Fiducial Planning for Error-Bounded Pose Estimation of a Panoramic Camera in Large Environments

Daniel G. Aliaga Ingrid Carlbom
{aliaga|carlbom}@bell-labs.com
Lucent Technologies Bell Labs

1. INTRODUCTION

Panoramic image sensors are becoming increasingly popular because they capture large portions of the visual field in a single image. These cameras are particularly effective for capturing and navigating through large, complex 3D environments. Existing vision-based camera pose algorithms are derived for standard field-of-view cameras, but few algorithms have been proposed to take advantage of the larger field-of-view of panoramic cameras. Furthermore, while existing camera pose estimation algorithms work well in small spaces, they do not scale well to large, complex 3D environments consisting of a number of interconnected spaces.

Accurate and robust estimation of the position and orientation of image sensors has been a recurring problem in computer vision, computer graphics, and robot navigation. Stereo reconstruction methods use camera pose for extracting depth information to reconstruct a 3D environment [12, 16]. Image-based rendering techniques [1, 3, 15, 19, 20, 28] require camera position and orientation to recreate novel views of an environment from a large number of images. Augmented reality systems [5] use camera pose information to align virtual objects with real objects, and robot navigation and localization methods [7, 8, 10, 30] must be able to obtain the robot's current location in order to maneuver through a (captured) space.

We can divide existing vision-based camera pose approaches into passive methods and active methods. Passive methods derive camera pose without altering the environment but depend on its geometry for accurate results. For example, techniques may rely upon matching environment features (e.g., edges) to an existing geometric model or visual map [11, 29, 34]. To obtain robust and accurate pose estimates, the model or map must contain sufficient detail to ensure correspondences at all times. Another class of passive methods, self-tracking methods, use optical flow to calculate changes in position and orientation [17]. However, self-tracking approaches are prone to cumulative errors making them particularly unsuited for large environments.

Active methods utilitize fiducials, or landmarks, to reduce the dependency on the environment geometry. Although fiducial methods are potentially more robust, the number and locations of the fiducials can significantly affect accuracy. Existing techniques often focus on deriving pose estimates from a relatively sparse number of (noisy) measurements [4, 6, 9, 18, 21, 27]. For large arbitrarily shaped environments, such as the ones presented in this

