# High-Precision Localization Using Visual Landmarks Fused with Range Data

Zhiwei Zhu, Han-Pang Chiu, Taragay Oskiper, Saad Ali, Raia Hadsell, Supun Samarasekera and Rakesh Kumar
SRI International Sarnoff, 201 Washington Road, Princeton, NJ, 08540
{zhiwei.zhu,han-pang.chiu,taragay.oskiper,saad.ali,raia.hadsell,supun.samarasekera,rakesh.kumar}@sri.com

## Abstract

*Visual landmark matching with a pre-built landmark database is a popular technique for localization. Traditionally, landmark database was built with visual odometry system, and the 3D information of each visual landmark is reconstructed from video. Due to the drift of the visual odometry system, a global consistent landmark database is difficult to build, and the inaccuracy of each 3D landmark limits the performance of landmark matching. In this paper, we demonstrated that with the use of precise 3D Lidar range data, we are able to build a global consistent database of high precision 3D visual landmarks, which improves the landmark matching accuracy dramatically. In order to further improve the accuracy and robustness, landmark matching is fused with a multi-stereo based visual odometry system to estimate the camera pose in two aspects. First, a local visual odometry trajectory based consistency check is performed to reject some bad landmark matchings or those with large errors, and then a kalman filtering is used to further smooth out some landmark matching errors. Finally, a disk-cache-mechanism is proposed to obtain the real-time performance when the size of the landmark grows for a large-scale area. A week-long real time live marine training experiments have demonstrated the high-precision and robustness of our proposed system.*

## 1. Introduction

High-precision localization is needed for a variety of applications, such as augmented reality, military training, robot navigation, etc. However, due to the drift issue of the vision-based navigation system, it is very hard to maintain tracking with high accuracy for hours, such as 5cm error for locating a trainer within a median-sized training facility during a course of hour-long training session. In order to reduce the drift, different techniques have been proposed by using either non-vision sensors such as IMU unit [8, 9], or loop closure [6, 11] and landmark matching with a pre-built visual landmark database [5, 7, 4]. Among them, using the landmark matching for correcting the long-term drift in the absence of GPS demonstrates the best performance so far.

The landmark matching solution does not require any additional environment instrumentation and it only needs to acquire a collection of the images and extract natural scene landmarks of the environment online or in advance. For example, in [10], the visual landmark database is built from a stereo-rig directly, which may not be global consistent since there is no guarantee that the site is able to provide enough loop closures to reduce the drift. In [1], a sparse 3D reconstruction of an indoor environment is first built using the structure from motion (SfM) techniques on a set of images collected via a pre-calibrated camera. Then a mobile user's orientation and position pose can be directly estimated by registering a captured view with respect to the reconstructed 3D point set. In [7], a set of appearance images is collected to build a complex 3D object using the structure from motion and it is organized into a database and that will be used to recover the camera pose. In [4], the 3D sparse point cloud of the scene is reconstructed from the images collected by a single calibrated cameras using the structure from motion. Then a query image is matched against the collected images of the scene to estimate its camera pose.

Most of the proposed techniques tried to recover the 3D structure using either the pre-calibrated stereo-cameras [10] or a single pre-calibrated camera together with the structure-from-motion techniques [7, 4]. Two major issues with the built landmark database using these 3D reconstruction techniques are the low accuracy of the reconstructed 3D point cloud, especially for the points that are far away from the camera, and the global drift in the whole integrated point cloud due to the accumulated drifts when the site is large. As a result, once these inaccurate 3D scene is used to estimate the camera pose for a query image, it can reduce the pose estimation accuracy dramatically. For example, as shown in [3], if the bias in the stereo based motion estimation is modelled, it is able to lead to a noticeable gain in performance. Therefore, it is essential to build a landmark database that both exhibits a global consistency and contains high-precision landmark positions.

In this paper, we have developed a system to achieve high-precision localization performance using a portable multi-sensor navigation system that can be easily mounted to a helmet or a robot platform. This was achieved by building a custom LIDAR and camera sensor rig mounted on a robot to rapidly collect and build a high-precision 3D Model and landmark database simultaneously. Specifically, the robot traverses the site under remote operation, continually scanning the environment using calibrated video cameras and 3D LIDAR range scanner, building an integrated 3D point cloud and collecting a set of visual landmarks which contains both 2D image features and 3D point locations. The 3D point cloud is used to build a 3D model by fitting planes and adding texture, and the landmarks are organized into a landmark database. During play, the system loads the landmark database and matches to it, thus placing them within a common coordinates system that is aligned with the 3D model.

Compared to the built visual landmark database at [10], there are two major improvements. First, via the lidar scans to obtain the camera poses for each scan, the camera pose of each landmark shot is more accurate in terms of the global consistency. Second, For each visual landmark, its 3d coordinates are replaced with the lidar 3d coordinates, which are much more accurate, especially for those that are far away from the camera.

## 2. System Overview

Our system includes two major components: a lidar-camera sensor unit for 3D model and visual landmark building, and a multi-camera sensor unit for localization.

