Ten-fold Improvement in Visual Odometry Using Landmark Matching

Zhiwei Zhu, Taragay Oskiper, Supun Samarasekera, Rakesh Kumar and Harpreet S. Sawhney
Sarnoff Corporation, 201 Washington Road, Princeton, NJ 08540, USA
{zzhu, toskiper, ssamarasekera, rkumar, hsawhney}@sarnoff.com

Abstract

Our goal is to create a visual odometry system for robots and wearable systems such that localization accuracies of centimeters can be obtained for hundreds of meters of distance traveled. Existing systems have achieved approximately a 1% *to* 5% *localization error rate whereas our proposed system achieves close to* 0.1% *error rate, a ten-fold reduction. Traditional visual odometry systems drift over time as the frame-to-frame errors accumulate. In this paper, we propose to improve visual odometry using visual landmarks in the scene. First, a dynamic local landmark tracking technique is proposed to track a set of local landmarks across image frames and select an optimal set of tracked local landmarks for pose computation. As a result, the error associated with each pose computation is minimized to reduce the drift significantly. Second, a global landmark based drift correction technique is proposed to recognize previously visited locations and use them to correct drift accumulated during motion. At each visited location along the route, a set of distinctive visual landmarks is automatically extracted and inserted into a landmark database dynamically. We integrate the landmark based approach into a navigation system with 2 stereo pairs and a low-cost Inertial Measurement Unit (IMU) for increased robustness. We demonstrate that a real-time visual odometry system using local and global landmarks can precisely locate a user within 1 meter over 1000 meters in unknown indoor/outdoor environments with challenging situations such as climbing stairs, opening doors, moving foreground objects etc..*

1. Introduction

Various computer vision techniques [10, 8, 1, 4, 7] have been proposed to develop a vision-based navigation system (or visual odometry) in the past few years. With the use of one, two or more cameras, most of the proposed visual odometry systems detect and track a set of stationary feature points from the scene and estimate the relative camera motion between two consecutive frames that are close in time. Subsequently, the relative motions between consecutive frames are chained together to obtain the absolute pose of the visual odometry system at each frame. However, such an incremental-motion-based visual odometry system is bound to accumulate errors and drift over time during navigation due to a variety of reasons including errors associated with calibration, image quantization, poor-quality images, inaccurate feature positions and outliers [10]. Because of this, visual odometry alone is not suitable for the long-distance navigation tasks where the drift grows superlinearly with the distance traveled [10]. For example, in [8], a real-time visual odometry using single stereo-pair is built and it produces between 1% to 5% drift error over runs that are several hundred meters long. In [5], another visual odometry system using stereo vision is built and it generates 4% error over runs of 100 meters. A simple and effective way to reduce the drift of a visual odometry system is to incorporate different types of sensors including GPS [5], IMU [12] and absolute orientation sensors [10]. However, even with the use of IMU or absolute orientation sensors, the drift will continue to grow, albeit at a slower rate.

In this paper, visual landmarks in the scene are utilized to reduce the drift of visual odometry. First, a dynamic local landmark tracking technique is proposed to track a set of local landmarks across a sequence of image frames and select an optimal set of tracked local landmarks as feature points for pose computation. As a result, the error associated with each pose computation is minimized so that the drift is reduced significantly. Second, we propose a framework that combines global landmark recognition with local visual odometry to correct the drift errors accumulated during navigation. Specifically, the proposed global landmark recognition technique is able to recognize whether a location has been visited in the past, allowing it to recompute the pose precisely with the use of recognized landmarks and correct the accumulated drift errors subsequently. Finally integrating the landmark based approach within an integrated multi-camera visual odometry plus IMU system enables us to achieve the precise results of close to 0.1% accuracy in unknown indoor/outdoor environments.

2. Related Works

In order to recognize the scenes during revisits, numerous techniques [15, 6, 13] have been proposed to perform visual place recognition previously. However, most of them [15, 6] focus on recognizing a small number of discrete known places with some complicated leaning techniques.

978-1-4244-1631-8/07/$25.00 ©2007 IEEE

On the other hand, the distinctive SIFT features [13] are detected and matched to a pre-built SIFT database map to locate itself for a robot globally. However, with the proposed database searching technique, the computational cost increases linearly with the database size [13], which may not be suitable for large-scale database. However, in recent years, using a text retrieval approach enables querying a large-scale database with an image very efficiently. In [14], a visual vocabulary is built with k-means clustering, and objects can be retrieved throughout a large movie database very fast with the trained visual vocabulary. In [9], using a hierarchical k-means clustering to build a visual vocabulary tree, the efficiency can be further improved.

