Ltd.
Wako-shi, Saitama, Japan
Shohei_Matsumoto@n.f.rd.honda.co.jp

David Ilstrup
Honda Research Institute USA, Inc
Mountain View, CA 94043
dilstrup@honda-ri.com

***Abstract*—While a large amount of work exists in the literature relating to place/location recognition, very few of these provide a robust way of dealing with large amounts of lighting changes in locations of interest. In this paper, we address the problem under the additional constraint that a pose estimate from the current location of the camera to the reference location is to be estimated. This requires robust feature matching to estimate corresponding points, and not just image-level matching, as is often done in the literature. We present a method to learn a matching function from training data that is representative of the lighting variations to be modeled, under weak assumptions. Lighting variation in the image descriptors is modeled using a probability distribution on the discretized descriptor space. Results are presented on a live visual SLAM system in outdoor environments and in an indoor simulated environment, which demonstrate the efficacy of the proposed method.**

Fig. 1. Images from the same location under different lighting conditions from the Maude loop dataset (explained in the experiments section)

I. Introduction

This paper deals with localization on maps constructed using visual simultaneous localization and mapping (VSLAM). These maps consist of sparse point clouds corresponding to features detected in the images and can be constructed using various algorithms such as FrameSLAM [1], MonoSLAM [2], and others [3], [4]. Localization in VSLAM maps is performed in two steps. The first step is location recognition wherein the image closest to the current image by appearance is retrieved from the map. If only qualitative localization is required, then nothing else needs to be done [5]. However, if exact metric localization is required, the second step is to compute the pose difference or rigid transformation between the map and the query image.

In this paper, we deal specifically with the problem of lighting change in visual localization. The main reason this is a challenge is due to the use of feature detection and descriptor based matching. Reliable localization requires most of the same features to be detected in the query image as exist in the map. However, common types of features such as corners and affine-invariant regions are not fully invariant to sharp changes in lighting such as direct sunlight and shadows (Figure 1). Additionally, even if the same features are being detected, the descriptor values change significantly making image matching and finding feature correspondences challenging.

We provide a method to localize under changing feature descriptor values. Following [6], this is done by learning a probability distribution, using a data-driven method, over the space of discretized feature descriptors. Our contribution through this paper is two-fold. First, we apply the work of [6], henceforth called "fine vocabulary", to the problem of location recognition under lighting change and present results on a live system under various environments. Second, we extend the fine vocabulary algorithm to also compute feature correspondence using a modified matching metric that takes into account the probability distribution over descriptors.

In experiments in outdoor and indoor environments using a real-time VSLAM system, we demonstrate that the use of a fine vocabulary significantly increases the frequency and accuracy of image-level location recognition. Our new feature matching method also makes pose estimation more accurate by providing a higher number of correct feature correspondences.

In the remainder of the paper, we first detail the dataflow for location recognition used by us. This involves the use of a vocabulary tree and is a standard approach in the state of the art. Following this, we describe the fine vocabulary algorithm and it's use for lighting invariant localization. We also describe here the extensions to the basic algorithm to accommodate feature matching. Finally, we provide detailed experiments to validate our algorithms and discuss our results.

II. Visual Localization

Our location recognition and pose estimation method roughly follows approaches such as [7], [8], which are standard in the state of the art. Descriptors are calculated at the site of features currently belonging to a set of inliers used for visual odometry [9], [10]. This set of descriptors is treated

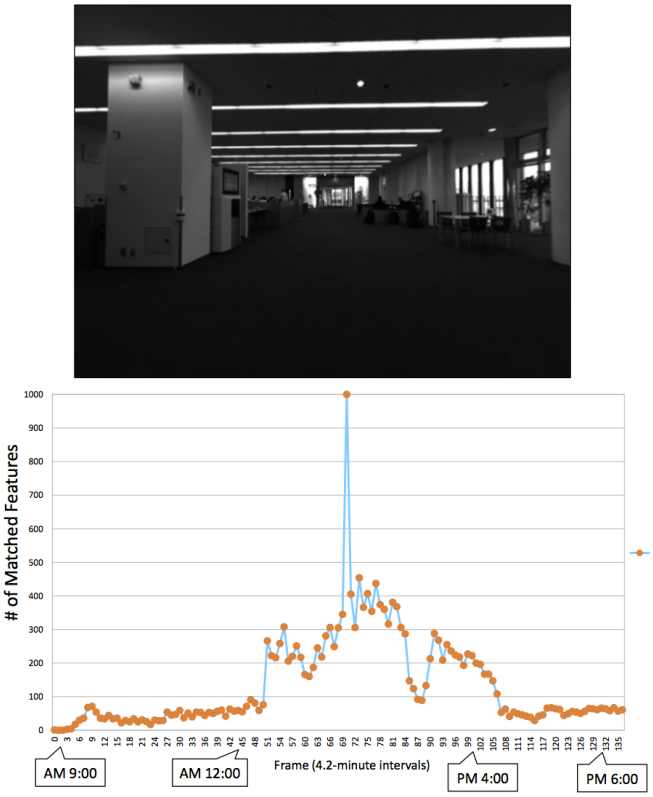

Fig. 2. Effect of lighting on feature matches: (Top) Scene on which time-lapse imagery is captured for 24hrs at approx. 5min intervals. (Bottom) Number of matched features using descriptor distance over time. The reference image is from about 2pm.

as a 'bag of words'; a model where the spatial location of the features is lost and only statistics in the form of a TF-IDF (term frequency - inverse document frequency) weighted histogram of *discretized descriptors* (words) represents the image. Discretization is performed using a standard clustering algorithm. Hierarchical k-means has proven to be efficient for this purpose. Clustering is done on a large set of descriptors computed from a representative set of images. Each Feature descriptor computed on an image is discretized as its matching cluster and a histogram over the clusters is obtained, thus representing the image as a bag of words.

