Large Scale Visual Localization in Urban Environments

Supreeth Achar, C.V. Jawahar and K Madhava Krishna

Abstract— **This paper introduces a vision based localization method for large scale urban environments. The method is based upon Bag-of-Words image retrieval techniques and handles problems that arise in urban environments due to repetitive scene structure and the presence of dynamic objects like vehicles. The localization system was experimentally verified it localization experiments along a 5km long path in an urban environment.**

I. Introduction

Localization is the process of determining the pose of the robot in an environment from sensory information and some sort of map or representation of the environment. Localization has applications in mobile robotics, autonomous vehicles and driver assistance systems. The use of visual data for robot localization has been studied extensively. One of the earlier successful implementations of a vision based localization is [1] where a robot equipped with a camera pointed directly upwards was able to use ceiling illumination values to determine its location using Monte Carlo filtering methods.

Many vision based localization systems make restrictive assumptions about the environment. [2] which uses images of building facades to localize in an urban environment. The underlying assumption is that all images used contain a dominant plane, this makes it difficult to automate the process of building a visual database of an environment.

Royer et al. [3] describe a monocular vision localization system which is presented as a preliminary vision based alternative to satellite based GPS localization. Keyframes are extracted from a video sequence of the robot's route / environment is used to build a three dimensional reconstruction of the environment. Localization is performed by extracting keypoints from the current camera view and matching them to each keyframe image. The frame with the largest number of matches is chosen and the exact position of the robot is determined from image point to 3D world point correspondences using Grunert's pose estimation algorithm. Although this method is effective and returns the metric pose of the robot it requires a computationally expensive reconstruction with complexity that is superlinear in the length of the video sequence. Also the current view needs to be matched against each of the keyframes which may not be feasible for use in real-time in large environments.

Qualitative localization methods have the advantage of not needing any metric modeling of the environment as they work directly upon the images themselves. Since qualitative localization involves finding previously seen images that are similar to the current view of the robot, methods for qualitative localization draw on those used for content based image retrieval. Features of some sort are extracted from the current view image and compared to those stored using a suitable similarity measure. The approaches can be divided into two broad classes depending on the type of features used, those that use global descriptors of the image as features and those that use local descriptors. Colour histograms [4] are simple global descriptors that have effectively been applied to robot localization, but they tend to give rather coarse results. The use of dimensionality reduction techniques to generate lower dimensional image representations that can be used for localization have also been studied. [5] uses kernel principal component analysis to generate global features. In [6], Fourier domain analysis of images captured by an omnidirectional camera was used to generate a Fourier signature that was used for localization.

Local features tend to be more robust to occlusions and changes in viewpoint. The successful use of local features in image retrieval applications motivated investigation of their applicability to qualitative robot localization. One approach would be to directly match local features (such as SIFT descriptors) between the current view and each of the stored images as done in [7]. This provides accurate results but because the descriptors extracted from the current view need to be matched against descriptors from all the stored images, the method does not scale well to large sets of images.

$$V_{C_{G_i} C_{W_i}} = \left[\begin{array}{cccccc} v_x^i & v_y^i & v_z^i & \omega_x^i & \omega_y^i & \omega_z^i \end{array} \right] \quad (1)$$

This paper presents a vision based localization method suitable for use in large scale urban environments especially for Indian roads. The method builds upon existing Bag-Of-Words techniques [8], [9] to address specific challenges that arise in urban environments such as occlusions by vehicles and similarity in visual appearance between different locations. The method can be used as a local localization scheme if a priori information of pose is available or can be used as a global localization method.

In outdoor environments, GPS is widely used for localization as it provides a straightforward way to obtain fairly accurate position and velocity estimates. In urban environments buildings can block the visibility of satellites and reflect signals which can cause GPS localization to fail. Crowded urban spaces with many buildings and indoor environments

Supreeth Achar is currently a graduate student at RI CMU; C.V. Jawahar is with the Center for Visual Information Technology, International Institute of Information Technology, Hyderabad, India; K Madhava Krishna is with the Robotics Research Center, International Institute of Information Technology, Hyderabad, India supreeth@cmu.edu jawahar@iiit.ac.in mkrishna@iiit.ac.in

where GPS is inaccurate or unusable tend to be rich in visual data. Thus vision based localization complements GPS well.