Different from the above discussed techniques, we are not going to assume that either a pre-built landmark database is available or the landmark database is fixed during navigation. Instead, as the user travels, a landmark database will be built on-the-fly by inserting new landmarks dynamically; simultaneously, whenever the user comes back to a previously visited place, an efficient landmark searching algorithm is able to retrieve its most similar landmarks from the landmark database in real-time regardless of its size. This is a very challenging task since the landmark database will become extremely large eventually after days of travel.

Similarly, a visual vocabulary tree is also built to index the landmark database in our paper. Differently, a generic visual vocabulary tree is first built from a large amount of representative training data off-line. Then, its configuration will be updated or trained incrementally with newly captured landmarks added into the landmark database on-the-fly. However, as more and more landmarks are added into the landmark database to train the visual vocabulary tree during travel, both the quality and efficiency will decrease dramatically [9]. In order to overcome it, a geo-spatial constraint is applied to maintain the landmark database with a reasonable size for the visual vocabulary tree so that the optimal quality and efficiency can be always maintained.

3. Visual Odometry with Local Landmarks

Similar to the system proposed in [8], two calibrated cameras that form a stereo head are utilized in our visual odometry system, and the motion of the system is determined from the pairs of stereo images captured by the stereo cameras. While in the system proposed in [8], feature points are matched from one frame to the next (or over a fixed frames) for pose computation and then discarded, we proposed to dynamically track a set of feature points over frames and utilize them as local landmarks for pose computation. Harris corners are extracted from images and serve as feature points. As a result, by allowing matching and pose estimating over longer motion baselines, the error associated with each pose computation is minimized.

3.1. Dynamic Local Landmark Tracking

The goal of local landmark tracking is to establish correspondence of features amongst non-consecutive frames so that semi-local pose computation can be performed to control drift in position and orientation of sequential locations. Figure 1 illustrates the local landmark tracking scheme. It consists of the following steps:

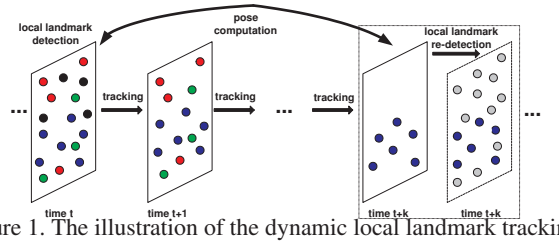


Figure 1. The illustration of the dynamic local landmark tracking.

Step 1: 3D local landmark initialization. Given an initial stereo pair, called reference frame, a set of 3D reference feature points are extracted. The reference feature points, called local landmarks, are tracked in subsequent stereo images.

Step 2: 2D local landmark tracking. Given a new stereo pair, the set of local landmarks are tracked. As the camera moves, some of the local landmarks will move out of the field-of-view of the cameras. Besides the local landmarks that move out of the field-of-view, some local landmarks also lose tracking due to their non-distinctive intensity distributions. Local landmarks with more distinctive intensity distributions survive longer during tracking. Hence, via tracking, the unstable local landmarks, which usually happen to be the false matches, are reduced significantly.

Step 3: Spatial distribution checking and pose computation. The aim of local landmark tracking is to obtain a set of reliably tracked local landmarks that contains few false matches. Although the number of falsely tracked local landmarks will decrease as the tracking continues, more and more stable local landmarks will also move out of the field-of-view of the camera as the system moves. Since the spatial distribution of the tracked local landmarks is essential for accurate pose estimation, a metric is needed to measure the spatial distribution of the tracked local landmarks. When the set of tracked local landmarks does not satisfy a spatial distribution criterion, a set of new local landmarks is initialized. A simple but effective metric is developed to measure the spatial distribution of the tracked local landmarks. Specifically, the left image of the 3D reference pair is divided into 10×10 grids. The total number of grids that contain local landmarks is used as a metric score. The spatial distribution metric is computed for each frame. If the spatial distribution metric is greater than a predefined threshold (50 empirically), the pose is computed from the 3D-2D correspondences of tracked local landmarks between the current stereo pair and the 3D reference frame.

Step 4: Updating 3D local landmarks. A new reference frame is established, if the spatial distribution metric for matches between subsequent frames is below a threshold. Therefore, the current stereo pair will be updated as the 3D reference frame, and a new set of reference feature points will be extracted as local landmarks. These newly extracted local landmarks are tracked in the subsequent stereo image frames, and the whole dynamical local landmark tracking technique repeats to recover the entire traveled 3D path.