Image-level localization is performed by comparing histograms of the query image and the reference images in the map [11]. Various metrics for the comparison are possible, the most popular being the L1 and L2 norms. Histogram comparison of the query image to reference images can be performed efficiently, in constant time with respect to the number of reference images, using the vocabulary tree data structure [11], which maintains an inverted file of the reference images containing at least one descriptor belonging to a particular cluster. Image level localization is accomplished by declaring the current location to be the same as the location of the reference image closest to the query image according to the histogram metric.

A. *Pose estimation*

The above procedure only provides rough qualitative localization since the reference and query images will rarely have been taken from exactly the same location. More exact localization is necessary for precise robot navigation indoors and for outdoor applications such as autonomous driving on roads where lane-level location may be desirable.

To compute relative pose between the reference and query images, correct feature correspondences must be established between the two. This is essentially the wide-baseline stereo problem [12], [13], yet in our setting considerable information is already available in the form of corner features and descriptors computed at the site of each such feature. Using this information, we focus on descriptor distance as a method of determining putative correspondence between points [14].

For descriptor matching to work at all, the feature detector must be *repeatable* under the changing conditions we are seeking to support; that is, the detector must continue to find image locations that correspond to persistant physical points in the scene. Corner detectors ([15], [16]), have been found to perform well in this regard ([17]), which justifies their use in this context.

Once feature correspondences have been obtained, the relative pose between the two images is computed using a 2D method such as the 8-point algorithm [18], or when the 3D locations of the feature points are available the 3-point algorithm [19] can be used. In practice, to increase robustness, putative matches are used within a RANSAC framework [20], with the model being the estimated pose and the model score being provided by the number of inliers.

The principal difficulty we address is that even though these descriptors possess robustness to changes in viewpoint and illumination, under large changes the distance between descriptor instances grows too large to discriminate true and false matches with any measure of success.

B. *Problems caused by lighting change in visual localization*

If the map is created at a certain time and localization is attempted at another time, problems arise due to the local nature of feature detection and descriptor computation. Localization may deteriorate or fail due to two reasons -

1) Features detected on the reference image may not be detected on the query image
2) Same features may be detected but descriptor values may have changed

The first cause, features not being detected, is usually caused due to sharp shadows, direct sunlight or other drastic lighting changes. In these conditions, local feature detectors cannot provide reliable localization and the procedure detailed above fails. Aside from using feature types known to be highly repeatable, we do not address this failure case in this paper.

The second cause for failure is the main focus of this paper. Since local descriptors capture characteristics such as local gradient of the image, a non-uniform lighting change affects the descriptor value. Further, this change depends on the nature

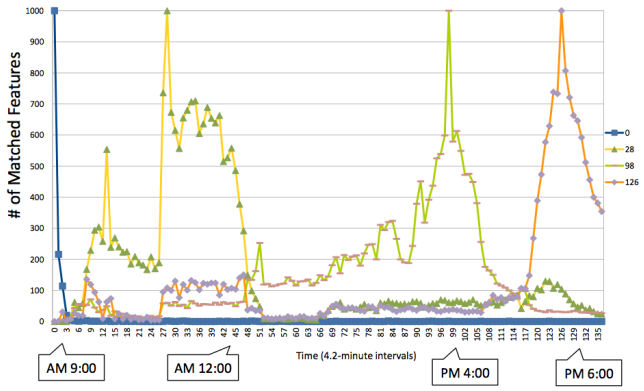

Fig. 3. Effect of lighting with reference images at four different times of the day for the 24hr dataset. Note that number of matches for the reference images in the early morning and evening fall of rapidly due to quickly changing lighting conditions.

of the scene and physical object on which the feature lies, and hence, is not generally predictable. For example, descriptors corresponding to features at the corners of a leaf and a piece of green cloth waving in the wind respectively, may match at some times of the day but will have very different values at other times depending, for instance, on the shadows cast by other parts of the tree on the leaf.

We describe some preliminary experiments to quantify the effect of lighting change. The first experiment was performed on data collected from a stationary camera which captured an image approximately every 5 minutes over a period of 24 hours in an indoor environment with large windows and artificial lighting. The number of feature matches over time is shown in Figure 2. Feature matches are evaluated by counting the number of matches that correspond to the groundtruth (same pixel location in this case) between images at two different times. In the figure, the reference image is taken around 2pm. The number of correct feature matches reduces drastically with time and this effect is solely due to the lighting in this case. Note that this result is from a relatively protected indoor environment with large planar surfaces and almost no self-occlusions. In outdoor environments with complex shapes such as trees, the drop-off in feature matching can be expected to be much higher. We used OpenCV implementations of GFTT features and SURF descriptors. The parameters are set so as to obtain 1000 features per image, and since many of these will have a *low* corner response they are not very repeatable, accentuating the dropoff.