Vision based localization can be performed using image retrieval. An image taken from the robot's current pose is compared against a database of images taken from various poses in the environment to find the best matching image, as images coming from nearby poses in the environment tend to have similar content it can be inferred that the robot is near the pose at which the retrieved image was captured. One approach is to directly match local features (such as SIFT features [10]) between the current view and each of the database images as done in [7]. However, for large image databases this method is too slow and requires too much storage to be feasible.

Bag-Of-Words image retrieval techniques where images are modelled as collections of visual words can scale effectively to much larger image databases. In [11] visual words are matched and verified geometrically for localization in an indoor environment. In [12] BoW based image retrieval is used to perform localization from a side mounted camera in an urban environment. Instead of using a previously determined, fixed vocabulary, the visual vocabulary used is built by choosing words which are more informative for the localization process. This results in an increase in performance for a given vocabulary size but requires all the images to be available before the vocabulary is built. If images are added incrementally, the chosen vocabulary will be suboptimal. FABMAP [13] learns a generative model of the set of visual words describing appearance of a scene. This model can be used to compute the similarity between the current view and database images and also determine the probability of the images having come from the same location which allows it to handle environments where spatially separated locations are highly similar in appearance.

Lack of discriminating landmarks comes from the fact that urban areas contain many structures and objects that appear repeatedly at different positions in the environment. As a result, widely separated locations may have similar appearances (figure 1(b)). Features in an image belonging to these common structures will be less informative for the purpose of determining robot pose. The method presented addresses this issue of non discriminative features by assigning weights to features on the basis of mutual information between robot pose and feature observation. As mentioned before, while navigating through an Indian road many dynamic objects such as other vehicles and pedestrians will be visible as shown in figures 1(a) and 3. These dynamic objects can occlude the scene and corrupt localization results. Our method uses geometric inferencing to detect moving objects and filter out the features they generate. Similar geometric inferencing techniques were used for localization in indoor environments by current authors in [14]. The resulting filtered sets of visual words are used for localization.

Even though the current state of the art feature detectors [15] [10] [16] generate good matches between image pairs, a large fraction of the features they detect are unstable to even small changes in viewpoint and are non repeatable. These unstable features will act like noise in the bag of words describing an image. They are unlikely to generate valid matches and could produce invalid matches with unrelated images in the image database. Our method addresses this problem by using closely sampled images instead of well separated key images but only adding features that are stable to the database.

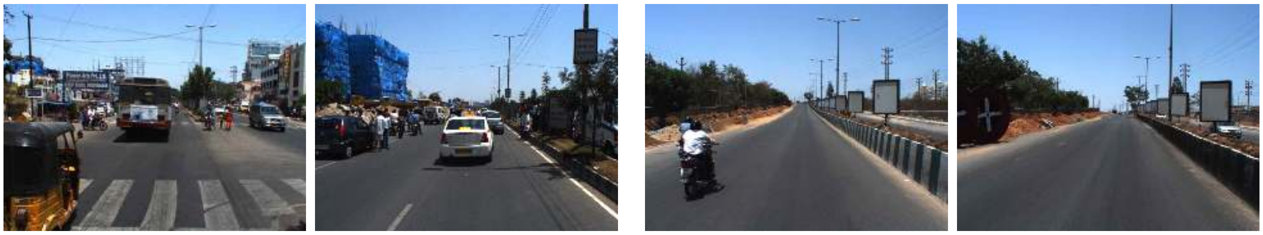
The method presented was tested by attaching a camera to the top of a car recording a video sequence while driving along a path roughly 5km in length down city roads. A localization database was built from the frames in this video and the algorithm was tested for both global localization and local localization by matching frames from two other runs along the same path to the database built from the training run. GPS readings were used to verify the correctness of the localization results.