With the use of dynamic local landmark tracking, a set of stable feature points that contains fewer false matches is obtained to produce a more accurate pose. On the other hand, during 3D reconstruction, the uncertainty of the reconstructed 3D coordinates of a feature point varies with the depth dramatically. since the established 2D-3D local landmark correspondences between two stereo pairs are usually more than 2 frames apart (could be several meters apart), depending on the moving speed and the motion type of the system, its 3D reconstructed error associated with each stereo pair can be considerably different. Therefore, a dynamic reference selection technique is proposed to automatically select the stereo pair with less 3D reconstruction error as the reference frame during pose computation. As a result, a much more accurate pose is computed.

During the dynamic local landmark tracking, a same set of local landmarks extracted from the 3D reference frame is tracked across its subsequent images so that the number of tracked local landmarks at each image can be used to characterize the degree of match between the image and 3D reference image. With the use of proposed spatial distribution metric above, it guarantees that there are enough local landmarks tracked across the images. Therefore, these images are roughly from the same scene and the 3D reference frame can be used as a representative image for the scene. Subsequently, the local landmarks extracted from the 3D reference frame is stored into a landmark database to represent the scene at this location accurately.

4. Visual Odometry with Global Landmarks

Another key element of our proposed framework is to recognize the revisits during navigation. The revisits need to be recognized even when the user returns to a previously seen location from a completely different direction. In order to address this, a multi-camera visual odometry system equipped with two pairs of stereo cameras is utilized. Specifically, as shown in Figure 2(a), one pair of stereo cameras faces forward while another pair of stereo cameras faces backwards so that the field of view of the system can be extended significantly. The cameras produce gray-scale 640×480 pixel images (Figure 2 (b)).

A big advantage of using the multi-camera approach is that it provides great robustness in situations where one camera is looking at a textureless area or only seeing moving objects. Multiple cameras looking backward and forward ameliorate this commonly occuring situation in real world scenes. The simple fusing technique proposed in [11] is adopted to fuse the pose outputs from both stereo pairs together as the final pose of the system. This robustness is further increased by integrating with an inexpensive IMU unit, which is a $2K Crista IMU that drifts over 720^o per hour. Such an inexpensive IMU alone with that high drift rate would be a non-starter, so a Kalman Filter (KF) is used to integrate the IMU measurements with the multi-camera visual odometry measurements [16]. The system automatically detects if either the IMU or multi-camera visual odometry readings has a large error and ignores them.

(a) (b)

Figure 2. The multi-camera system: (a) Frontal and back views of the system; (b) Captured images of both stereo pairs.

So far, with the use of local landmarks and an IMU in the proposed multi-camera visual odometry system, the drift can be reduced significantly compared to the original one-stereo-pair system proposed in [8]. However, drift from the true trajectory due to accumulation of errors over time is inevitable in any relative measurement system. This is where global landmark recognition and resetting of the camera location plays a significant role in reducing global drifts.

4.1. The Proposed Framework

Figure 3 shows a flowchart of the global landmark-based route correction algorithm for our visual odometry system with two stereo-camera pairs. The benefit of using such a multi-camera based visual odometry system is to extend the field of view of the system so that the chances of recognizing the global landmarks during route re-visits are increased.

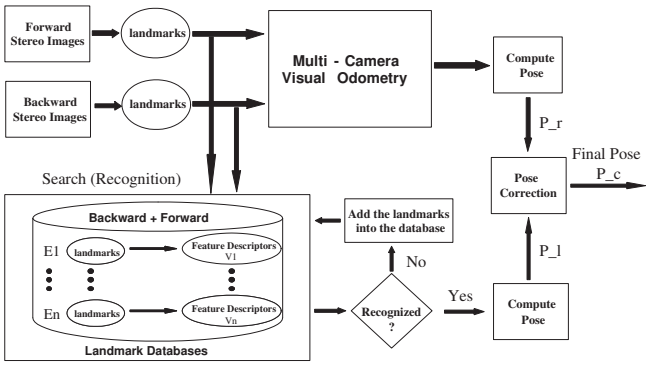


Figure 3. The global landmark-based route correction flowchart for a visual odometry system with forward/backward stereo pairs.

As shown in Figure 3, given a time stamp t, the first step

of our proposed algorithm is to detect and track a set of natural local landmarks from the images of the forward and backward stereo pairs individually. Subsequently, with the use of the extracted local landmarks, the pose of each stereo pair is estimated individually and they are further fused together via the adopted fusing technique to estimate the pose of the visual odometry system P_r.

Simultaneously, the extracted landmarks from both stereo pairs at the current time stamp t are used to search the landmark database for their most similar landmarks via the efficient database searching technique described in Section 4.3. Once a set of similar landmarks is returned, a new pose P_l is estimated by comparing the current image position of these landmarks with the 3D location stored in the database. The drift error accumulated in the pose P_r is corrected by P_l to obtain the final pose P_c. Note if matched landmarks are not found in the database then the new landmarks are inserted into the landmark database dynamically.