A graph of the above experiment with four reference images is shown in Figure 3. The effect of lighting on feature matches is much more pronounced in the morning and evening as expected. Note that for consistent pose estimation, we would like around 30 to 40 features to be matched between the query and reference images. To achieve this, we would need to combine reference images from at least four different times. Thus, map building would require four runs at different times for each location resulting in an onerous procedure.

The result of correct feature matches over a similar dataset collected over 50 hours, with one image every 5 minutes, is

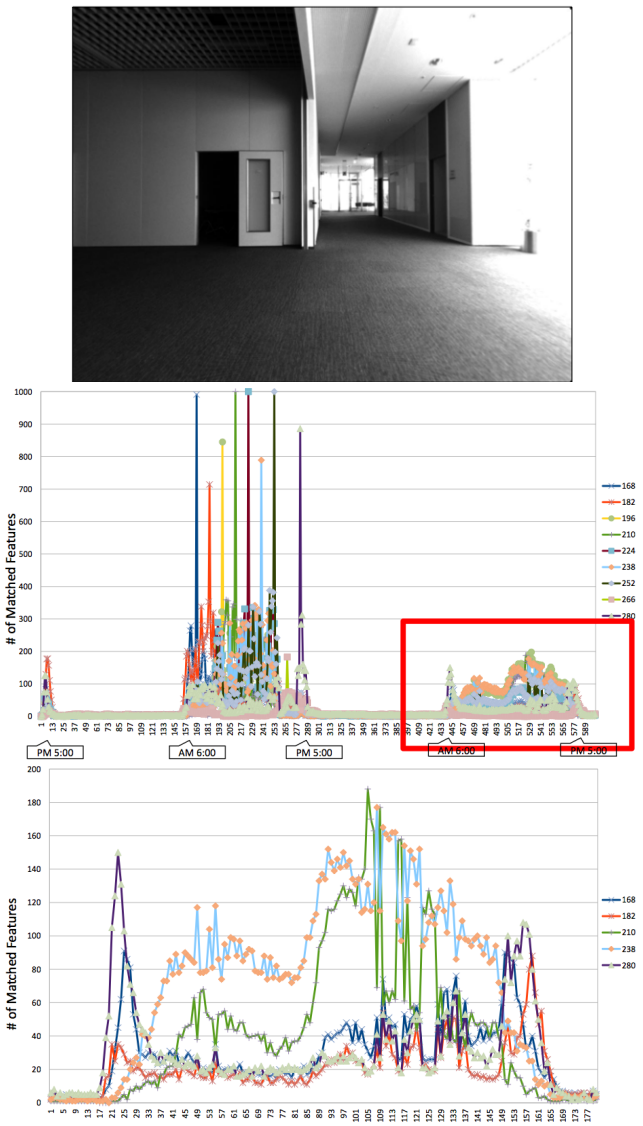

Fig. 4. Result of feature matching across two days with reference images only in the first day. The top image shows the scene of the experiment with a large bright window to the right. The middle image shows the complete result while the bottom image shows only the result from the second day (red highlighted rectangle in the top image).

shown in Figure 4. This figure shows the feature matching pattern over multiple days and the results are as expected with the number of feature matches peaking at the corresponding reference time even on the next day.

The sharp drop-off in feature matching emphasizes the need for a special strategy to address lighting changes in location recognition as otherwise the system would hardly be robust. Also whereas we have used a simple matching strategy of just using the descriptor distance, more robust matching methods, such as rejecting matches where the ratio of first-best to second-best match distance is below a certain threshold, are used in practice. While these methods can increase the number of correct matched features in practice, this does not affect the overall qualitative result of our experiments above.

III. Related work

Since location recognition itself has a vast literature, we focus our related work on methods that address illumination-invariant location recognition. The closest work to ours, which our work is also based on, is the paper by Mikulik et al. [6], which introduced the use of fine vocabularies and probabilistic modeling for illumination and viewpoint invariant image retrieval. We improve on their general method mainly by providing a mechanism for recovering feature correspondence in addition to performing image retrieval. In the specific domain of location recognition for VSLAM, we improve on that work by providing an automatic method for harvesting training data required for the probabilistic modeling. While a large part of the [6] is geared towards the use of their method for image retrieval in large internet-style image collections, we refocus the algorithm towards location recognition in VSLAM in particular and provide extensive experiments for this.