As shown in Figure 1, the lidar-camera sensor unit includes one lidar sensor (Hokuyo UTM-30LX), two pairs of stereo cameras (Point Grey Flea2 CCD), one IMU unit (CloudCap Crista) and a high-precision rugged pan-tilt unit (Model PTU-D48). Specifically, the lidar sensor is mounted on top of the cameras and the cameras are configured into two stereo pairs, one facing forward and one facing backward. They are all synchronized externally during data collection. The images are captured at $20fps$ with a resolution of $640 \times 480$ pixels. Via the Hokuyo lidar unit, a 3D point is able to be measured with a less than 3cm error within 30 meters. Together with the high-precision PTU-D48 pan-tile unit that provides extremely precise positioning ($0.006°$), it allows to generate a very accurate point cloud for a $360°$ scan.

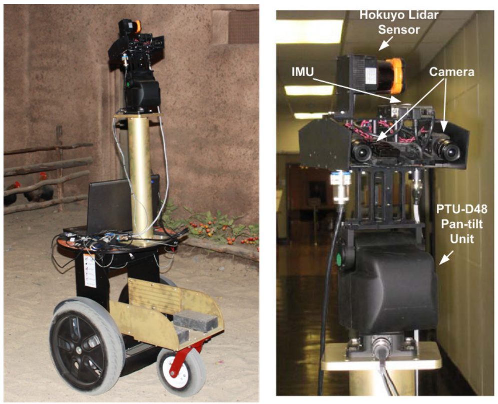


Figure 1. A segway robotic platform (RMP400) with LIDAR-camera sensor unit.

The lidar sensor, cameras, IMU and the pan-tilt unit are calibrated automatically before any data collection. The key calibration is between the lidar sensor and the cameras, which is done via a similar approach to [2]. Once the calibration is done, given a $360°$ lidar scan, we are able to transform a point $\mathbf{X_L}$ in the lidar coordinates system into the coordinates of the front pair's left camera coordinates system $\mathbf{X_C}$ as follows:

$$\mathbf{X_C} = \mathbf{P_{L\to C}}\mathbf{X_L} \tag{1}$$

where $\mathbf{P_{L\to C}}$ is the extrinsic calibration between the lidar and camera.

In order to carry and move the lidar-camera sensor rig easily, a Segway robotic platform (RMP400) is used to station it as shown in Figure 1. Currently, the Segway is driven remotely through the site to build the 3D model and the visual landmark database.

Once the 3D model and the visual landmark database is built, the visual landmark database is used to provide the absolute camera pose of a vision-based localization system in the 3D model. As shown in Figure 2, the vision-based localization sensor consists of two pairs of stereo cameras (AVT ProSilica GigE Camera) and one IMU unit (CloudCap Crista). Specifically, both stereo pairs are synchronized with the IMU and they are placed as one facing forward and the other facing backward, which allows for robust feature tracking even if one pair is completely occluded. All of them are installed compactly inside a ruggedized cover so that they can be easily mounted into a helmet, a robot or vehicle platform for plug and play. As shown in Figure 2, the sensor head is mounted into a helmet and connected to a laptop that can be put into a backpack for people to wear.

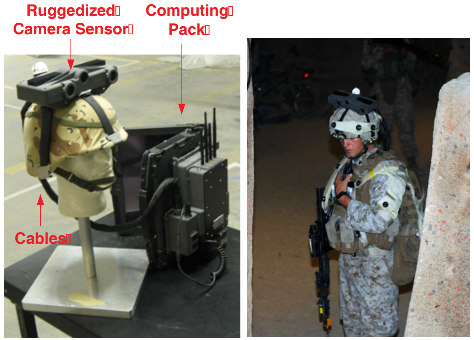


Figure 2. A ruggedized portable vision-based localization sensor system.

### 2.1. Simultaneous 3D Model and Visual Landmark Database Construction

For the localization task in a given site, the first step of our approach is to build a $3D$ model and collect visual landmarks using a mobile platform as shown in Figure 1. As the robot traverses the training site (autonomously or user-controlled), it stops at regular intervals and pans for 180 degrees, recording full omni-directional visual and 3D range data at each position. Each of these local data collections is called a 360° scan, and each 180 degree pan gives uniform 3D points and video over a full 360 degrees. The distribution of the scans should be uniform across the site to ensure consistent landmark matching. Therefore, the robot is tele-operated through the site while continually scanning such that the spacing between scans is kept under five meters. Overlapping 360 degree scans are aligned using ICP (iterative closest point). Since the range of the LIDAR is 30 meters, and the overlapping scans are less than five meters apart, the alignment algorithm is well constrained and is effective in computing the relative position of each scan. After all scans are collected, an accurate and consistent point cloud is built. The last step is to generate a set of visual landmark from each scan. The computed scan position is used to compute an accurate camera pose for each image in the video, and 2D features are extracted from the image. Each 2D feature is associated with a 3D location which can be computed either by stereo matching or by projection of the 3D point cloud.