From Figure 3, we can see that there are two important components in the proposed framework: landmark recognition and pose correction. Both components are tightly coupled to benefit from each other.

4.2. Dynamic Landmark Database

During navigation, the local scene information at any particular point along the route is captured by "landmark snapshots". Each landmark snapshot is composed of the 2D coordinates, 3D coordinates and HOG descriptors of the landmarks as well as the estimated 3D camera location. The landmark database consists of a list of these snapshots extracted from the scene at the location associated with the 3D reference frame during local landmark tracking along the route. The combination of depth-dependent HOG descriptors described at Section 4.2.1 along with the spatial configuration of the landmarks makes them very distinctive and they serve as the fingerprint of the location. When the user arrives at a new location, the extracted landmarks from the scene are matched to the landmark database to decide whether the new location has been visited before. In addition, storing only the local landmarks extracted from each 3D reference frame during local landmark tracking produces a compact but rich landmark database.

4.2.1 Epipolar Constrained Landmark Matching using HOG

As we discussed in Section 3, given a pair of stereo images captured at each time t, we first detect a set of Harris corners as natural feature points from the left and right images respectively. Once the feature points are matched between the left and right images, the 3D coordinates of each feature point are computed by triangulation and they will serve as the local landmarks for the stereo pair. In order to characterize local appearance of each local landmark distinctly, the HOG descriptor [2] is computed from the left image of the stereo pair to represent each local landmark.

Instead of computing the HOG descriptor with a fixed scale to select the image patch size, the scale S of the HOG descriptor for each landmark is determined automatically via the following equation:

$$S = S_{ref} \times \frac{Z_{ref}}{Z} \tag{1}$$

where Z is the depth of the landmark in the local camera coordinate system at the current position, and S_{ref} is the scale used for the landmarks whose depth is equal to Z_{ref}. S_{ref} and the Z_{ref} can be set heuristically. Therefore, the closer the landmark to the camera, the larger the scale is.

Under the above proposed scheme, each image is represented by a set of extracted landmarks, and each landmark is associated with the 2D image coordinates, 3D coordinates and an HOG descriptor. Given two images taken at different locations, the task of landmark matching is to match the extracted landmarks between them using the HOG descriptors. Specifically, for each landmark in the first image, all the potential landmarks in the second image are searched for the correspondence that produces the highest similarity score. The search is based on the cosine similarity score of the HOG descriptor vectors between two landmarks. Since the matching technique described above does not consider any geometric or motion constraints, a large percentage of false matches are obtained. In order to improve the landmark matching, epipolar geometry constraints are utilized to eliminate the false matches. Specifically, from the obtained matches between two images, the fundamental matrix F is first estimated via the robust RANSAC technique [3]. Subsequently, based on the estimated fundamental matrix F, those matches that produce larger residuals than a predefined threshold value are treated as false matches and discarded.

Via the proposed technique above, the landmark matching between two images can be performed very effectively. However, its computational expense is quite expensive so that it is not suitable to search a large landmark database (more than 1000 images) using the proposed landmark technique directly. In order to overcome this issue, an efficient hierarchical landmark database search strategy is proposed. First, an on-the-fly database indexing technique with a generic vocabulary tree is introduced.

4.2.2 On-The-Fly Database Indexing with a Generic Vocabulary Tree

As the user travels, the size of the landmark database will increase rapidly as new landmarks are captured. Therefore, it is quite different from most of the previous applications where the database is usually fixed so that both the vocabulary-building and the database indexing can be performed on the same database off-line. In addition, building a large vocabulary tree is very time-consuming, which makes it impossible to perform both vocabulary-building

and database indexing simultaneously on the updated landmark database when new landmarks were added in real-time.

Therefore, a generic vocabulary is built off-line from a large set of training HOG descriptors, aiming to cover all the possible scenes. Similar to [9], a vocabulary tree is built by hierarchical k-means clustering. Once the generic vocabulary tree is built, given a database D, the standard weighting mechanism called "Term Frequency-Inverse Document Frequency (TF-IDF)" is performed to train the built generic vocabulary tree. Specifically, a weight w_i is assigned to each node i as follows:

$$w_i = \log \frac{N}{N_i} \tag{2}$$

where N is the number of images in the whole database, and N_i is the number of images in the database with at least one descriptor path through node i. Therefore, for a vocabulary tree with k nodes, each image I_d in the database can be represented as a vocabulary quantization vector $V_d = (t_1, t_2, ..., t_k)^T$ as follows:

$$t_i = \frac{n_{id}}{n_d} w_i \tag{3}$$

where n_{id} is the number of descriptors path through node i in the image I_d, and n_d is the total number of descriptors in the image I_d.