A number of works in the literature focus on appearance-only localization or image-level localization. Unlike in this paper, no metric localization is performed. However, they rarely address long and short term lighting variations in localization. A representative work that has been applied to large domains is the FABMAP algorithm [21] where a Chow-Liu tree [22] is used to a learn probabilistic relationship between features in an image that are useful for image matching. However, FABMAP is not very successful at dealing with lighting variations. An appearance only SLAM method using omnidirectional images is [23]. The use of omnidirectional camera removes the variation due to camera orientation but has no effect on the mismatches due to lighting variations. Another similar recent work is [24] which also attempts to overcome drastic lighting changes by matching sub-sequences of images rather than individual pairs. However, this work currently has certain restrictive assumptions, such as the speed of the car needing to be similar between the reference and query runs. Valgren et al. [5] do extensive tests across time of day, weather, and seasons to determine which features are effective for localization. They recommend a variant of SIFT called upright SIFT and require the use of reciprocal matching and enforcement of epipolar constraints which they admit makes the algorithm too slow.

Most current metric mapping algorithms that focus on longterm operation ignore the effect of lighting. Latif et al. [25] use trajectory consistency in addition to point-to-point location recognition. Their method is, in many ways, orthogonal to ours and the two can be used in conjunction. Similarly, [26] focuses on updating the map to accommodate changes in the environment over time and neglects location recognition under changing conditions.

Among methods that do address lighting is [27] which uses a grid-map representation with 3D point clouds and also performs salient edge detection to avoid lighting problems. Map matching is also performed in an illumination-invariant manner. Good results for localizing under various lighting and weather conditions were obtained but the system is too complex to operate in real-time. Cadena et al. [28] use a bag of words model but in addition perform geometric matching using a conditional random field (CRF) which provides greater lighting invariance than pairwise matching. This idea is similar to the hypergraph matching paradigm [29] which performs matching between subsets of features rather than pairwise matching. However, this method is too slow for practical use.

Hardware solutions to the illumination problem are also possible using high-dynamic range cameras. Irie et al. [30] describe such a solution where they address the problem of combining multiple exposures while the robot is moving.

IV. Fine vocabularies for learning lighting invariance

We follow Mikulik et al. [6] in probabilistically modeling variation in descriptor values using the fine vocabulary method (FVM). Descriptor distance is the basis of using the vocabulary tree for matching images. However, since computing the full descriptor distance for matching a large set of images would be too slow, the descriptors are discretized and considered the same if they fall into the same cluster. Ideally, we would like to make the clusters large enough to capture all the variation in the descriptors due to lighting, viewpoint etc. However, if the clusters are too large, distinct features risk being grouped together. On the other hand, if the clusters are too small, the same feature can get classified into various clusters. Hence, there is a tension in determining the size of the clusters.

The FVM is based on the thesis that descriptor distance is, in general, not a good indicator of image patch similarity. This thesis is borne out by the simple experiments presented by us above in Section II-B. The FVM method specifically tries to minimize the extra information required for any alternative approach.

The basic idea of the FVM is to oversegment the descriptor space very finely and learn the variation in descriptors as a probability distribution over the clusters, also called visual words. For each cluster (visual word), we learn the other possible visual words into which descriptors could possibly fall. Specifically, this is done by computing the probability distribution of observing a visual word W_j when visual word W_q is expected, $p(W_j|W_q)$ (notation following [6]). The observed words W_j are called alternate visual words.

The alternate word distribution $p(W_j|W_q)$ is learned in a data-driven manner from a training set of matching descriptors under different lighting conditions. Since we are interested in 3D reconstruction, the matching descriptors are taken to be the ones corresponding to the same 3D landmark. As in [6], we marginalize over the 3D landmarks to compute the alternate word probability distribution -

$$p(W_j|W_q) = \sum_i p(W_j|Z_i)p(Z_i|W_k) \tag{1}$$

Learning the fine vocabulary

Learning for the FVM consists of two steps - learning the fine clustering in the descriptor space, and learning the alternate word probability distribution. The fine clustering of the descriptors is done as in the usual vocabulary tree using

a representative set of descriptors. We use FastANN [31] for nearest neighbor calculations in the clustering and vector quantization phases.

A major difference in our learning phase of the FVM from [6] is in our method for obtaining this training set. In [6], the matched set of descriptors is obtained by applying wide-baseline matching to a huge collection of internet images, followed by transitively identifying features corresponding to the same 3D landmark. However, in our case, since we specifically want to improve localization in a VSLAM map, we adopt a different and somewhat simpler technique.

We collect video data over the same route at multiple times of the day. This procedure is repeated for multiple routes. Matching descriptors are collected from this set of video sequences in two ways. Firstly, we track features across consecutive frames within each sequence. Since each track corresponds to one 3D landmark, this satisfies our requirement of (1). However, descriptor tracks obtained through feature tracking only provide viewpoint change and rarely provide lighting change. Therefore, secondly, we use the groundtruth location of image frames to find feature matches between video sequences. If the location and pose of the camera is the same for images in different sequences, then descriptor matches between the two images are found simply through pixel location in the image (features with same pixel coordinates belong to the same 3D landmark). Harvesting transitive matching relations among the features (if feature A matches feature B, and B matches C, then A matches C) increases the number of matching descriptors in the training dataset.

During data collection, we ensure that a few locations are common between the sequences so that the frames at these locations provide the feature match links between sequences. In outdoor scenarios for vehicular applications, we record the precise camera pose using a high-grade GPS/IMU combination to obtain groundtruth on location. In this case, image frames in different sequences will have close, but not exactly the same pose. In this case, we select frames that are close to each other within some strict threshold distance. Subsequently, nearest neighbor matching in the image is used between the images to find corresponding features.