Specifically, a single 360° scan $S_k$ includes LIDAR, camera and pan tilt unit data recorded from time $t_i$ to $t_j$, which is the time taken by the pan-tilt unit to rotate for scanning. The LIDAR data consists of a set of scan lines $L[t_i...t_j]$, the camera data consists of a set of images $I[t_i...t_j]$, and the pan-tilt unit outputs a corresponding set of lidar poses (one for each scan line or image since they are all synchronized) $P[t_i...t_j]$ formed from its readings. The data is processed automatically, producing a point cloud model of the site as well as a lidar-coordinates consistent visual landmark database as follows:

```
For each scan $S_k$
    (1) integrate $L[t_i...t_j]$ using $P[t_i...t_j]$ to get $X_k$
    (2) query point cloud database $DB_{3D}$ for overlapping scans $X_{DB}$
    (3) align $X_k$ with $X_{DB}$ using ICP algorithm
    (4) transform $X_k$ with $X_k^{'}$ = $P_{ICP}$ $X_k$
    (5) add $X_k^{'}$ to $DB_{3D}$
    (6) transform $P[t_i...t_j]$ with $P'[t_i...t_j] = P_{ICP}^{-1}P[t_i...t_j]$
    (7) extract visual landmarks from $I[t_i...t_j]$ to form a set of landmark shots $LM[t_i...t_j]$
    (8) find the 3d coordinates of each visual landmark from $X_k$ for each landmark shot $LM[t_i...t_j]$
    (9) add $(P^{'}[t_i...t_j], LM[t_i...t_j])$ to the landmark database $DB_{LM}$
End
```

Once all scans have been processed and the two databases have been populated with landmarks and point clouds, post-processing can be done to apply global transformations, remove redundant data, or subsample the point clouds or the landmark shots to a uniform density. Figure 3 shows the 3D model built from the point clouds collected at the Immersive Infantry Trainer (IIT) at Camp Pendleton, CA. Totally, there are 487 scans collected and the built visual landmark database is around 1.8GB. The final aligned scan locations are displayed as red in the bottom of Figure 3.

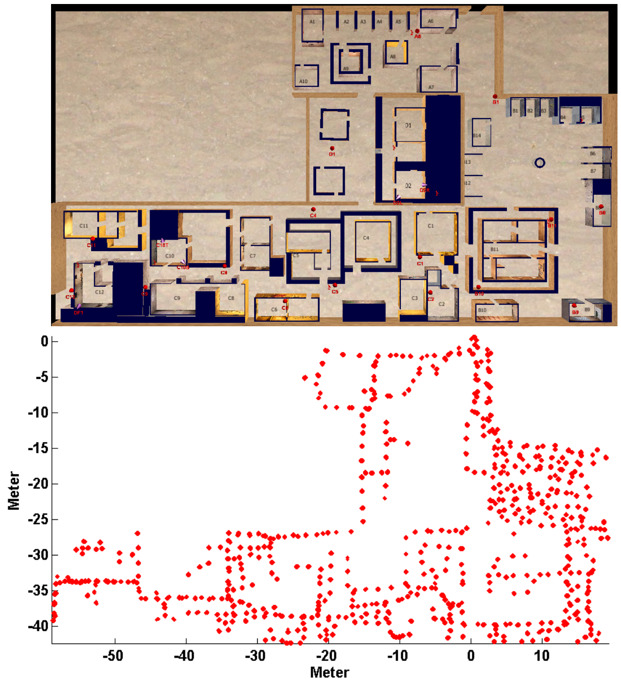


Figure 3. Top: the bird-view of the built 3D IIT model with textures; Bottom: the collected 487 scan positions.

## 3. Localization With Landmark Matching

Figure 4 illustrates the flowchart of the localization system. Mainly, the system contains three components: IMU-fused multi-stereo visual odometry, landmark matching, and Kalman filtering based global localization estimation.

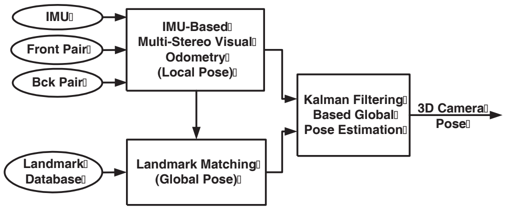

IMU
Front Pair
Bck Pair
IMU-Based Multi-Stereo Visual Odometry (Local Pose)
Landmark Database
Landmark Matching (Global Pose)
Kalman Filtering Based Global Pose Estimation
3D Camera Pose

Figure 4. The flowchart of the proposed localization system.

The system starts with loading the pre-built visual landmark database. Once the landmark database is loaded, the system captures the synchronized stereo image pairs and IMU measurements in real time. The captured stereo pairs and IMU measurements are sent to the visual odometry module that extracts Harris corners from each image and estimates the initial camera pose from the extracted harris corners. Once the initial camera pose is estimated from the image, it is further fused with the IMU measurements to refine the estimation of the camera poses. Via the fusion of the IMU measurements, the system is able to estimate the camera poses (local pose) accurately even when there are few corners extract due to the lack of textures in the scene or the sudden occlusions of one or both cameras.

Once the local camera poses are estimated, the visual odometry module sends the left images together with its extracted Harris corners and the estimated pose to the landmark matching module. Since visual odometry and landmark matching operates in its own thread, it will continue capturing the synchronized stereo pairs and IMU measurements while landmark matching is still being performed.