Once the vocabulary tree is trained via a given database D, given a query image I_q, the images in the database can be ranked by the similarity scores between the computed query vector V_q and all the vectors V_d in the database. During the implementation, inverted files [14] are utilized to facilitate efficient database indexing. Specifically, each node i is associated with an inverted file F_i, and each inverted file F_i stores a list of database images (including image ID, the *term frequency* $\frac{n_{id}}{n_d}$) that path through node i with at least one descriptor, and the length of the list N_i. Given a query image I_q, only the inverted files of the nodes that it paths through need to be utilized for similarity scoring (the number of inverted files is at most $L \times n_d$ with a tree that has L levels). Therefore, the use of inverted files makes the retrieval extremely fast.

When a new image of landmarks is added into the database, the vocabulary tree is updated incrementally to accommodate the new image in the following two steps. First, the database size N will be updated as $N = N + 1$. Second, for each landmark descriptor in the new image, the inverted file of each node that it visited at each level will be updated by inserting the new database image ID, updating the *term frequency* $\frac{n_{id}}{n_d}$ and the new list length N_i. Similarly, when an existing image of landmarks is removed from the database, the database size and the inverted file of each visited node can be updated accordingly.

4.3. Efficient Database Searching

Via the above proposed on-the-fly database indexing technique, the database images can be ranked efficiently according to the similarity scores computed with a query image. In order to improve the retrieval quality, the top-ranked m candidates in the above step will be further matched with the query image using the proposed epiplar-constrained landmark matching via HOG. Usually, the most similar images will be in the top-ranked images so that the number of candidates m can be set to small. With the above two-step scheme, the landmark database searching can be done extremely efficient.

However, the size of the landmark database increases rapidly as the travel continues so that the database can contain several billion images easily after several days' travel. Therefore, it is really difficult to guarantee that the selected top-ranked m candidates will always contain the right image for such a large-scale database. As a result, we propose an efficient hierarchical database search strategy composed of the following three steps:

Step 1: Fast Database Pruning via the geo-spatial constraints. When the user travels to a new location, based on its estimated 3D location as well as the estimated drift rate (or uncertainty), a geo-spatial search region is obtained automatically, which is shown as a yellow circle in Figure 4. Subsequently, with the use of the geo-spatial search region, a set of candidates of the landmark snapshots can be obtained quickly from the landmark database and put into a landmark candidate database.

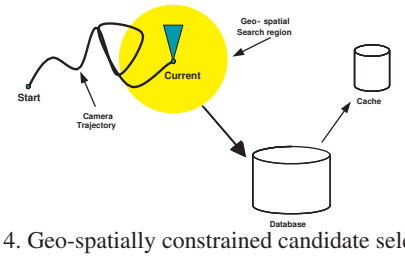


Figure 4. Geo-spatially constrained candidate selection.

Since the number of selected candidates in the candidate database is only determined by the size of the geo-spatial search region, which is typically much smaller than the whole database, we can reduce a large-scale database (millions or more) to a smaller candidate cache (thousands) during landmark database search.

Step 2: Fast Candidate Pruning via a vocabulary tree. During travel, the system usually starts with an empty landmark candidate database. As travel continues, new images will be added into the landmark candidate database and odd images will be removed from it periodically via the geo-spatial constraint in Step 1. Simultaneously, the generic vocabulary tree is updated incrementally with both the old and new images as described in Section 4.2.2. Since the number of the old and new images is small, the updating of the

vocabulary tree is done quickly.

Once the vocabulary tree is updated, it is used subsequently to rank all the images in the landmark candidate database for a given query image. Finally, a smaller candidate cache that contains only the top-ranked m (usually less than one hundred) is obtained.

Step 3: Epipolar-Constrained Landmark Matching via HOG. Once a smaller candidate cache is obtained, the proposed HOG-based landmark matching algorithm is activated to search the candidate cache. The number of matched landmarks is used to characterize the degree of matching for each snapshot. Once a snapshot that satisfies a predefined similarity measurement threshold is returned, the search stops.

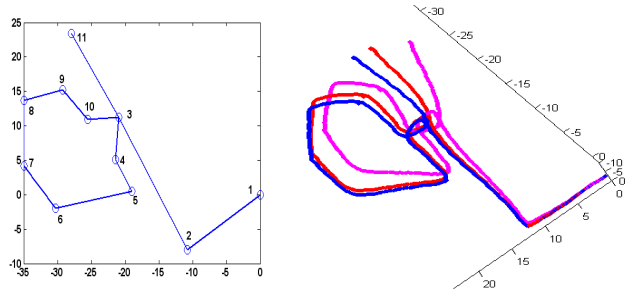
4.4. Global Landmark-based Pose Correction

With the use of recognized landmarks serving as the reference points, a new pose P_r of the visual odometry system at the current position can be recomputed directly with the standard technique using preemptive RANSAC followed by iterative refinement. At the same time, it will propagate back to a list of landmark snapshots at the previous positions in the database and correct their poses one by one.