Using the above schemes, we can find a large set of training data with matched features without using descriptor distance for the matching.

Efficient computation of probability distribution

The number of 3D landmarks, Z_i in (1), in the training dataset of matched descriptors may number in the millions. Hence, efficient computation of the probability distribution is necessary. In [6], this was done by maintaining an inverted file data structure for computing the distribution $p(Z_i|W_k)$.

However, the speed of computing the probability distribution can be increased further by parallelizing the entire computation. To this end, we note that the distribution $p(W_j|Z_i)$ is computed per landmark and is much smaller in size than $p(Z_i|W_k)$. Hence, we split the training set into subsets of landmarks along with their matched descriptor tracks. The sum in (1) is computed independently and in parallel for each subset, with a normalized $p(W_j|Z_i)$ distribution for each landmark Z_i but an unnormalized $p(Z_i|W_k)$ distribution (since not all landmarks are in the subset). After the sum has been computed for each subset, the result from each computation is combined and normalized to obtain the final distribution on alternate words. Our implementation of the algorithm uses MPI to distribute the computation of the alternate word probability distribution across different CPU cores.

A. Image retrieval using the FVM

The implementation of the retrieval stage is done just as in the standard way using a vocabulary tree [11] where inverted files are used to select candidate images. The only major difference is that rather than just adding the visual word contributions directly to the query histogram, we add the probabilities of the alternate words given by the appropriate probability distribution. The contribution of each visual word is weighted by the idf weight [32]. The same procedure is followed during insertion of the reference images into the fine vocabulary tree. In addition, we truncate the probability distribution at some pre-specified threshold (usually 0.001) to increase efficiency by ignoring alternate words whose contribution is negligible.

B. Feature correspondence using fine vocabularies

So far we have only discussed image-level localization through image retrieval using the FVM method. The next requirement for pose estimation is to compute feature correspondence between the query image and the retrieved image. This is used for pose estimation as explained in Section II-A by means of the 3-point algorithm.

We can no longer use the descriptor distance to provide putative matches as this metric is not reliable under illumination change. Hence, we introduce metrics for finding putative matches that involve the alternative word probability distribution.

The metric, called the probability distance $M1$, simply computes the distance between the probability distributions of the visual word to which two descriptors belong. For instance, if descriptor D_1 belongs to visual word w_{D_1} and descriptor D_2 belongs to visual word w_{D_2}, then the distance between the two descriptors according to the metric $M1$ is

$$M1(D_1, D_2) \equiv \sum_i \|p\left(w_i|w_{D_1}\right) - p\left(w_i|w_{D_2}\right)\| \qquad (2)$$

Various norms and metrics that compute the distance between two discrete distributions such as the L1 and L2 norms, chi-square metric and symmetric KL-divergence can be used in place of $\|.\|$.

The above metrics are used to find putative feature correspondences for the RANSAC part of the pose estimation. Note that since this is the only use of these metrics, they do not need to have high precision in finding true correspondences.

V. Experiments

We use the VSLAM system described in [33] which employs a calibrated stereo camera and runs in real-time at up to 10hz on a laptop with a GPU. Our modified FVM is used in the location recognition of this VSLAM system. We use PointGrey Grasshopper cameras for the stereo setup and only monochrome images are used. The original VSLAM system uses corners detected using the Good Features To Track (GFTT) [34] algorithm and SURF descriptors. We evaluated various descriptors for their illumination invariance in an experiment similar to the one shown in Figure 2, where SIFT and SURF were found to provide the best performance. All subsequent results herein are shown using the SIFT descriptor.

For learning the fine vocabulary, we use the same vocabulary tree as used by [6], which is publicly available at http://ptak.felk.cvut.cz/mikulik-eccv2010/mikulik-eccv2010.html. The fine clustering in this dataset has been learned using more than one billion SIFT descriptors from over 6 million images. The final clustering consists of 4096 clusters in a two-level hierarchy resulting in a total of 16M clusters at the leaf level. We decimate the number of clusters to take every 8th cluster at both levels for a total of 256k clusters since our datasets are not so big. The authors used wide-baseline stereo matching on this large collection of images to harvest over 12 million tracks containing more than 5 features each (i.e. the same landmark was observed at least 5 times).

We use the Hessian-Affine detector and the SIFT descriptor implementations available at the url mentioned above to enable our use of the vocabulary provided by Mikulik et al. [6].

A. Datasets

We present results on two datasets collected outdoors and one collected in a simulated environment in an indoor environment. The first dataset, called the *National Avenue dataset*, consists of multiple sequences over the same route about a mile in length and is in the shape of a single large loop. The sequences were gathered at various times of the day, and on different days, by mounting a stereo camera pair on a vehicle. The streams are monochrome VGA images recorded at 30 hz. A total of 6 sequences were collected including one in winter at a different time of the year from the rest. Groundtruth for location recognition was obtained by manually labeling the frame numbers in all the other sequences corresponding to the locations of every 100th frame of a query sequence.