For the landmark matching, after receiving the extracted Harris corners, the HOG descriptor is extracted for each Harris corner from the received left image. Once the HOG descriptors are extracted, the normalized 2D image coordinates, 3D coordinates, and the HOG descriptors of the corners together with the camera pose is formed as a landmark shot. The landmark shot will serve as a query landmark shot and search the landmark database for a list of potential similar landmark shots and then an image-based 2D matching is performed to refine the potential landmark shot list. Once a matched landmark shot is found and pose estimation is performed to compute a global camera pose. In order to further refine the landmark matching results, a visual odometry trajectory based consistency check is performed, which will be described in Section 3.2.

After a successful landmark matching, due to the possible errors during the landmark matching, the estimated global camera pose are sent to the global localization module for the final camera pose fusion.

### 3.1. Efficient Landmark Matching with Disk-Cache Mechanism

The landmark database size is a function of two factors: 1) coverage area of the training site; 2) density of the landmark shots collected at each scan. For any real-time visual odometry system, fast access to landmark database for matching is critical. A small size landmark database can be kept in memory to enable this fast access, however, when database size grows, keeping entire database in memory is not possible and a portion of it has to be stored in an external disk. Therefore, it is important to devise a disk-cache indexing and searching solution of a large landmark database that enables fast fetching of landmark shots that are in close spatial proximity of the system's current position with a limited memory usage.

The architecture of disk-cache scheme is shown in Figure 5. It has two main components: 1) offline database spatial indexing; and 2) online database search, fetching, and cache updating. Spatial indexing and search is performed by a module called disk-cache manager. For offline spatial indexing of landmark shots, we have used R-trees that allow splitting of coverage area into hierarchically nested region blocks, which are possibly overlapped depending on how landmark scans are distributed and the designated minimum bounding rectangle for each scan. The minimum bounding rectangle is referred to as region block size. The splitting parameter controls how many entries we want at each non-leaf node of R-tree. We set this parameter as 5, which means that the region block is subdivided and new nodes are added if we have more than 5 scans in a region block. The tree is built by iterating over all collected scans, computing centers for each scan, computing their bounding rectangle, and inserting them into the tree. Each leaf node stores the scan id and its bounding box.

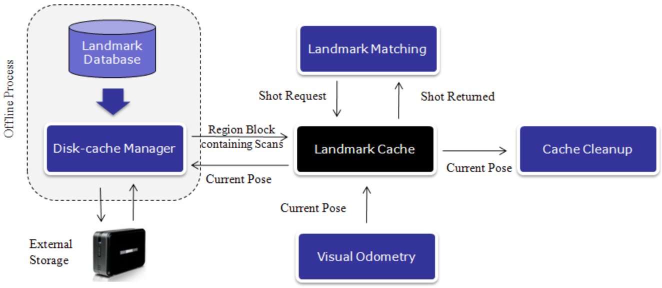


Figure 5. The disk-cache scheme for landmark matching.

In order to compute the scans that are in the neighborhood of the system's current position and bring them into the landmark cache, we use 5x5 meter bounding box around the current location as a search rectangle to perform its intersection with the bounding boxes in the R-tree. The search starts from the root node of the tree. Every internal node contains a set of bounding boxes and pointers to the corresponding child node and every leaf node contains the rectangles of the scan. For every rectangle in a node, the overlap with the search rectangle is computed. If the overlap exists, then corresponding child node is also searched. Searching is done in a recursive manner until all the overlapping nodes have been traversed. When a leaf node is reached, the contained bounding boxes are tested against the search rectangle and all the scan ids within the search rectangle are fetched.

Now that we have landmark database indexing and searching in place, we ensure that landmark scans in the immediate vicinity of the system are always in the memory using above mentioned search strategy so that pose estimation can proceed without any delay. Our cache updating strategy uses two conditions to remove any unwanted scan from the cache. The first condition ensures that scans that are beyond 5x5m bounding rectangle of the current system location are removed first. We can very quickly identify the scans that are not in 5x5m vicinity of system's current position through set intersection and difference operations performed in real-time using R-tree. If there is still not enough cache space available, the second replacement condition ensures that least-used scans are removed first. For this purpose, disk-cache manager maintains a table of last accessed time stamps for each scan that is in the cache.

In addition, in order to reduce the number of disk-access operation, we bring one complete scan into the cache as oppose to individual landmark shots. This results in significant performance gains as disk access is the most expensive operation. Another important factor for integrating the disk-cache into the system is the initial warmup time. To ensure that pose estimation can quickly latch on to a correct location, it is important to have corresponding landmark scans in the memory. We bootstrap this process by uniformly selecting scans over the given spatial region and adding them into the cache during the system startup.

### 3.2. Local Visual Odometry Trajectory Based Consistency Check

After a successful landmark database matching, it will compute a camera pose for the input query landmark shot. In order to improve the robustness of the localization, the camera pose computed from the matched landmarks is first checked with the camera poses from the visual odometry. It serves as an outlier removal stage to remove any possible wrong matches, which works as imposing the temporal consistency over the obtained landmark matching.