The localization method proposed performs qualitative visual localization using image retrieval techniques. It builds upon standard bag-of-word based retrieval to handle specific issues that arise during large scale localization in outdoor urban environments. Bag of Words based methods model images as sets of local features that have been quantized into visual words. Two images are matched by comparing the sets of visual words they generate. For localization, features selected from the images should ideally have the following three qualities

1) Stability: The images features selected should be stable to small changes in camera viewpoint. Visual words that have poor repeatability are not useful for localization.
2) Staticness: Image features that come from dynamic objects in the environment such as traffic and pedestrians can obviously not be used to localize the robot.
3) Distinctiveness: In almost any environment, there will be objects (such as lane markings on the road) that appear frequently at many different places. Features from such objects can not give much information about the pose of the robot as they do not help discern between different poses.

Also, the localization process should use a priori information about the robot pose. Once the robot has been localized to a particular location, it can be safely assumed that the robot will be at a nearby position in the next frame. Hence some knowledge of which frames are likely to be returned by the image retrieval process is generally available beforehand which is not the case in most image retrieval applications.

An outline of the proposed localization method is illustrated in figure 2. SIFT features from a query image are extracted and quantized into visual words. Quantization is performed using the greedy N-Best paths search [12]. Features from dynamic objects are detected and filtered out by analyzing the motion of feature points between frames as described in section I-B. This filtered bag of visual words is passed on to the image matcher which finds candidate images most similar in appearance to the given query (sections I-C and I-D). Weights are assigned to each of the visual



(a) Heterogeneous traffic in Indian roads

(b) The images show views that are highly similar in appearance which are from widely separated locations

Fig. 1.

words used by the matcher. Weights are assigned (section I-C) so that words which are more distinctive around the current robot position hypotheses get higher weights. Feature matches from the best candidate images are then geometrically validated (section I-D) to obtain the localization results. These results are fed back into the system by updating the robot position distribution (section I-E). To ensure that unstable features are used for localization, only features that have matches in neighboring frames in the training image sequence are added to the localization image database.

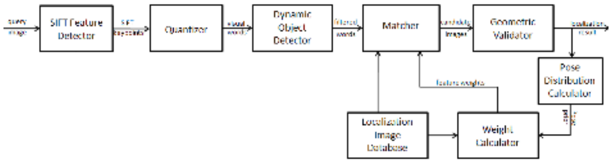


Fig. 2. A Block Diagram showing the design of the proposed localization system

A. Notation Used

The function $U(x;a,b)$ denotes a discrete uniform probability between a and b. $\mathcal{N}(x;\mu,\sigma^2)$ represents a Gaussian distribution with mean μ and standard deviation σ, $\hat{\mathcal{N}}(x;\mu,\sigma^2)$ is the discretized version of the same normal distribution. The robot pose X is treated as a discrete variable. A pose is labelled by the index of the nearest image in the image database.

B. Detecting Dynamic Scene Elements

In an urban scene the fastest changing elements will be those due to vehicular and pedestrian traffic. For a vision based localization system to work in this sort of dynamic environment it is important to be able to detect traffic for two reasons. Firstly, there will be occasions when the field of view of the camera is dominated by traffic and it will not be possible to localize. The system should be able to detect these failure cases. Secondly, using visual features that come from dynamic scene elements can corrupt localization results.

Learning the appearance of objects belonging to general categories like vehicles and pedestrians and using the resulting model for detection is a complicated problem and it would be difficult to design such a detector that could run in real-time. Instead, we use camera geometry and motion cues to detect traffic. Motion is detected by matching features between frame pairs. To ensure that there is a large enough base line and that vehicle motion is sufficient to create measurable disparities, the n^{th} frame in a video sequence is not matched to its predecessor but the $n-k^{th}$ frame instead. If a significant proportion of the features matched between the two frames do not change positions, then it is inferred that the robot is currently stationary. If not, the robot is in motion. When the robot is stationary, detecting dynamic scene elements is trivial. All features whose position in the image have changed between frame n and frame $n-k$ are from objects in motion.