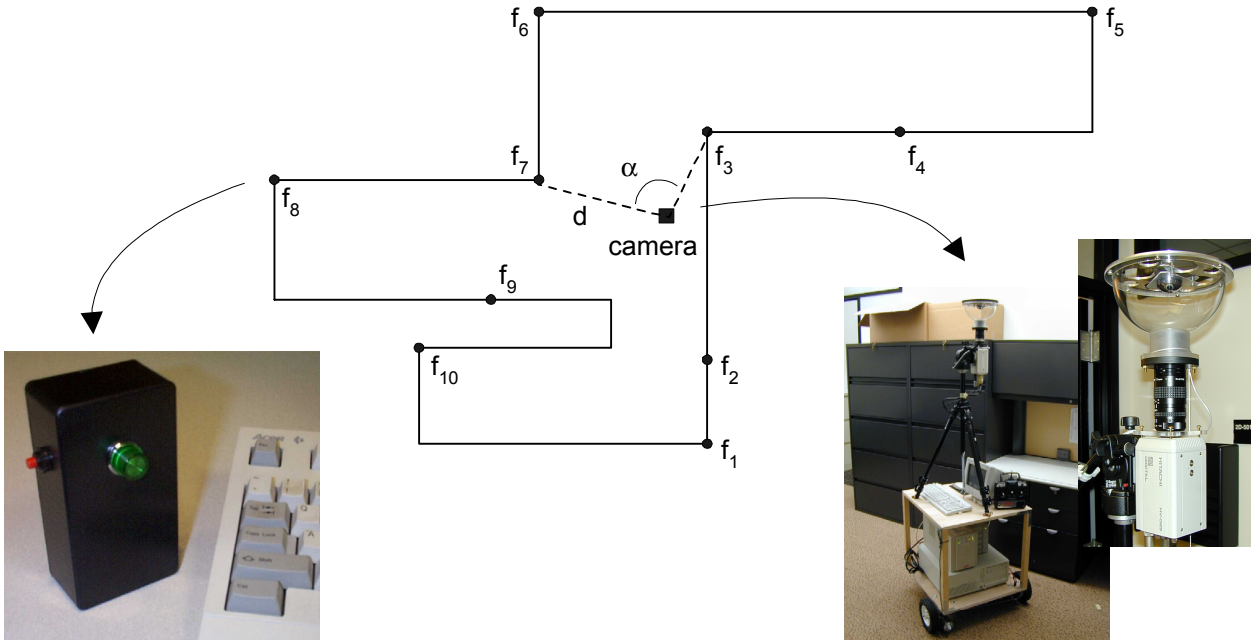


Figure 1. Example setup. *We show a floor plan and fiducial locations that from all camera viewpoints within the environment satisfy a set of visibility, distance, and angle constraints. To the left, we show a picture of one of our small portable fiducials. To the right, we show our remote-controlled capture system, including a computer, panoramic camera, battery, and motorized cart.*

article, there does not exist a method for determining the optimal number of fiducials or their optimal placement in order to achieve a desired pose accuracy.

In this article, we present a robust camera pose algorithm and a working system to compute bounded-error estimates of the position and orientation of panoramic images captured within large, arbitrarily complex environments while moving the camera within a plane. We use a planning algorithm to place fiducials in an environment so as to satisfy a set of fiducial constraints, including the number of visible fiducials, the distance from the viewpoint to the fiducials, and the angle subtended by pairs of fiducials. Combined with an analytic error model, we can either provide fiducial placements to achieve a desired pose estimation accuracy, or bound the pose estimation error for a given fiducial placement (Figure 1).

Our algorithm inserts small portable fiducials (e.g., light boxes) into an environment and triangulates camera pose from the projections of the fiducials onto the panoramic images. We use a coarse 2D floor plan and a heuristical solution to a variation of the classical art-gallery problem to suggest fiducial locations that satisfy the fiducial constraints for viewpoints within the environment. Exact fiducial locations are not necessary and will be obtained later via an optimization method. At the expense of more fiducials, enforcing stricter constraints increases pose estimation accuracy. Our system requires little setup time and does not significantly alter the environment. We have

used our method with several environments, covering 500 to 1000 square feet and with an average pose accuracy of up to 0.66 cm. Our approach includes the following contributions:

- **Planning Algorithm**: our fiducial planning algorithm provides fiducial placements for arbitrarily complex environments. By creating a network of fiducials, we can estimate pose in large environments, potentially consisting of multiple interconnected spaces (or rooms). Solutions vary from those containing a minimal-number of fiducials to solutions containing a highly-redundant set of fiducials that provide high-accuracy pose estimation; and,
- **Error Model**: our error model conservatively approximates the region of pose uncertainty allowing us to determine bounds on pose estimation error. Moreover, it allows us to propose fiducial placements for achieving a desired degree of pose accuracy.

The article is organized as follows. In the following section, we highlight related work. Section 3 describes our fiducial planning algorithm while Section 4 describes fiducial tracking and pose estimation. Section 5 explains the error model and Section 6 provides implementation details of both our algorithm and complete system. We present the results of our planning algorithm and error model in Section 7. Finally, we conclude with a discussion of future work in Section 8.

2. RELATED WORK

Many approaches to camera pose estimation for large 3D environments have been proposed in the literature. Some methods rely purely on computer vision techniques, while others combine computer vision with additional sensor data (e.g., global positioning systems or GPS) or with interactive techniques. Another group of approaches install complex hardware infrastructures within the environment in order to obtain sensor measurements. In this section, we highlight some approaches that are particularly relevant.

Structure from motion techniques [14, 22, 25] track environment features during an image sequence and, using an optimization, obtain camera pose as well as 3D information about the scene. While the results are promising, it is difficult for frame-to-frame tracking systems to scale to long image sequences in large environments. Occlusion changes and drift, particularly in large interconnected spaces, hinders accurate and robust camera pose estimation.

Taylor [30] describes a system for computing pose using panoramic images. The user selects points and edges as features in a number of keyframes within an image sequence. A reconstruction method obtains the camera pose for the keyframes and the 3D location of the features. Subsequently, the features are tracked from keyframe to keyframe, yielding camera pose in all frames. Scaling to long sequences in large environments burdens the user with the task of manually determining which features to select in order to obtain good pose estimates.