5. Experiment Results

5.1. Performance of using Local Landmarks

In order to evaluate the improvements using local landmarks, we applied it to a set of collected video sequences with ground-truth (only the frontal stereo pair). Each video sequence was recorded at 30fps while a user wearing our system was traveling along a set of predefined routes. A set of key-points was established along the path, and the user had to pass through them exactly. The key-points' locations along each path were measured by a high-precision Differential GPS (DGPS) with centimeter accuracy and they will serve as the ground-truth to evaluate the performance of our improved visual odometry system. Figure 5 (a) shows a measured 3D trajectory of the user by DGPS, which is around 106 meters long.



(a) (b)

Figure 5. The comparison between (a) the 3D trajectory measured by DGPS, and (b) the 3D trajectories estimated by our improved visual odometry using local landmarks

However, the coordinate systems of the DGPS and the visual odometry are different so that the accuracy of the visual odometry cannot be measured directly. Therefore, a relative localization accuracy metric is computed as follows. First, the coordinates of the set of key-points along the 3D trajectories measured by both DGPS and our visual odometry system are collected. Second, select a key-point as a reference point, and compute the distance between the reference point and each of the remaining key-points in both coordinate systems individually. Once all the distances are computed, they are summed in both coordinate systems individually and the summed distances are subtracted and divided by the number of remaining key-points to obtain an average error. Finally, the absolute value of the obtained average error is assigned to the reference point. Finally, the average of the relative localization accuracies over all the key-points is utilized to characterize the overall performance of our improved visual odometry system.

Figure 5 (b) shows an example of the measured 3D trajectories of a user by our system under different improvements proposed in Section 3.1. Table 1 also summarizes the relative localization accuracies of the system under these improvements. Clearly, after integrating the proposed improvements into the visual odometry system, the average relative localization error decreases from 4 meters (original technique proposed in [8]) to less than 1 meter eventually for single stereo-based visual odometry.

Table 1. The relative localization accuracies of the improved visual odometry system under different improvements (meters)

	Original (magenta)	Dynamic local landmark tracking (red)	Dynamic reference selection (blue)
Min.	0.1480	0.6679	0.0648
Max.	10.4110	3.8894	2.0981
Med.	2.4633	0.8933	0.7907
Avg.	4.1788	1.5710	0.8311

5.2. Performance of Landmark Database Indexing

As introduced in Section 4.2.2, the key to make the online landmark matching with a large-scale landmark database possible is the use of geo-spatial constrained online updating of a generic vocabulary tree. In order to build such a generic vocabulary tree, a large amount of videos were recorded while the user was travelling through various indoor and outdoor environments, such as offices, malls, forests, downtowns, etc.. Finally, around $45k$ images are selected from these videos, and all the HOGs of the Harris corners in all the images are extracted to build a vocabulary tree with 10 branch factors and 6 levels. Specifically, similar procedures discussed in [9] using the hierarchical k-means clustering is applied, and the number of times that the EM algorithm will run is set to 25 for each level.

Once the generic vocabulary tree is built, given a landmark candidate database at any time t during travel, depend-

ing on the geo-spatial search region, usually there are several new images that need to be inserted into the generic vocabulary tree to update it. It is done very efficiently and only the inverted files of the visited nodes of the trees need to be updated. When scoring each image in the landmark candidate database, we found that the inner nodes of the vocabulary tree is not useful so that only the inverted files associated with the leaf nodes are involved in the similarity scoring, which further saves some memory usages and computations by cutting all the inverted files of the inner nodes.

In order to demonstrate the effectiveness of the proposed geo-spatial constrained database indexing technique, a video sequence containing around $27k$ frames was recorded first when the user wearing our system travelled along a predefined route that has 10 key-points marked on the ground are shown in Figure 6. All the images are inserted into generic vocabulary tree and the images captured when the user stepped onto these 10 key-points serve as ground truth. During test, at each key-point, 10 images was collected while standing randomly within its 2-meter radius. These test images exhibits quite large camera view-point changes from their database images. Without the geo-spatial constraint, the average rank of these 100 retrievals is 94.2, while the average rank of these 100 retrievals is only 3.8 with geo-spatial constraint. It clearly shows that using the geo-spatial constraint will improve the rank significantly.