Specifically, it operates as follows. As the person travels from location A to location B, the visual odometry will compute a relative pose $\Delta P_{visodo}$ between A and B. Meanwhile, if both images from location A and B are matched to the landmark database successfully, a different relative pose $\Delta P_{LM}$ can be computed from the obtained global camera pose $P_A$ and $P_B$. An error or a pose difference $P_{diff}$ is computed via $\Delta P_{visodo} \times \Delta P_{LM}^{-1}$ first.

Usually, if both the time and the travelled distance between location A and B is small, the local trajectory from A to B can be estimated from visual odometry very accurately. Since $P_{diff}$ encodes both the position and orientation differences, it can be served as a criteria to measure how good a landmark matching is.

### 3.3. Kalman Filter Based Fusion for Global Localization

To generate the final camera pose of each frame in global coordinates system, the local pose estimated from our IMU-based multi-stereo visual odometry module is transformed to the global coordinates system using the global camera pose estimated from the first successful landmark matching.

For each successful landmark matching after the first matching, both the query landmark shot and matched database landmark shot as well as the estimated global camera pose of the query shot are sent to the Kalman filtering based global pose estimation module. Then the global pose transformed from the IMU-based multi-stereo visual odometry module is further fused with the global landmark point measurements in the Kalman filter. The global landmark point measurements are modelled from the 2D to 3D feature point correspondences between the query landmark shot features and the 3D local point cloud on the matched database landmark shot.

We transform every 3D local landmark point $\mathbf{X}$ to the global coordinates system using the estimated global pose of the query shot. We denote pose $\mathbf{P}_{LG} = [\mathbf{R}_{LG} \;\; \mathbf{T}_{LG}]$ such that $\mathbf{X}$ can be transformed via the following equation:

$$\mathbf{Y} = \mathbf{R}_{LG}\mathbf{X} + \mathbf{T}_{LG} \tag{2}$$

which can be written under small error assumption as

$$\hat{\mathbf{Y}} + \delta\mathbf{Y} \simeq (\mathbf{I} - [\boldsymbol{\rho}]_\times)\hat{\mathbf{R}}_{LG}(\hat{\mathbf{X}} + \delta\mathbf{X}) + \hat{\mathbf{T}}_{LG} + \delta\mathbf{T}_{LG}$$

where $\boldsymbol{\rho}$ is a small rotation vector. Neglecting second order terms results in the following linearization

$$\delta\mathbf{Y} \simeq \hat{\mathbf{R}}_{LG}\delta\mathbf{X} + \left[\hat{\mathbf{R}}_{LG}\hat{\mathbf{X}}\right]_\times \boldsymbol{\rho} + \delta\mathbf{T}_{LG} \tag{3}$$

and letting $\tilde{\mathbf{X}} = \hat{\mathbf{R}}_{LG}\hat{\mathbf{X}}$, the local 3D point covariance, $\mathbf{\Sigma}_Y$, can be represented in the global coordinates system in terms of the local reconstruction uncertainty, $\mathbf{\Sigma}_X$, the weight factor of the query landmark shot, $\boldsymbol{W}$, and landmark pose uncertainty in rotation and translation, $\mathbf{\Sigma}_{\mathbf{R}_{LG}}$ and $\mathbf{\Sigma}_{\mathbf{T}_{LG}}$, as

$$\mathbf{\Sigma}_Y \simeq \hat{\mathbf{R}}_{LG}(W\mathbf{\Sigma}_X)\hat{\mathbf{R}}_{LG}^T + [\tilde{\mathbf{X}}]_\times \mathbf{\Sigma}_{\mathbf{R}_{LG}} [\tilde{\mathbf{X}}]_\times^T + \mathbf{\Sigma}_{\mathbf{T}_{LG}}$$

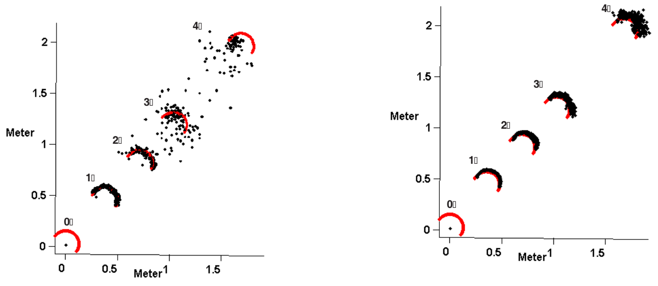
We model the weight factor of the query landmark shot as the distance between the query landmark shot and the matched landmark shot. This factor reflects the accuracy of landmark matching. If the distance is smaller, the landmark matching is more accurate and the resulting 3D uncertainty of all points on the query landmark shot is smaller. The Kalman filter thus relies more on these landmark point measurements since they have small covariance due to accurate matching.

The above is applied to all the point correspondences returned as a result of successful landmark matching. We use the same techniques as [8] to fuse these point measurements inside our Kalman filter.

## 4. Experimental Results

### 4.1. Landmark Matching Performance

In order to test the performance of the landmark matching with fused Lidar measurements, a set of 5 Lidar scans together with the images are collected as shown in Figure 6. At each location, the videos are recorded at 20fps while it rotates 180° to scan for 30 seconds, and so there are totally 600 image frames collected. After the Lidar-point cloud alignment, the camera pose for each image can be computed. For simplicity, only the front cameras are used in the following two experiments, and their computed camera trajectories are shown as red at each location in Figure 6. Due to the high-accuracy of the point cloud, the camera poses computed from Lidar are served as the ground-truth.