The goal of the MIT City scanning project [31] is to compute pose for images captured over a city-size environment. They employ GPS data to initialize a vision-based system. Subsequently, they exploit the typical

building structures present in city-size environments to refine the camera pose estimates. Their method has created images databases, with camera pose, for a large campus-size environment. Their approach is specific to outdoor environments with building-like structures and does not work in indoor environments.

There exist several hardware trackers for computing the position and orientation of a small sensor. Such hardware uses magnetic, acoustic, or optical signals to locate the sensor (e.g., Polhemus, Ascension). The sensor can be attached to a camera in order to obtain camera pose. Recently, 3rd Tech Inc. developed a commercial version of the UNC Ceiling Tracking project [33], which uses infrared LEDs as fiducials in the ceiling panels of an interior office environment. By determining which LEDs are visible by the sensor, the system triangulates the position and orientation with very high accuracy. Unfortunately, all of these devices require complex hardware installations and not all of them yield high precision results.

3. FIDUCIAL PLANNING

3.1 Placement Constraints

The objective of our planning algorithm is to place fiducials so that predefined fiducial constraints are satisfied for all viewpoints within an environment. Our approach is inspired by the classical art-gallery problem: given the floor plan of an art gallery, determine the minimum number of guard positions so that every part of the gallery is visible by at least one guard [24]. The general solution to this problem is NP-complete, but a number of approximate solutions exist. We reformulate this problem to address the placement of fiducials (instead of guards). We position fiducials so as to satisfy the following constraints:

- **Visibility**: at least $V \geq 2$ fiducials must be visible from every position in the environment to allow pose estimation by triangulation.
- **Distance**: the distance d from the viewpoint to any fiducial used for pose estimation must be less than or equal to a predetermined distance D. This reduces errors from measuring excessively large distances as well as prevents the tracking of fiducials whose projections become too small.
- **Angle**: the angle α subtended by the vectors from the viewpoint to at least one pair of currently visible fiducials, both meeting the distance constraint, is greater or equal to a predetermined angle $A \leq 180$ degrees. This avoids small acute angles that may lead to numerical problems.

3.2 Planning Algorithm

We use a heuristic approach to find a near-minimum number of fiducial locations, f_i, for a given set of constraint values V, D, and A. The first step creates an initial sufficient number of fiducials by decomposing the floor plan into convex planar polygons (using a binary space partition) and placing a fiducial at every polygon vertex. The polygons are further subdivided until the distance and angle constraints are met. To ensure that viewpoints within

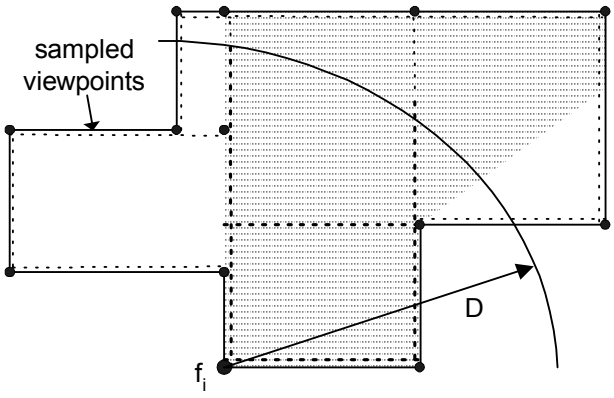


Figure 2. Fiducial removal. *To ensure the fiducial constraints are still met after removing fiducial f_i, we verify the constraints from sampled viewpoints within distance D from the fiducial and on the perimeter of the visible convex regions. The shaded area represents the area visible from f_i and the circular arc represents the border of the area within distance D.*

all polygons meet the visibility constraint, the polygon edges are partitioned until at least V vertices per polygon are present.

The next step iteratively removes fiducials to obtain a near-minimum set for satisfying the constraints. In order to decide which fiducials to remove, we prioritize fiducials based on their effectiveness in satisfying the constraints. Fiducials at reflex vertices (e.g., corners) are the most desirable because they form an approximate solution to the art-gallery problem for a convex decomposition of space. Fiducials in the interior of the environment are least desirable because they may interfere with camera movement. Hence, we separate the initial fiducial set into three lists: interior fiducials, wall fiducials, and reflex fiducials. Then, we attempt to reduce the number of fiducials by first removing the interior fiducials, then the wall fiducials, and finally the reflex fiducials, while still satisfying the constraints of visibility, distance, and angle.