The second dataset, called the *Maude loop dataset*, again consists of 6 sequences over a distance close to 3 miles, with the sequences gathered at different times of day using the same stereo pair setup. However, this dataset includes a high-precision GPS/IMU data stream that annotates each frame with location. This is used as groundtruth for location recognition and pose estimation.

The third dataset, called the *Blender dataset*, consists of simulated sequences generated in a manner similar to [35]. Groundtruth is available from the rendering camera position. There are a total of ten sequences in this dataset with each sequence rendered with the sun at a different angle using constant indoor lighting.

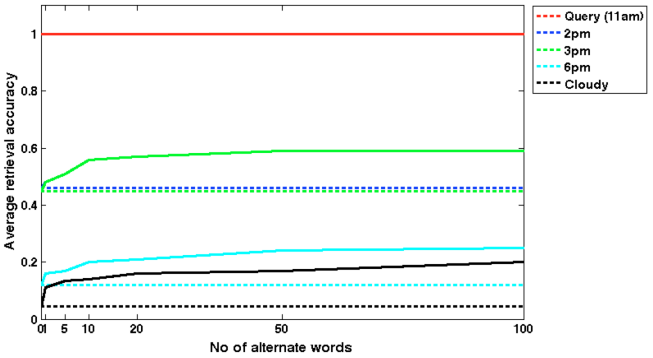

Fig. 5. Image-level localization result for national avenue dataset at various times of the day against the number of alternate visual words in the probability distribution. Query sequence was collected at 11am and time of each reference sequence is denoted by color. Dashed lines show the performance with the standard vocabulary tree and solid lines correspond to the FVM result. The red line shows that when the query sequence is localized against itself, the result is a 100% recognition using both methods. The FVM results for 2pm and 3pm coincide.

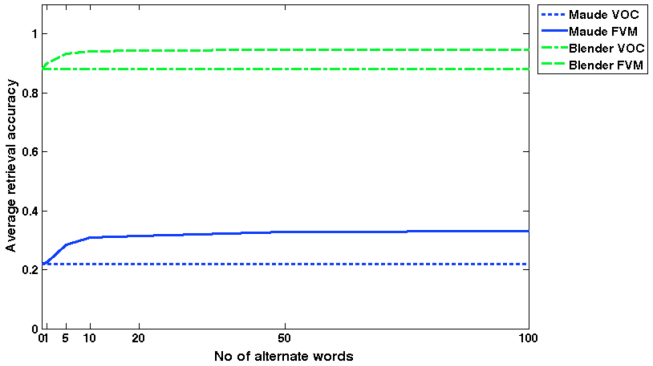

Fig. 6. Mean average precision of location recognition using the standard vocabulary tree (VOC) and the FVM for the Maude loop and Blender datasets.

B. Image retrieval for VSLAM

Our first experiment relates to image-level location recognition. To test the effect of using the FVM, we built the map at one time of the day and localize against it at another time. We compare the use of a standard vocabulary tree algorithm [11] against our FVM implementation.

The result of performing this comparison on the National Avenue dataset is shown in Figure 5. The figure shows the various curves which correspond to the map built using sequences gathered at different times. The query sequence remains the same and was collected at 11am. It can be seen that even a small alternate visual word distribution of length 5 to 10 words provides a significant increase of more than 10% in correct location recognitions. In this manually labeled dataset, the total number of frames with groundtruth that were tested was 123.

The result from the corresponding experiment on the Maude loop dataset is shown in Figure 6. This figure shows the mean average precision (accuracy) of location recognition over using every possible pair of sequences as query and reference sequence respectively, where the reference sequence is the one

Fig. 7. Feature correspondences computed using traditional descriptor distance for a pair of images at the same place captured at different times. Note the large number of wrong correspondences.

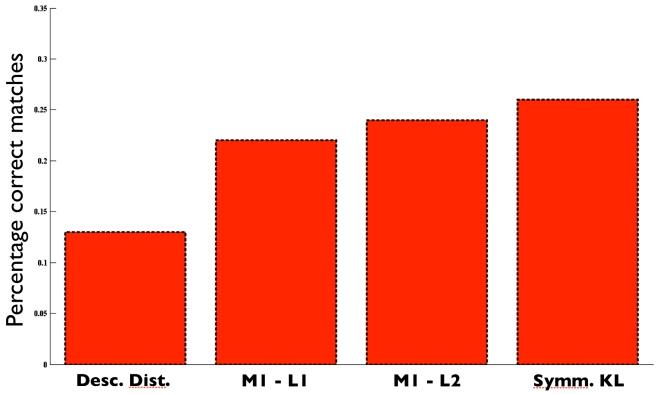

Fig. 8. Comparison of feature correspondence accuracy using various metrics. From left to right, descriptor distance, probability distance with L1-norm, probability distance with L2-norm and probability distance with symmetric KL-divergence metric.

used to build the map, and the query sequence is the one used to perform the localization. Thus, since there are 6 sequences in the Maude loop dataset, Figure 6 is the average result of 30 localization experiments. A significant increase of greater than 10% in the number of correct localizations over the standard vocabulary tree algorithm is again visible. For the purpose of this experiment, a correct location recognition was declared when the distance between the query frame and the retrieved frame as given by the GPS was less than 1m.