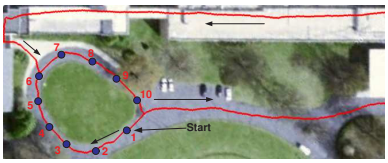


Figure 6. The part of the travelled route.

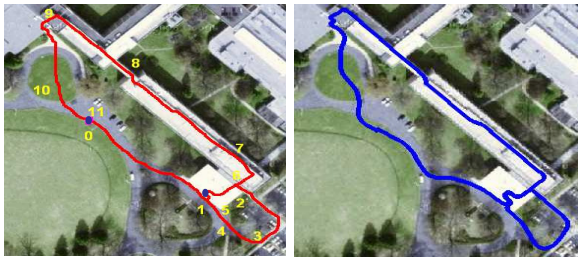
5.3. Performance of the Multi-camera Based Visual Odometry with Global Landmark Correction

In order to evaluate the performance of the proposed multi-camera based visual odometry system integrated with local and global landmark matching, a set of experiments were conducted under the indoor and outdoor environments.

5.3.1 Indoor/Outdoor Sequence with Loop Closure

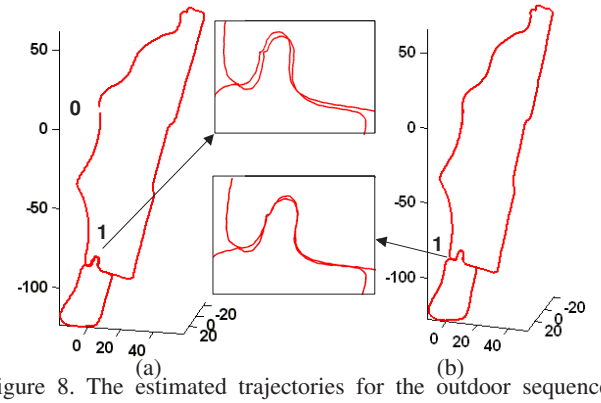
In this experiment, a video sequence was recorded at 15fps while the user wearing our system travelled along a predefined route that goes through outdoors and indoors as shown in Figure 7 (a) and two key-locations are marked as blue along the route. The first blue key-location associated with "0" is the starting point and the second one associated with "1" is a crossing point along the route. Both key-locations were marked on the ground and the user had to pass through them during travel. The recorded video sequence is around 7 minutes and 15 seconds long, and the total travelled distance is around 545.51 meters.

Figure 8(a) shows the estimated trajectory of the multi-camera visual odometry system without global landmark recognition. From the enlarged region around the location "1", it shows clearly that the error accumulates and the drift grows gradually during navigation. Table 2 reports the measured distance deviations for two fixed key-locations along the route during the revisits. In terms of the loop closure deviation, the error using multi-camera visual odometry and IMU is around 2.08 meters for this 546-meter long trip.



(a) (b)

Figure 7. (a) The planned route on the map (manually drew, directed by numbers); (b) The estimated route plotted on the map.



(a) (b)

Figure 8. The estimated trajectories for the outdoor sequence: (a) Multi-camera based visual odometry alone. (b) Multi-camera based visual odometry with global landmark correction.

However, if we integrate the global landmark recognition into the multi-camera visual odometry system, all the revisits along the route can be successfully recognized to correct the drift. Figure 8 (b) shows the estimated trajectory with global landmark recognition. From the enlarged region around the location "1", it shows clearly that the error is corrected. From Table 2, we can see that the measured distance deviations for both fixed key-locations are less than 0.5 meters during the revisits. In terms of the loop closure deviation, the error is around 0.24 meters for this 546-meter long trip. Figure 7 (b) shows the estimated trajectory plotted on the map that the user traveled and it corresponds quite accurately with the planned route shown in 7 (a).

Table 2. The deviations at the key-positions during revisits (meters)

	Point 1	Point 0
No Landmark Correction	1.05m	2.08m
Landmark Correction	0.16m	0.24m

5.3.2 Indoor Three-Floor Building Sequence

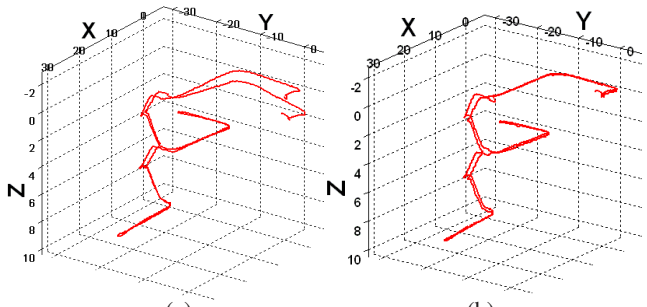
In this experiment, a video sequence was recorded at 15fps while the user traveled inside a three-floor build as shown

in Figure 9 (a). Specifically, starting from a fixed key location in a room on the third floor, the user left the room and traveled through the corridor and then took the stairs to the second floor. On the second floor, the user traveled through the corridor and came back to the same stairs and then went down to the first floor. After traveling through the corridor on the first floor, the user went back to the same stairs and took the stairs directly to the third floor and back to the room of origin and the start location. The whole video sequence is 4 minutes and 20 seconds long, and the user traveled around 242 meters. Some randomly selected images are shown in Figure 9 (b). As shown in the video, during the travel, the user opened the doors along the route thus exposing new areas and also subjecting the system to a wide variety of illumination and geometry scenarios.