To find the most redundant fiducial within a particular list, we calculate a redundancy value, r_i, that estimates how removing the fiducial affects the local constraints. First, we identify which fiducials can be removed while still satisfying the constraints. Then, from among these fiducials, we remove the one with the highest redundancy value. We calculate the redundancy value as a weighted sum of how much the minimum number of visible fiducials, v_{min}, differs from V, how much the minimum distance to another fiducial, d_{min}, differs from D, and how much the minimum subtended angle, α_{min}, differs from A, using weights w_v, w_d, and w_α, respectively:

$$r_i = w_v \left| v_{\min} - V \right| + w_d \left| \frac{D - d_{\min}}{D} \right| + w_\alpha \left| \frac{\alpha_{\min} - A}{180 - A} \right|. \qquad (1)$$

To determine if a fiducial can be safely removed, we temporarily remove the fiducial and verify the constraints from all viewpoints visible from the removed fiducial's location and within distance D (Figure 2). For each convex region

of viewpoints, we check the constraints from a dense sampling of viewpoints along the perimeter and exploit the following three properties to ensure the constraints are met for all interior viewpoints:

1. If from every viewpoint along the perimeter of a convex region at least V fiducials outside the region are visible, then for any viewpoint inside the region, at least V fiducials outside the region are also visible.
2. The maximum distance between a viewpoint inside a convex region to a particular fiducial outside the region is less than the largest distance between a viewpoint on the perimeter of the region and the same fiducial.
3. The smallest angle subtended by vectors from a viewpoint inside a convex region to a pair of fiducials outside the region is greater than the smallest angle subtended by vectors from a viewpoint on the perimeter of the region and the same fiducial pair.

4. TRACKING AND POSE ESTIMATION

As our panoramic camera moves through the environment in a plane, typically at eye-height, we capture images to disk and track the projections of the fiducials placed in the environment. We can generate approximate camera poses during capture or we can perform an offline global optimization to obtain more accurate pose estimates. In this section, we describe our tracking algorithm, pose estimation method, and global optimization.

4.1 Tracking

The tracking algorithm calculates for every image the projection of the visible fiducials. The algorithm is initialized with either a user-provided camera pose estimate or user-identified fiducial projections. The algorithm predicts a fiducial's projected position in subsequent images from an approximation to its image-space linear velocity. To handle occlusion changes, the 3D fiducial locations in the floor plan determine which fiducials should be visible in the image. The actual projection of a fiducial is searched for in a small window surrounding the predicted projection of the fiducial.

Within each search window, the fiducial projection is selected from among all approximately circular blobs of pixels that exceed an intensity threshold. To decide if a blob of pixels is circular, we compare its actual circumference and aspect ratio to that of a circle of similar radius. We initialize the tracker with the blob whose radius is closest to the predicted radius, but for subsequent images we choose the blob that best correlates with the fiducial projection of the previous image and has a radius similar to the predicted radius.

4.2 Pose Estimation

We obtain position and orientation estimates by triangulating the camera with pairs of tracked fiducials that are within distance D from the camera and subtend an angle greater than or equal to A degrees. If the number of

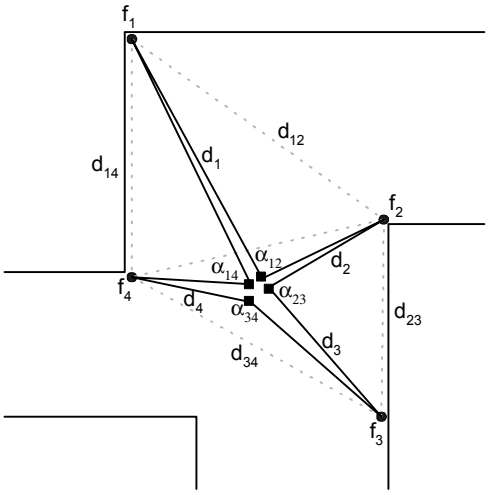


Figure 3. Optimization. *Given multiple triangulations, we perform a bundle adjustment to minimally perturb the pose estimates and fiducial locations so as to obtain the set of fiducial locations and pose estimates that minimizes the difference between the fiducial projections and the tracked fiducials. We show four position estimates using fiducials f_1, f_2, f_3, and f_4.*

tracked fiducials is T, then we obtain at most $R=\binom{T}{2}$ valid fiducial pairs and estimates. In the real-time system, we combine these estimates using weighted averaging.