When the robot is in motion, detecting other moving objects becomes more difficult. Three images are used frame n, frame $n-k$ and frame $n-2k$ referred to as I_0, I_1 and I_2 respectively. The feature correspondences between I_0 and I_2 are used to estimate the essential matrix E between the two views. The essential matrix is then decomposed to give the camera rotation R and translation T (up to scale) between the two views

Now that the relative pose of the second camera with respect to the first ($[R_2|T_2]$) is known, the 3D positions P_j of each feature with respect to the camera I_0 can be calculated by triangulating their coordinates in the image. A pose estimation algorithm is used to determine the position $[R_1|T_1]$ of camera I_1. The expected coordinates of each feature in I_1 can be determined from the camera projection equation $\hat{p_{1j}} = K[R_1|T_1]P_j$

The expected coordinates of each of the features in I_1, $\hat{p_{1j}}$ are compared to the coordinates actually observed p_{1j}. If they are significantly displaced from each other, it can be inferred that the j^{th} feature is from a moving object in the scene. Both the robot and the other vehicles will be typically moving in the same direction, roughly along the camera's principal axis. In this case estimating the 3D positions of feature points and checking whether they remain consistent over a frame triplet as described above is not always reliable because triangulation of features when the camera baseline is along the principal axis and the features are close to the camera center is a poorly conditioned problem.

Instead we exploit the fact that as the robot moves forward, the apparent size of objects should increase. Objects will only appear to shrink if they are also moving forward at a speed faster than the robot. Any object whose size decreases with time when the camera is moving forward can be assumed to be a moving object. Due to uneven road surfaces, camera vibration etc there will be some rotational component to the camera motion even while the robot is in forward motion. This rotation is corrected for using a homography H to warp the scales s_{2j} and image coordinates p_{2j} of features in I_2 before comparing them to I_0.

Any feature whose scale $\tilde{s_{2j}}$ in I_2 is larger than its scale s_{0j} in I_0 or which is further from its nearest neighboring feature in I_2 than in I_0 is marked as a feature from a moving object. Figure 4 shows some examples of the motion detection algorithm. Most of the false motion detections are due to feature mismatches.

The motion detection algorithm is used both while building the image database and while localization is being performed. When the image database is being built, features determined to be from moving objects are not added to the database. During localization, dynamic features are removed from the bag of words describing the current view.

C. Assigning Weights to Visual Words

Visual words that appear frequently at many different places provide less information regarding pose than those that appear rarely and whose occurrences are tightly clustered around a single location. The features that are most helpful in localizing are those that can be reliably detected around a single location. We assign weights to visual words such that words that tend to be more useful in determining pose are given higher weight and those that are unstable or occur at many locations have low weights. The weight assigned to the j^{th} visual word in the vocabulary tree is denoted as W_j.

The usefulness of a visual word can be interpreted in terms of how strongly its observation correlates with the robot being near a particular pose. This notion of usefulness can be quantified in terms of information gain. If the pose of the robot is modeled by probability mass function $P_X(x)$, the entropy of the distribution, $H(X)$ is a measure of the 'randomness' of the distribution of X. If an observation of the presence of visual word Z_j from the current position is considered, then the distribution of X conditioned over Z_j is $P_X(x|Z_j)$.

The conditional entropy $H(X|Z_j)$ will always be less than or equal to $H(X)$ with equality holding only if X is completely independent of Z_j. If there is any dependence between the random variables X and Z_j, then knowledge of the absence or presence of Z_j from the current view helps decrease the uncertainty in the robot's pose X. The mutual information $I(X;Z_j)$ is a measure of how much measuring Z_j reduces the uncertainty of X and is defined as $I(X;Z_j) = H(X) - H(X|Z_j)$

The probability of a feature Z_j being visible from a position x_0 is approximated as

$$P_{Z_j}(Z_j = 1|X = x_0) = \frac{1}{|N(x_0)|} \sum_{x \in N(x_0)} V(Z_j, x)$$

Where $V(Z_j, x)$ is an indicator function whose value is zero unless visual word Z_j was seen at pose x in the training video sequence in which case its value is unity. $N(x_0)$ is a set of poses such that elements $x \in N(x_0)$ form a local neighborhood around pose x_0. This averaging of Z_j over a neighborhood around x_0 is necessary because it is possible for the feature extractor to fail to detect a word in an image even if it is present. The conditional distribution $P_X(x|Z_j = 1)$ is given by

$$P_X(x|Z_j = 1) = \frac{1-\eta}{|F(Z_j)|} \sum_{(\hat{x} \in F(Z_j)} U(x; \hat{x} - w, \hat{x} + w) + \eta U(x; 1, |X|)$$