For the above two experiments, the result was poor when using the distribution learned from the dataset of [6]. Hence, we learned a new probability distribution from feature tracks harvested from the Maude loop dataset when testing on the National avenue dataset and vice versa. Feature track collection was done as described in Section IV.

The corresponding result from the Blender dataset is also shown in Figure 6. Here we only took one query sequence and averaged the location recognition results from considering every other sequence to be the reference sequence. One important difference to note here from the previous results is the much higher number of correct localizations overall. This may be either due to an easier indoor environment or non-realism in rendering lighting effects in the 3D graphics model or both. In any case, a significant improvement on using our FVM implementation is again evident.

C. Feature correspondence using the probability metric

We now compare the effect of the FVM on feature correspondence computation using our probability distance metric defined in Section IV-B. For this purpose, we selected 10 pairs of images from the Maude loop and National avenue datasets where the camera was approximately at the same location in each pair. However, the time at which each image in a pair was captured was significantly different. Figure 7 shows such a pair along with some correspondences obtained using the traditional descriptor distance.

We compared the number of correct feature matches obtained using the different metrics. The percentage correct matches for various metrics is shown in Figure 8. Note the low percentage overall due to large changes in lighting. However, our probability distance metric outperforms pure descriptor distance and the symmetric KL-divergence metric even doubles the number of correct feature matches compared to descriptor distance. We find that the L1-norm provides a good trade-off between speed and accuracy. Note that even though the percentage of correct feature correspondences may be low, the increase over the use of descriptor distance is significant since even a small improvement in the number of correct putative feature matches can improve runtime by reducing the number of RANSAC iterations required.

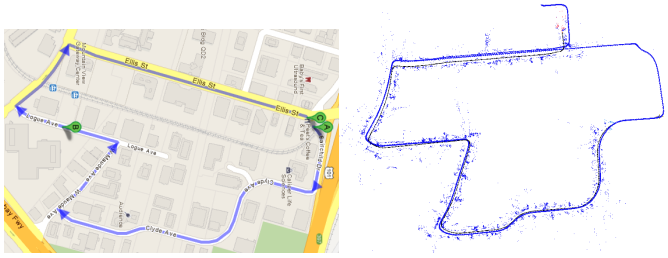

Fig. 9. (Left) Maude loop trajectory. (Right) GPS groundtruth (blue) and localization trajectory upon using FVM (black). Blue points represent the 3D landmarks. At this scale, the difference between the standard vocabulary tree and FVM localization trajectories is hard to see.

D. Pose estimation and localization error

While the above experiments give some strong indication of the improved performance due to our fine vocabulary method (FVM), the final practical test is to incorporate it into the localization system. We compared the localization performance on the Maude loop dataset. The GPS groundtruth and trajectory after localization with the FVM are shown in Figure 9. Compared to the groundtruth, the average localization error was 2.8m with the standard vocabulary tree and 2.2m with our FVM method.

VI. Conclusion and Discussion

We have presented a method to perform visual localization over long timeframes that takes into account strong changes in illumination. The main idea is to learn a probability distribution over the discretized descriptor space in a data-driven manner. Training data for learning the distribution is a set of descriptor tracks, each of which is known to correspond to the same feature. The descriptor tracks contain

representative instances of the lighting changes that we seek to overcome. Our method adds minimal computing overhead during runtime while providing significant improvements in localization performance as demonstrated by our experiments.

The main challenge in our method is in obtaining the training dataset to learn the probability distribution. We solved this problem by collecting descriptor tracks within a sequence through feature tracking, and between sequences using a high-precision GPS device in an outdoor environment. Our other main contribution in this paper is to compute feature correspondences using the FVM for which we introduced a family of probability distance metrics. We demonstrated on a practical traffic scene scenario that the use of our new method improves localization error and significantly reduces drift during localization.

Collection of indoor training data is a challenge which we worked around by the use of simulated data. We plan to test the result of using distributions learned from the simulated data on actual indoor environments to see if this improves performance. It is also future work to test to what extent a probability distribution learned in one environment can be used in a differing one. For instance, is it enough to learn a separate probability distribution for indoor and outdoor, or is it necessary to learn separate distributions for a city road, a parking lot, a highway and so on. Our method can be combined with orthogonal location recognition methods, particularly those which operate on sequences (e.g. [28]).

ACKNOWLEDGMENT

Special thanks to Andrej Mikulík for providing us with the dataset from [6] promptly and for very helpful comments.

REFERENCES

[1] M. Agrawal and K. Konolige, "Frameslam: From bundle adjustment to real-time visual mapping," *IEEE Transactions on Robotics*, vol. 24, no. 5, October 2008.

[2] A. Davison, I. Reid, N. Molton, and O. Stasse, "MonoSLAM: Real-time single camera SLAM," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 29, no. 6, pp. 1052–1067, Jun 2007.

[3] J. Lim, J.-M. Frahm, and M. Pollefeys, "Online environment mapping," *Computer Vision and Pattern Recognition*, pp. 3489–3496, 2011.