To calculate the camera position and orientation (x_{ij}, y_{ij}, ω_{ij}) relative to the i-th and j-th fiducial, we use the tracked fiducial coordinates (u_i, v_i) and (u_j, v_j), and a calibrated camera model. First, we obtain the distances d_i and d_j, as projected onto the horizontal plane, between the camera's focal point and both fiducials. Then, we compute the angle α_{ij} between the vectors from the focal point to the fiducials and we estimate the distance between the two fiducials (d_{ij}). Using these values, we solve an over-determined triangulation to obtain the camera's position and orientation relative to the fiducial pair. Finally, by using the world-space coordinates of the fiducials, we derive the camera's position and orientation in world-space.

4.3 Global Optimization

To determine a globally consistent set of camera poses and 3D fiducial locations, we use bundle adjustment [32], a non-linear least squares optimization method. We alternate between computing pose estimates from the fiducial locations and computing the fiducial locations from a subset of the camera pose estimates (e.g., about 10% of the pose estimates uniformly distributed through the dataset).

The goal of the bundle adjustment is to find for each fiducial i its 3D location (X_i, Y_i, Z_i) and for each image k its global camera pose ($\hat{x}_k$, $\hat{y}_k$, $\hat{\omega}_k$) that minimizes the difference between the observed fiducial projections (u_{ik}, v_{ik}) and the projections of the current fiducials. The function $P(X_i, Y_i, Z_i, \hat{x}_k, \hat{y}_k, \hat{\omega}_k)$ encapsulates the projection from world-space onto our panoramic images [23]. We assume that the observed error is zero-mean Gaussian, thereby

bundle adjustment corresponds to a maximum likelihood estimator. The error term for bundle adjustment is given below (the Cronecker delta term δ_{ik} is 1 when fiducial i is tracked on image k):

$$e = \sum_i \sum_k \delta_{ik} \left\| \vec{P}_{ik}(X_i, Y_i, Z_i, \hat{x}_k, \hat{y}_k, \hat{\omega}_k) - (u_{ik}, v_{ik}) \right\| \qquad (2)$$

When this process has converged, we obtain a set of panoramic images captured on a plane with calibrated camera parameters.

5. ERROR MODEL

We develop an error model in order to bound the uncertainty of the pose estimates computed by our algorithm. Given fiducials placed in an environment, an image sequence through the environment, and tracked fiducial projections, the model computes for each image the region of the environment that may contain the camera's center-of-projection (COP). In addition, the error model allows us to formulate a tradeoff between the fiducial constraints and the pose estimation accuracy.

There are multiple sources of error that cause uncertainty in the pose estimates. In our system, the dominant source of error is the positional uncertainty of the fiducials. The global optimization begins with approximate fiducial locations and obtains optimal position estimates for the fiducials. Nevertheless, there is still uncertainty in their positioning. Since camera pose is triangulated from their 3D positions, this directly affects the accuracy of camera pose.

Angular measurement errors are another source of uncertainty, especially for distant fiducials. Since it is difficult to determine the center of each fiducial projection with subpixel accuracy, we assume all angle measurements to have an error equal to the angle subtended by a pixel.

There are additional sources of uncertainty, in particular tracking errors and camera calibration errors. Our fiducials are, by design, easy to track so we can generally assume good tracking. We do avoid gross tracking errors by ignoring outliers. Calibration errors for our camera are small and are close to the limiting accuracy of a panoramic sensor of our type and resolution [2].

Figures 4(a-b) depict the region of uncertainty for pose estimates assuming only distance measurements. Given the position of a single fiducial with error e and a distance estimate d between the camera's COP and the fiducial, the camera may lie anywhere within an annulus surrounding the fiducial (Figure 4a). For multiple fiducials and distance estimates, the region of uncertainty can be constructed by intersecting the annuli surrounding each fiducial (Figure 4b).