Where $F(Z_j)$ is the set of all poses $x \in X$ from which feature Z_j was visible. w defines the size of the neighborhood in terms of the width of a window around each feature occurrence in which it contributes to $P(x|Z_j)$. η is a mixing coefficient which is set a small value (between zero and one) to ensure that $P_X(x|Z_j = 1)$ takes a non zero value for all values of x. We can now calculate the probability of observing a feature Z_j from the marginal distribution as $P(Z_j = 1) = \sum_{x \in X} P_{Z_j}(Z_j = 1|X = x)P_X(x)$

The conditional probability $P_X(x|Z_j = 0)$ can be expressed as

$$P_X(x|Z_j = 0) = \frac{P_X(x) - P_X(x|Z_j = 1)P(Z_j = 1)}{1 - P(Z_j = 1)} \quad (2)$$

In terms of the probabilities calculated above, the mutual information $I(X;Z_j)$ between X and Z_j takes the form

$$W_j = I(X;Z_j) = \\ -\sum_{z \in 0,1} \sum_{x \in X} P(Z_j = z) P_X(x|Z_j = z) \log P_X(x|Z_j = z)$$

At each localization iteration, the prior distribution $P_X(x)$ of the robot pose changes as described in section I-E. The weights assigned to the features depend on this prior distribution and change with $P_X(x)$. The number of visual words in a vocabulary tree tends to be large (order of 10^5 to 10^6), calculating new weights for all the visual words at each localization iteration is not feasible. However, only the weights of visual words present in the current view are needed which means that only a few thousand weights need to be calculated at each iteration.

D. Localization by Image Matching

When a new query frame is captured for localizing, SIFT feature points are extracted and are then quantized using the vocabulary tree to get a set of visual words. Weights for each of the visual words present in the query frame are calculated (Section I-C) using the current pose pdf. The features extracted from the query are matched against the features in all the database images.

In generalized applications features in the query are considered to be possible matches to all features in the database images that have the same visual word label. Since the vehicle or robot to which the camera is attached moves along the road, there are constraints on camera motion that can be used to filter out many false feature matches. Camera roll will be minimal and so matching features will always be detected at similar orientations. Orientation is quantized into 32 bins and features are stored in the image database by word and orientation. Features are matched only against other features in the same or neighbouring orientation bins. Also, if pitching motion can be neglected, features that have positive height at some viewpoint will always have positive height and will thus always be detected in images above the principle point of the camera. Similarly, a feature detected below the principle point of the camera will always be detected below the principle point. This is used to further filter out false feature matches.

After feature matching, each frame in the database has been assigned a score equal to the sum of the weights of all visual words that appear in both the query and the database frame. Each database frame's score is normalized by the total number of features it contains. These scores are then low pass filtered and the highest scoring database images are selected for geometric validation.

To geometrically validate the matches from the high scoring database images, an essential matrix is fitted between the query features and the features from each of the selected images. Each matched feature pair that lies close to its epipolar lines is counted as an inlier. The database frame with the highest number of inlier matches is returned as the image in the database closest to the robot's current position.

E. Updating The Pose Probability Distribution

Once the robot has been localized, the pose pdf $P_X(x)$ needs to be updated. The possible poses of the robot are the high scoring frames from the localization step with the frame having the highest number of geometrically validated matches (the localization result) being the most likely position. If k position hypotheses $(x_1, x_2, ..., x_k)$ were geometrically validated during localization and the i^{th} hypothesis had c_i valid matches, then $P_X(x)$ is updated as

$$P_X(x) = (1-\eta)\sum_{i=1}^{k} \alpha_i \hat{\mathcal{N}}(x; c_i, \sigma) + \eta U(x; 1, |X|) \quad (3)$$

η is a mixing coefficient which is set to some small value between 0 and 1 that ensures that $P_X(x)$ is non zero for all values of x. This helps in recovery if a localization fails and the system loses track of the robots' position.