(a) (b)

Figure 6. The landmark matching results at different locations: (a) Stereo only; (b)Fused with Lidar measurements.

The first experiment is to extract the visual landmarks from the scan location 0 and forms a reference landmark database, but only the camera poses computed from lidar for each landmark shot are injected. Then the landmark matching is performed to search the reference landmark database for each image at the remaining scan location 1, 2, 3, 4, which are respectively $0.54m$, $1.03m$, $1.53m$, $2.43m$ away from the reference scan location 0. The camera pose for each image is estimated via the landmark matching, and its computed 3d location is plotted as the dark points in Figure 6 (a). As shown in Figure 6 (a), the computed camera pose deviates from their actual camera poses more and more as they move away from the reference scan position 0. Table 1 and Table 2 summarizes the maximal error that contains 69, 95 and 99.7 percentiles of data from the lowest in terms of positions and euler angles (degree).

Table 1. The Computed Positional Upper Error Bounds (cm)

| | Stereo | | | Lidar | | |
|---|---|---|---|---|---|---|
| Scan<br>ID | Lowest<br>69% | Lowest<br>95% | Lowest<br>99.7% | Lowest<br>69% | Lowest<br>95% | Lowest<br>99.7% |
| 1 (0.5m) | 1.70 | 2.80 | 3.76 | 3.03 | 3.63 | 3.97 |
| 2 (1.0m) | 3.28 | 6.79 | 11.41 | 1.40 | 2.24 | 3.10 |
| 3 (1.5m) | 11.02 | 32.96 | 146.70 | 5.20 | 8.29 | 10.14 |
| 4 ( 2.4m) | 26.42 | 67.07 | 106.04 | 8.55 | 12.03 | 14.67 |

Different from the first experiment, once the reference landmark database is extracted, besides injecting the computed camera poses from lidar for each landmark shot, the 3D coordinates of each extracted visual landmark are replaced with the 3D coordinates from the lidar measurements. Then the landmark matching is performed again to search this lidar-point integrated reference landmark database for each image at the test scan locations. The estimated camera poses from the landmark matching are plotted as the dark points in Figure 6 (b). As shown in Figure 6 (b), although the estimated camera poses deviate more from their actual ones as the scans move away from the reference scan 0, their deviations are much smaller. From the Table 1 and Table 2, we can see that for a scan that is 2.4 meters away from the reference position, landmark matching is able to provide a positional error within $14.67cm$ and an angular error with 0.61 degree for 99.7% of the data, which is a significant improvement.

Table 2. The Computed Angular Upper Error Bounds (Degree)

| | Stereo | | | Lidar | | |
|---|---|---|---|---|---|---|
| Scan<br>ID | Lowest<br>69% | Lowest<br>95% | Lowest<br>99.7% | Lowest<br>69% | Lowest<br>95% | Lowest<br>99.7% |
| 1 (0.5m) | 0.11 | 0.21 | 0.39 | 0.11 | 0.19 | 0.27 |
| 2 (1.0m) | 0.19 | 0.46 | 0.67 | 0.16 | 0.30 | 0.38 |
| 3 (1.5m) | 0.43 | 1.47 | 2.51 | 0.17 | 0.35 | 0.60 |
| 4 ( 2.4m) | 1.29 | 3.25 | 4.28 | 0.27 | 0.45 | 0.61 |

### 4.2. Global Pose Fusion

In the following experiments, a landmark database is built from 128 lidar scans collected inside a building, whose locations are shown as pink circles in Figure 7. The database is around 0.69GB and contains 17539 landmark shots.

In the experiment, a person who wearing our system started near the front entrance, walked around the first floor through the hallways, rooms and then went outside through the back entrance and went inside through the front entrance. It lasts 5 minutes and 38 seconds, and the person travelled around 223 meters. As shown in Figure 7, the system locates itself inside the building immediately as the system turns on and estimates the travelled trajectories of the person, which is marked as red. Along the trajectory, there are totally 819 landmark matching returned from the landmark database, which are marked as blue circles in Figure 7. After imposing the local visual odometry trajectory-based consistency check, a set of bad landmark matching hits are removed and only 592 good landmark matchings are passed through. Those passed landmark matching hits are marked as dark in Figure 7, and they are utilized for final localization fusion via kalman filtering.

As shown in the bottom portion of Figure 8, a portion of the estimated trajectory is enlarged. Since each landmark matching contains a different level of error, the accuracy of the computed camera pose varies. If we accept the computed camera poses directly from the landmark matching and resets the estimated trajectory, the final trajectory is marked as red in Figure 8, which apparently is jumpy and contains large localization errors in certain portion. However, via the proposed fusion scheme via the Kalman filtering, it will automatically give higher weights to those landmark matching hits with higher accuracy and generate a smoother and more accurate trajectory as shown the blue curve in Figure 8.