Our camera pose algorithm also provides us with measurements of the observed angle between the camera's COP and pairs of fiducials. These angle measurements further restrict the region of uncertainty for the pose estimates. In particular, for a given pair of tracked fiducials, we use the cosine rule to formulate a relationship between the

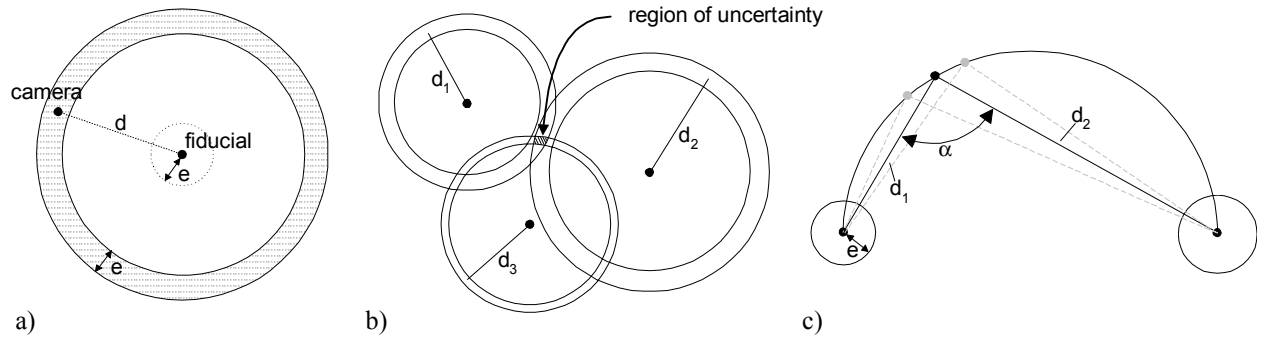


Figure 4. Region of uncertainty for pose estimates*. (a) If the distance estimate to a fiducial is d and the position of the fiducial is known with error e, then the region of uncertainty of the camera is an annulus surrounding the fiducial. The radius of the middle of the annulus is the distance estimate d while the "width" of the ring is equal to the error e. (b) Given multiple distance estimates and fiducials, the camera may lie anywhere within the region of uncertainty formed by the intersection of all the annuli. (c) If we are also given a measurement of the observed angle between the camera and two fiducials, we can further limit the distance values and thus further reduce the size of the region of uncertainty.*

distance estimates and the observed angle. For example, as shown in Figure 4c, an observed angle of α=90 degrees restricts the camera to lie on a (semi) circle that passes through both fiducials. (In general, the camera is restricted to lie on a smooth curve that can be computed from the cosine rule; but, since we are usually concerned with relatively small segments of this curve, we always assume the curve to locally be an arc of a circle.) A fiducial positioning error of e can also be interpreted as a fixed fiducial position and a distance estimate d_i with error e. If we force the distance estimate d_1 to be its smallest value (i.e., we subtract the error e from the original distance estimate), then the angle measurement dictates a maximum value for distance d_2. If we force the distance estimate d_1 to its largest value, then we obtain the minimum value for distance d_2. Similarly, we repeat this operation for distance d_2. These limits on the distance values further reduce the width of the annuli and thus the size of the region of uncertainty.

We approximate the overall region of uncertainty by using axis-aligned bounding boxes. To intersect a pair of annuli, we compute the exact intersection (e.g., by intersecting pairs of circles) and then surround the intersection with an axis-aligned bounding box. To intersect one annuli intersection with another, we instead intersect the bounding boxes. In our results section, this error model is used to conservatively report bounds on pose uncertainty as half the length of the bounding box diagonals. The actual camera pose error may be smaller.

6. IMPLEMENTATION DETAILS

We implemented our system in C/C++ running under Windows in a client-server arrangement. The server sits on a motorized cart that also carries the panoramic camera, disks to store capture data, and a large battery (Figure 1).

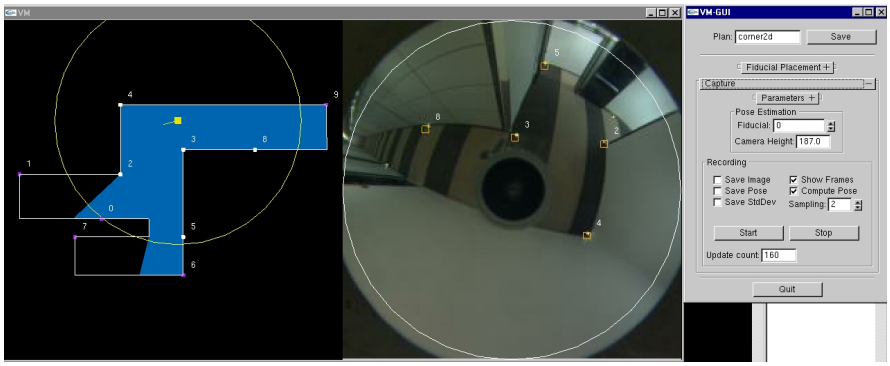


Figure 5. System interface*. The client interface allows a user to specify the fiducial constraints and generate fiducial locations and, subsequently, to receive graphical feedback from the server during a tracking session. The server tracks the fiducials and computes the pose estimates. Afterwards, a global optimization refines the estimated camera poses and fiducial locations.*

The client computer is a laptop that communicates with the server via a wireless Ethernet connection. The motorized cart is controlled via a radio remote control unit.