$P_X(x)$ takes the form of a number of peaks around the most likely poses with a small value at other poses. If localization is performed at a high rate $P_X(x)$ is can be used directly as the prior in the next localization iteration, otherwise it needs to be updated using a motion model.

II. Experimental Results

The proposed localization method was tested in an urban environment. A van fitted with a forward facing monocular camera (Flea2 colour Firewire camera with a $5mm$ lens) and a GPS receiver that was used to record vehicle position for each image captured that was used as ground truth data for quantitatively verifying the localization results. Frames for building the image database and performing localization iterations were captured at $7.5fps$. The internal parameters of the camera were determined beforehand.

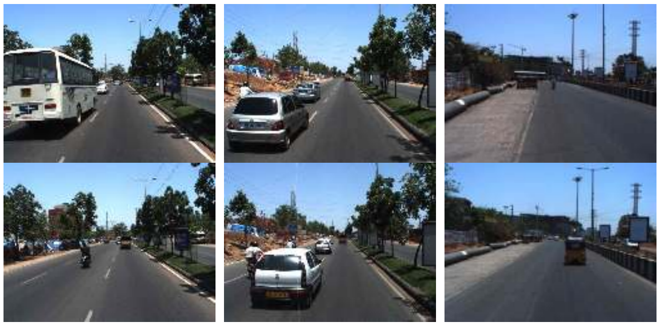


Fig. 3. Some challenging queries for which the proposed localization method succeeded. The images in the upper row are the queries, the images in the lower row are the results that were returned.

The vocabulary tree used was built over a set of 1500 images containing a total of roughly 1.7 million SIFT features. The vocabulary tree built had a height of 6 and a branching factor of 12 for a total vocabulary size of approximately 248k. While quantizing features, the best 6 paths were followed. The vocabulary tree was built from an image set captured outside the environment in which the localization was performed to demonstrate that localization performance is good even when the tree is not tailor made to describe the image features present in the vehicle's environment.

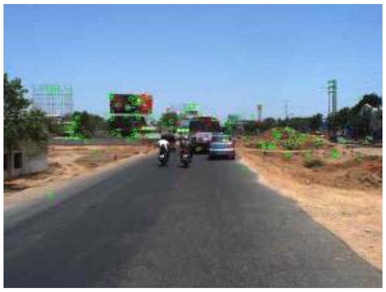


Fig. 4. Dynamic Object Detection: Matched features that were determined to be from moving objects are marked with red crosses, matched features that appeared to be stationary are marked with green circles

The path followed for testing the algorithm is shown in figure 5. The total length of the circuit was roughly 5 kilometers. The car was driven down the path 3 times. The first run was used for building the image database (containing 4953 images) and the video sequences captured over the next two runs (Test1 and Test2) were used for verifying the localization algorithm. Test1 covered a full circuit along the path shown in figure 5 while Test2 covered around three fifths of the entire path. Both the test sequences Test1 and Test2 started from different positions on the path.

Both the test sequences were used to test the localization

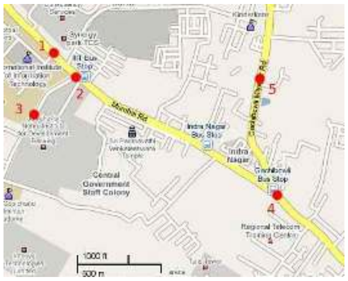


Fig. 5. The car was driven along a 1 → 2 → 3 → 2 → 4 → 5 → 4 → 2 → 1 path

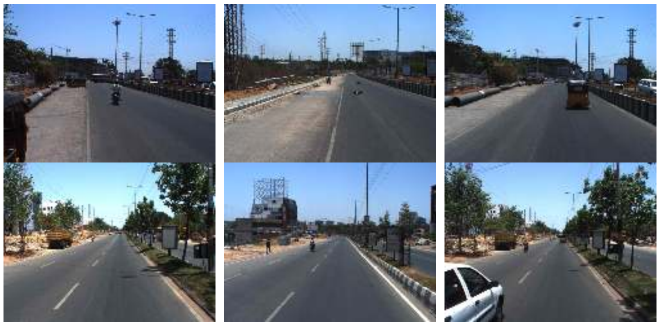


Fig. 6. Some frames for which the localization failed. The images in the left column are the queries, the images in the middle column are the incorrect results that were returned and the right column contains the correct matching images determined from GPS ground truth data.

algorithm as a local localization and global localization method. For local localization, the frames from the test videos were provided in sequence and the algorithm used the resulting continuity in vehicle position to maintain a pose distribution estimate as described in section I-E. For global localization, the same algorithm was used, but a robot pose distribution was not maintained. For each query frame, the prior distribution of the robot was assumed to be uniform.