Figure 9 (a) shows the distance between the query landmark shot and the matched landmark shot over a 60-frame

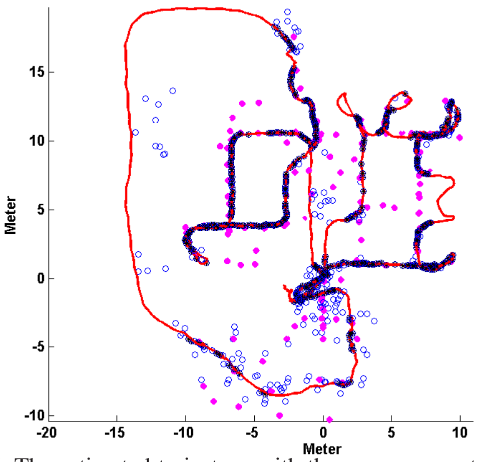


Figure 7. The estimated trajectory with the poses computed from landmark matching.

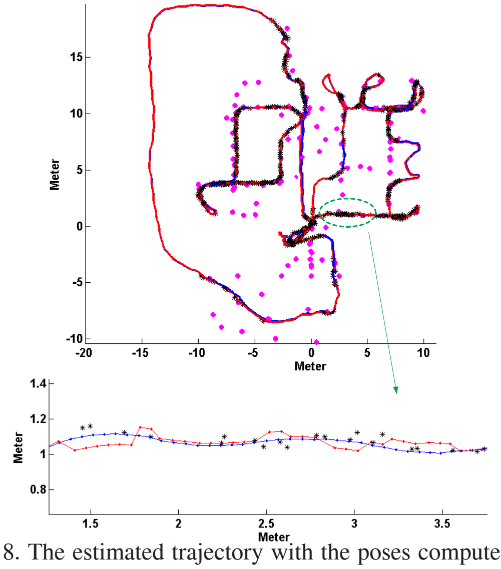
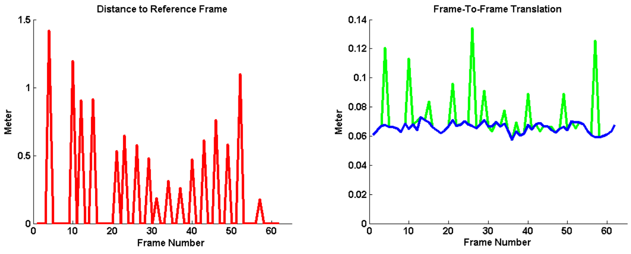


Figure 8. The estimated trajectory with the poses computed from landmark matching.

period taken from the whole trajectory. Figure 9 (b) shows frame-to-frame pose translation estimated before (green) and after (blue) global fusion with Kalman filtering respectively. Since the walking speed of the user doesn't change much in a very short period (such as one frame, 0.0677 seconds), the translation between frames should be very smooth.

However, the peaks of the green curve in Figure 9 (b) caused by landmark matching (Figure 9 (a)) generates the jitter/jump of pose estimation. Big distance between the query landmark shot and the matched landmark shot may cause errors in pose estimation from landmark matching. By capturing the 3D uncertainty of landmarks and thus relying more on accurate landmark shots (with small distance to the matched shots) in the Kalman filter, we can reduce the errors in the final global pose estimation.



(a) (b)

Figure 9. (a) The computed distance between the current frame to its matchedreference landmark shots; (b) The frame-to-frame distance comparison before and after kalman filtering fusing.

## 4.3. Real-World Operation Results

To demonstrate the performance of our proposed localization system, a week-long test has been conducted on the marines who were performing training exercises in the IIT training facility. During their training exercises, the marines will walk, run, crouch, shoot, stay in dark region or in crowds, etc., and the system will go through a set of very difficult conditions that will fail the visual odometry alone badly. However, with the constant landmark matching hits, we are able to track them within around 15cm error very robustly for hours. The ruggedized machine we are using is DELL Latitude E6400 XFR, which is a dual-core 2.8Ghz machine with 2GB memory. The whole system runs at 10fps and the frequency of landmark matching thread maintains stably at 2HZ. Together with our ruggedized sensor head, it even survived after being banged into wall or door-frames constantly while they were in exercises. Each training session lasts around one hour and 8 helmet sensor heads are used at each session. There are four sessions every day and our system is able to provide accurate positions for each marine, which is wireless-transmitted back to a server machine that displays their positions in the 3D model and then projects it into the wall.

Figure 10 shows the red estimated trajectory with the first landmark matching only and displayed in the 3D model. We can see that without continuous landmark matching, the drift of the marine's location grows continuously and it fails to tell the correct location of each marine after a while. However, with the continuous corrections from the landmark whenever there is a chance, our proposed system is able to track the marine's location very accurately when they move through the whole training site. Its estimated trajectory is displayed as blue in the 3D model.

Figure 11 show a total of 17 test runs for a group of marines who were engaged in the real training exercises inside the IIT training site.

As shown in an online video "http://www.zhiweizhu.com/demo/Online_Tracks.avi", as the marines went into the training site, out system is able to immediately lock their positions inside the training site and track them accurately as they move through the whole

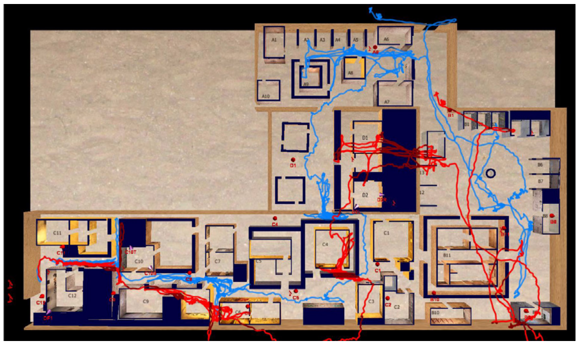


Figure 10. The comparison between the estimated trajectories with continuous landmark correction or not.