Our panoramic camera uses a high-resolution 3-CCD color video camera (JVC KY-F70, 1360x1024 progressive pixels at 7.5Hz) and a calibrated paraboloidal catadioptric system [2, 13, 23] based on a commercial Cyclovision/Remote Reality S1 unit. Each frame contains the bottom-half of a hemispherical view of the environment. We capture frames and transfer-to-disk at an average rate of 6.5 frames-per-second. (The decrease from 7.5Hz to 6.5Hz is due to disk performance and not algorithm overhead.) Since the cart moves at an average speed of 0.2 m/sec, we are simulating a capture frame rate of roughly 30Hz for a robot moving at a speed of 1 m/s.

The server and client statically partition the tasks so as to reduce the server load while also maintaining low network traffic. The server captures frames, tracks fiducials, and computes estimated camera pose. The client maintains the floor plan and information about the fiducial configuration. For every frame, the client uses the current pose estimate to send the server the current fiducial visibility estimate. The server sends back the camera pose. For graphical feedback, the server optionally transmits a low-resolution captured frame and the location of the tracked fiducials.

The client application consists of a graphical interface that communicates with the server via TCP/IP (Figure 5). A typical session begins with a display of a floor plan. The interface allows the user to select the constraint values and compute fiducial locations. To begin a capture and tracking session, the user provides an initial estimate of the position and orientation of the cart or uses the mouse to click on initial fiducial projections.

The server application waits for instructions from the client. Upon receiving a camera pose estimate and a fiducial visibility estimate, it attempts to find and track the actual fiducials in the current captured frame. The resulting

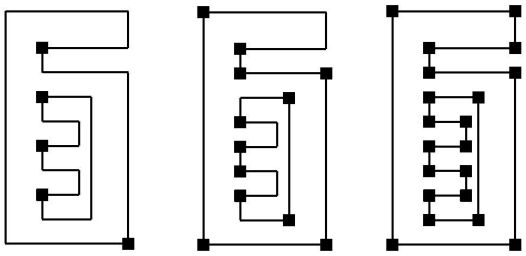


Figure 6. Visibility constraint*. We show three fiducial sets for different minimum number of fiducials V in the same environment. Left: V=1 (fiducial locations also correspond to a set of guard positions for the related art-gallery problem). Middle: V=2. Right: V=4. Other placement constraints remain constant at D=∞ and A=0 degrees.*

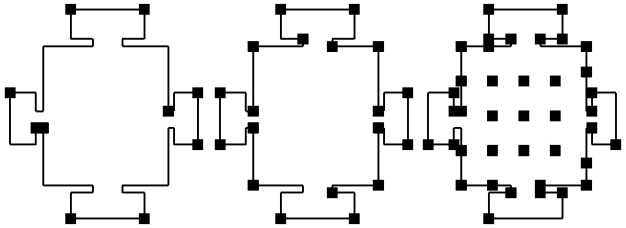


Figure 7. Distance constraint. *We show three fiducial sets for different maximum distance D to fiducials in the same environment. Left: D=∞. Middle: D=1/4 of the floor plan diagonal. Right: D=1/8 of the floor plan diagonal. Other placement constraints remain constant at V=2 and A=0.*

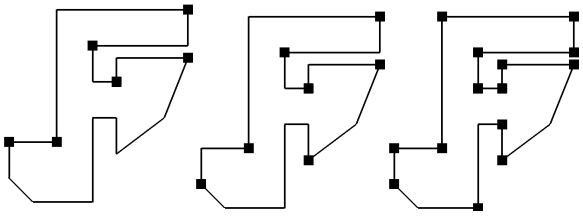


Figure 8. Angle constraint*. We show three fiducial sets for different minimum angles A in the same environment. Left: A=0 degrees. Middle: A=60 degrees. Right: A=120 degrees. Other placement constraints remain constant at D=∞ and V=2.*

camera pose estimate is transmitted back to the client. Frames and pose data are optionally saved to disk in real-time. After capture, the user may perform the global optimization over the captured data to improve pose estimates.