Figure 3 shows some examples of frames from the test sequences that were successfully localized despite large parts of the image being occupied by vehicles. Figure 6 shows some of the frames for which localization failed and the image matching returned erroneous results.

The graphs in figures 7 show the indices of the result found in the database during localization for each frame from Test1 and Test2 respectively. In each figure the graph on the left shows the matching images found using a direct image retrieval system that does not detect dynamic objects or use a pose prior and uses uniform feature weights. The graphs in the center show the results of global globalization and the graphs on the right show local localization results.

For a query frame, the localization result returned was defined to be correct if it was captured from a location 7.5 meters or less from the point at which the query image was captured. Table I shows the localization error rate for direct image matching and the proposed localization algorithm (both local and global variants) on the Test1 and Test2 sequences. Table II shows the mean distances between test frames and localization results for the two sequences using both global and local localization.

TABLE I
LOCALIZATION ERROR RATE

	Direct Retrieval	Global Localization	Local Localization
Test1	3.0612%	1.7493%	0.9371%
Test2	2.18%	2.0%	0.7463%

TABLE II
MEAN LOCALIZATION ERRORS

	Direct Retrieval	Global Localization	Local Localization
Test1	8.7177m	7.6287m	3.1695m
Test2	2.4275m	2.5656m	2.2928m

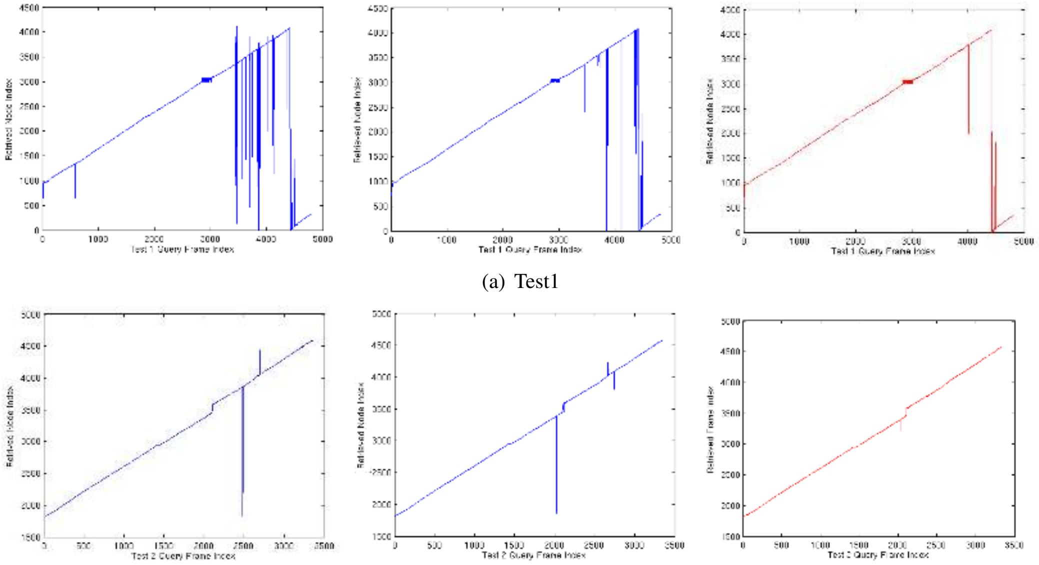
A. Analysis

In both Test1 and Test2, the proposed method gives better results that direct image retrieval with fewer jumps and mismatched frames for both local localization and global localization as shown in figure 7. During the training sequence and in Test1 the vehicle stopped briefly at a traffic signal. This is the cause of the jitter in the retrieved frame indices observed around query frame number 3000 visible in 7a. In Test2 the vehicle moved through the same traffic signal without pausing which resulted a small jump in the retrieved frame index around query frame number 2200 which can be seen in figure 7b. As expected, local localization which exploits the continuity in vehicle pose over time by using a prior pose distribution outperforms global localization. Using the proposed method localization reduced the number of erroneous results by around two thirds compared to direct image retrieval.