[4] K. Konolige and J. Bowman, "Towards lifelong visual maps," in *International Conference on Intelligent Robots and Systems*. St. Louis, USA: IEEE/RSJ, 2009.

[5] C. Valgren and A. Lilienthal, "Sift, surf and seasons: Longterm outdoor localization using local features," in *Proceedings of the European Conference on Mobile Robots*, 2007.

[6] A. Mikulík, M. Perdoch, O. Chum, and J. Matas, "Learning a fine vocabulary," in *European Conference on Computer Vision*, 2010.

[7] B. Clipp, C. Zach, J. Lim, J.-M. Frahm, and M. Pollefeys, "Adaptive, real-time visual simultaneous localization and mapping," in *Proceedings of the 2009 Workshop on Applications in Computer Vision*. IEEE, 2009.

[8] A. Irschara, C. Zach, J. Frahm, and H. Bischof, "From structure-from-motion point clouds to fast location recognition," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, 2009, pp. 2599–2606.

[9] D. Nistér, O. Naroditsky, and J. Bergen, "Visual odometry," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, Jun 2004, pp. 652–659.

[10] D. Scaramuzza and F. Fraundorfer, "Visual odometry [tutorial] part II," *Robotics & Automation Magazine, IEEE*, vol. 18, no. 4, pp. 80–92, 2011.

[11] D. Nistér and H. Stewénius, "Scalable recognition with a vocabulary tree," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2006.

[12] F. Schaffalitzky and A. Zisserman, "Multi-view matching for unordered image sets, or "how do i organize my holiday snaps?"," in *Proceedings of the 7th European Conference on Computer Vision, Copenhagen, Denmark*, vol. 1. Springer-Verlag, 2002, pp. 414–431.

[13] T. Tuytelaars and L. Van Gool, "Wide baseline stereo matching based on local, affinely invariant regions," in *british Machine vision conference*, 2000, pp. 412–425.

[14] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27. IEEE, 2005, pp. 1615–1630.

[15] C. Harris and M. Stephens, "A combined corner and edge detector," *Proceedings of the 4th Alvey Vision Conference*, pp. 147–151, August 1988.

[16] J. Shi and C. Tomasi, "Good features to track," in *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR'94., 1994 IEEE Computer Society Conference on*. IEEE, 1994, pp. 593–600.

[17] H. Aaneas, A. L. Dahl, and K. S. Pedersen, "On recall rate of interest point detectors," in *the Fifth International Symposium on 3D Data Processing, Visualization and Transmission*, Paris, France, May 2010.

[18] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.

[19] L. Quan and Z. Lan, "Linear n-point camera pose determination," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 21, no. 8, pp. 774–780, 1999.

[20] M. Fischler and R. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[21] M. Cummins and P. Newman, "Appearance-only slam at large scale with fab-map 2.0," *The International Journal of Robotics Research*, 2010.

[22] C. Chow and C. Liu, "Approximating discrete probability distributions with dependence trees," *Information Theory, IEEE Transactions on*, vol. 14, no. 3, pp. 462–467, 1968.

[23] E. Menegatti, M. Zoccarato, E. Pagello, and H. Ishiguro, "Image-based Monte-Carlo localisation with omnidirectional images," *Journal of Robotics and Autonomous Systems*, vol. 48, no. 1, pp. 17–30, August 2004.

[24] M. J. Milford and G. F. Wyeth, "Seqslam: Visual route-based navigation for sunny summer days and stormy winter nights," in *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2012.

[25] Y. Latif, C. Cadena, and J. Neira, "Robust loop closing over time," in *Robotics: Science and Systems (RSS)*, 2008.

[26] K. Konolige and J. Bowman, "Towards lifelong visual maps," in *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2009.

[27] K. Irie, T. Yoshida, and M. Tomono, "Mobile robot localization using stereo vision in outdoor environments under various illumination conditions," in *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2010, pp. 5175–5181.

[28] C. Cadena, D. Galvez-Lopez, J. D. Tardos, and J. Neira, "Robust place recognition with stereo sequences," *IEEE Transactions on Robotics*, vol. 28, no. 4, pp. 871–885, 2012.

[29] J. Lee, M. Cho, and K.-M. Lee, "Hyper-graph matching via reweighted random walks," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2011.

[30] K. Irie, T. Yoshida, and M. Tomono, "A high dynamic range vision approach to outdoor localization," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*. IEEE, 2011, pp. 5179–5184.

[31] M. Muja and D. Lowe, "Fast approximate nearest neighbors with automatic algorithm configuration," in *International Conference on Computer Vision Theory and Applications (VISSAPP'09)*, 2009, pp. 331–340.

[32] R. Baeza-Yates and B. Ribeiro-Neto, *Modern Information Retrieval*. ACM Press, 1999.

[33] J. Lim, J.-M. Frahm, and M. Pollefeys, "Online environment mapping," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2011.

[34] J. Shi and C. Tomasi, "Good features to track," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 1994, pp. 593–600.

[35] B. Heisele, G. Kim, and A. J. Meyer, "Object recognition with 3d models," in *Proc. BMVC*, 2009, pp. 29.1–29.11, doi:10.5244/C.23.29.