(a) (b)

Figure 9. (a) The three-floor building; (b) The randomly selected left images of the frontal pair.

Figure 10(a) shows the estimated 3D trajectory of the user for this indoor stair sequence without landmark recognition. The error accumulates gradually as the user travels so that the estimated trajectory is not able to end at the location where it starts. In terms of loop closure accuracy, the final measured distance deviation is around 1.8 meters.



(a) (b)

Figure 10. The estimated trajectories for the indoor stair sequence: (a) Multi-camera based visual odometry alone; (b) Multi-camera based visual odometry with global landmark correction.

However, as shown in Figure 10(b), the 3D trajectory of the user can be estimated precisely once we integrate global landmark recognition. The final measured distance deviation is only 0.13 meters, which shows that the drift is corrected successfully and the estimated trajectory ends at the location where it started precisely. The three-floor structure and the stairs are clearly revealed.

6. Conclusion

In this paper, visual landmarks are utilized to improve the visual odometry. Equipping the visual odometry with the capability to recognize a set of stationary visual landmarks from the scene locally and globally during navigation, we can achieve close to 0.1% localization accuracy. Using the proposed techniques, we have developed a real-time wearable multi-camera visual odometry system with one pair facing forward and the other pair facing backward, and it runs at 15fps comfortably in a machine with a Duo-Core 3.4Ghz processor and 2G memory.

References

[1] P. Corke, D. Strelow, and S. Singh. Omnidirectional visual odometry for a planetary rover. In *IEEE Conference on IRS'04*, 2004.

[2] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *IEEE Conference on CVPR'05*, 2005.

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

[4] A. Johnson, J. Montgomery, and L. Matthies. Vision guided landing of an autonomous helicopter in hazardous terrain. In *IEEE Conference on ICRA'05*, 2005.

[5] K. Konolige, M. Agrawal, R. C. Bolles, C. Cowan, M. Fischler, and B. Gerkey. Outdoor mapping and navigation using stereo vision. In *International Symposium on Experimental Robotics*, 2006.

[6] J. Kosecka, L. Zhou, P. Barber, and Z. Duric. Qualitative image based localization in indoors environments. In *IEEE Conference on CVPR'03*, 2003.

[7] A. Milella and R. Siegwart. Stereo-based ego-motion estimation using pixel tracking and iterative closest point. In *IEEE Conference on CVS'06*, 2006.

[8] D. Nister, O. Naroditsky, and J. Bergen. Visual odometry. In *IEEE Conference on CVPR'04*, 2004.

[9] D. Nister and H. Stewenius. Scalable recognition with a vocabulary tree. In *IEEE Conference on CVPR'06*, 2006.

[10] C. Olson, L. Matthies, M. Schoppers, and M. Maimone. Rover navigation using stereo ego-motion. *Robotics and Autonomous Systems*, 43(4):215–229, 2003.

[11] T. Oskiper, Z. Zhu, S. Samarasekera, and R. Kumar. Visual odometry system using multiple stereo cameras and inertial measurement unit. In *IEEE Conference on CVPR'07*, 2007.

[12] S. Roumeliotis, A. Johnson, and J. Montgomery. Augmenting inertial navigation with image-based motion estimation. In *IEEE Conference on ICRA'02*, 2002.

[13] S. Se, D. Lowe, and J. Little. Global localization using distinctive visual features. In *IEEE Conference on IRS*, 2002.

[14] J. Sivic and A. Zisserman. Video google: a text retrieval approach to object matching in videos. In *IEEE Conference on ICCV'03*, 2003.

[15] A. Torralba, K. Murphy, W. Freeman, and M. Rubin. Context-based vision system for place and object recognition. In *IEEE Conference on ICCV'03*, 2003.

[16] Z. Zhu, T. Oskiper, O. Naroditsky, S. Samarasekera, H. S. Sawhney, and R. Kumar. An improved stereo-based visual odometry system. In *Proceedings of the Performance Metrics for Intelligent Systems (PerMIS'06)*, 2006.