III. CONCLUSION

This paper presented a qualitative visual localization system that can be used in large scale outdoor environments. Challenges arising in these environments such as dynamic objects and perceptual aliasing due to visual self similarity are addressed. The probability density functions being used to update pose distribution probability and determine the probability of the robot being at a pose given a feature observation are heuristics. Finding theoretically well founded ways of finding these distributions is a direction for future work.

Dynamic objects are currently detected by analyzing the motion of detected local features over time. However there are cases where this method is badly conditioned and unreliable. It may be possible to address this problem by using motion cues to gradually learn appearance models of the dynamic objects typically present in an environment and then using a combination of both appearance and motion for detection of moving objects.



(a) Test1

(b) Test2

Fig. 7. The above graphs show the frame by frame localization results for the sequences Test1 and Test2. The first graph in each row shows the frame each image in the corresponding test sequence was localized to using direct image retrieval. The red graphs in the center show the results obtained by the proposed global localization method and the graphs on the right show the local localization results where the previous localization iteration is used to generate a robot pose prior

References

[1] D. Fox, W. Burgard, F. Dellaert, and S. Thrun, "Monte carlo localization: Efficient position estimation for mobile robots," in *In Proc. of the National Conference on Artificial Intelligence (AAAI*, pp. 343–349, 1999.

[2] D. Robertsone and R. Cipolla, "An image-based system for urban navigation," in *British Machine Vision Conference*, 2004.

[3] E. Royer, M. Lhuillier, M. Dhome, and T. Chateau, "Towards an alternative gps sensor in dense urban environment from visual memory," in *British Machine Vision Conference*, 2004.

[4] C. Zhou, Y. Wei, and T. Tan, "Mobile robot self-localization based on global visual appearance features," in *IEEE International Conference on Robotics and Automation*, 2003.

[5] T. Hashem and Z. Andreas, "Global visual localization of mobile robots using kernel principal component analysis," in *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems*, 2003.

[6] E. M. T. Maedab and H. Ishiguro, "Image-based memory for robot navigation using properties of omnidirectional images," *Robotics and Autonomous Systems*, vol. 47, no. 4, pp. 251–267, 2004.

[7] W. Zhang and J. Kosecka, "Image based localization in urban environments," in *International Symposium on 3D Data Processing, Visualization and Transmission, 3DPVT*, 2006.

[8] J. Sivic and A. Zisserman, "Video google: A text retrieval approach to object matching in videos," in *International Conference on Computer Vision*, 2003.

[9] D. Nistér and H. Stewénius, "Scalable recognition with a vocabulary tree," in *IEEE Conf. on Computer Vision and Pattern Recognition*, vol. 2, pp. 2161–2168, June 2006.

[10] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, pp. 91–110, November 2004.

[11] F. Fraundorfer, C. Engels, and D. Nister, "Topological mapping, localization and navigation using image collections," in *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems*, pp. 3872–3877, 2007.

[12] G. Schindler, M. Brown, and R. Szeliski, "City-scale location recognition," in *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, 2007.

[13] M. Cummins and P. Newman, "FAB-MAP: probabilistic localization and mapping in the space of appearance," *The International Journal of Robotics Research*, vol. 27, no. 6, pp. 647–665, 2008.

[14] D. Santosh, S. Achar, and C. V. Jawahar, "Autonomous image-based exploration for mobile robot navigation," in *IEEE International conference on Robotics and Automation*, 2008.

[15] J. Matas, O. Chum, M. Urban, and T. Pajdla, "Robust wide baseline stereo from maximally stable extremal regions," in *In British Machine Vision Conference*, vol. 1, pp. 384–393, 2002.

[16] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," in *9th European Conference on Computer Vision*, May 2006.