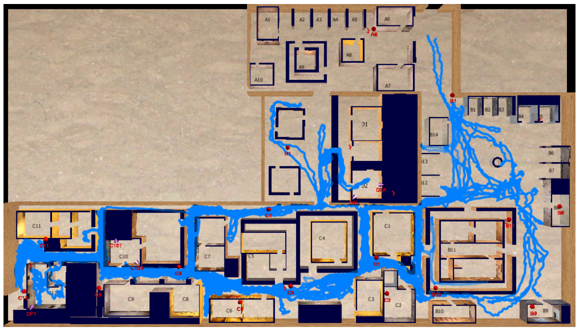


Figure 11. The bird view of the estimated 3D trajectories overlayed in the IIT 3d model.

site. With the use of the 3D model, the detailed position and orientation information of each marine, as well as their surroundings in the real environment is able to be viewed from various angles in the 3D model as shown in Figure 12, which is very important for the after action review for each training exercise.

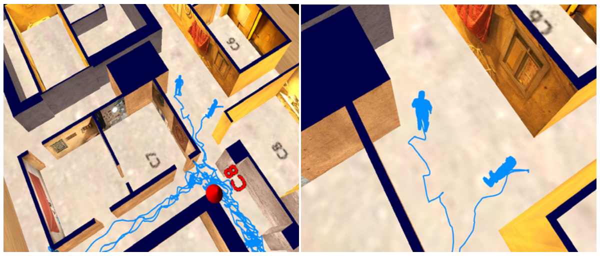


Figure 12. Different camera views of the people in the IIT 3d model.

## 5. Application for Augmented Reality

Augmented reality training systems using Head Mounted Displays (HMDs) require very high precision knowledge of the 3D location and 3D orientation of the user's head. This is required by the system to know where to insert the synthetic actors and objects in the HMD. The inserted objects must appear stable and not jitter or drift. We have developed a system to achieve this performance using our proposed localization system mounted on the HMD. A short video demo can be downloaded at "http://www.zhiweizhu.com/demo/AugmentedReality.wmv", which shows a training scenario that inserts a group of virtual guys for the trainer to interact with.

## 6. Conclusion

In this paper, we showed that when a high-precision visual landmark database is built, it is able to improve the landmark matching performance dramatically. We built a Lidar-Camera sensor rig mounted on a mobile robot that is able to traverse around to collect both video and 3D range data to build a global consistent and accurate visual landmark database. Once the landmark database is built for a given environment, a disk-cache mechanism is proposed to achieve real-time performance for the landmark matching. In order to reduce the possible landmark matching errors, a local visual odometry trajectory based consistency check is performed to remove some large errors and a subsequent kalman-filtering based fusion is used to further smooth out some small landmark matching errors. Experiments on the week-long real training exercises on a group of marines in the IIT training site have confirmed both robustness and high-precision of our proposed localization system.

## References

[1] C. Arth, D. Wagner, M. Klopschitz, A. Irschara, and D. Schmalstieg. Wide area localization on mobile phones. In *IEEE International Symposium on MAR'09*, 2009. 81

[2] S. Bileschi. Fully automatic calibration of lidar and video streams from a vehicle. In *IEEE 3DIM Workshop*, 2009. 82

[3] G. Dubbelman and F. Groen. Bias reduction for stereo based motion estimation with applications to large scale visual odometry. In *IEEE Conference on CVPR'09*, 2009. 81

[4] A. Irschara, C. Zach, J.-M. Frahm, and H. Bischof. From structure-from-motion point clouds to fast location recognition. In *IEEE Conference on CVPR'09*, 2009. 81

[5] G. Klein and D. Murray. Improving the agility of keyframe-based slam. In *IEEE Conference on ECCV*, 2008. 81

[6] A. Levin and R. Szeliski. Visual odometry and map correlation. In *IEEE Conference on CVPR'04*, 2004. 81

[7] J. Mooser, S. You, U. Neumann, and Q. Wang. Applying robust structure from motion to markerless augmented reality. In *IEEE WACV'09*, 2009. 81

[8] A. Mourikis and S. Roumeliotis. A multi-state constrained kalman filter for vision-aided inertial navigation. In *IEEE Conference on ICRA'07*, 2007. 81, 85

[9] 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*, 2007. 81

[10] Z. Zhu, T. Oskiper, S. Samarasekera, R. Kumar, and H. Sawhney. Real-time global localization with a pre-built visual landmark database. In *IEEE Conference on CVPR'08*, 2008. 81, 82

[11] Z. Zhu, T. Oskiper, S. Samarasekera, H. Sawhney, and R. Kumar. Ten-fold improvement in visual odometry using landmark matching. In *IEEE Conference on ICCV*, 2007. 81