We use small light bulbs as fiducials because they are easy to track and their projections are at least a few pixels wide even at a distance of 5 to 7 meters. Figure 1 shows one of our battery-powered fiducials. After tracking, we could use image post-processing techniques to replace the small fiducial projections with the average local image color. For the weights used to compute the redundancy value of a fiducial, we assigned w_v a large weight so as to encourage prompt removal of fiducials in areas were many are visible. For the other weights, w_d and w_α, we choose

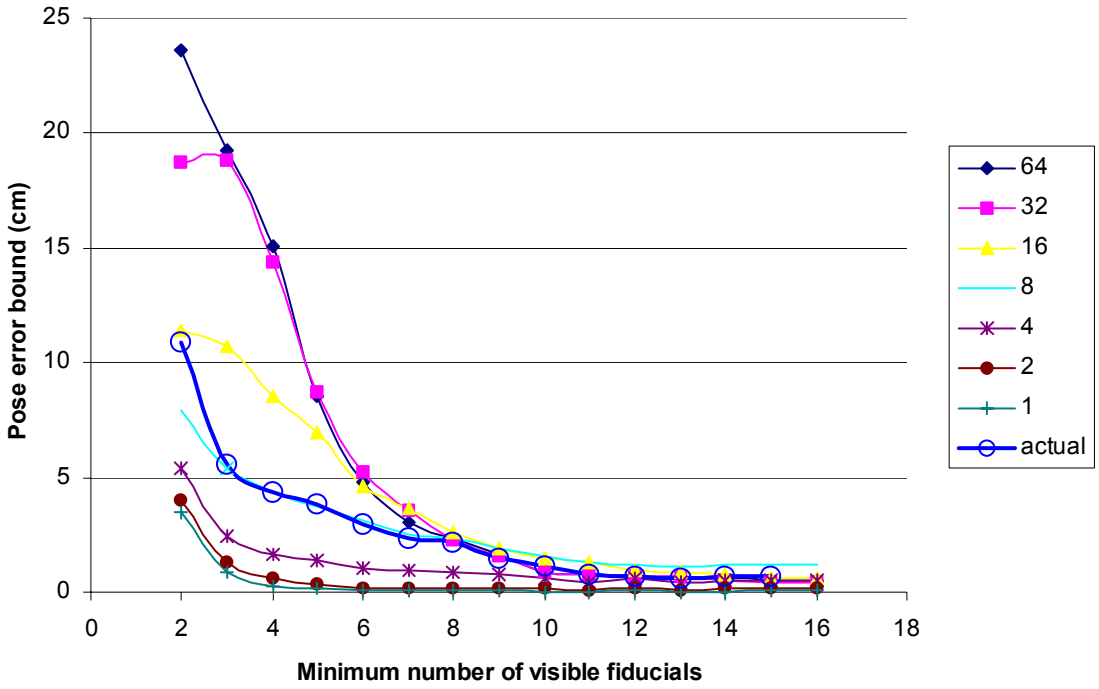


Figure 9. Pose error vs. minimum number of visible fiducials*. This graph shows how varying numbers of visible fiducials and varying fiducial placement accuracies affect pose estimation. The horizontal axis indicates the minimum number of visible fiducials for any image. Each graph line corresponds to a different fiducial placement accuracy (in cm). The vertical axis shows a bound on the mean pose estimation error as determined by our error model using the example sequence through a test environment.*

values that, on average, scale the corresponding terms to approximately equal values thus equalizing the tradeoff of distance and angle.

7. RESULTS

We have used our approach to generate fiducial positions and camera pose for several environments. Computing fiducial positions is relatively fast. On a 1 GHz PC it takes, on average, less than a minute to compute fiducial sets for the environments shown in this article. The floor plans for the test environments were obtained by making simple straight-line measurements using a tape measure. Alternatively, the floor plans could be easily extracted from CAD files, if such are available.

Figures 6-8 demonstrate the fiducial planning algorithm in three example environments. In the first environment (Figure 6), the fiducial placement guarantees the minimum number of visible fiducials V is 1, 2 or 4, respectively. The solution for V=1 corresponds to a set of guard positions for the classical art-gallery problem. In the second environment (Figure 7), the fiducial placement guarantees at least two fiducials are always visible and the

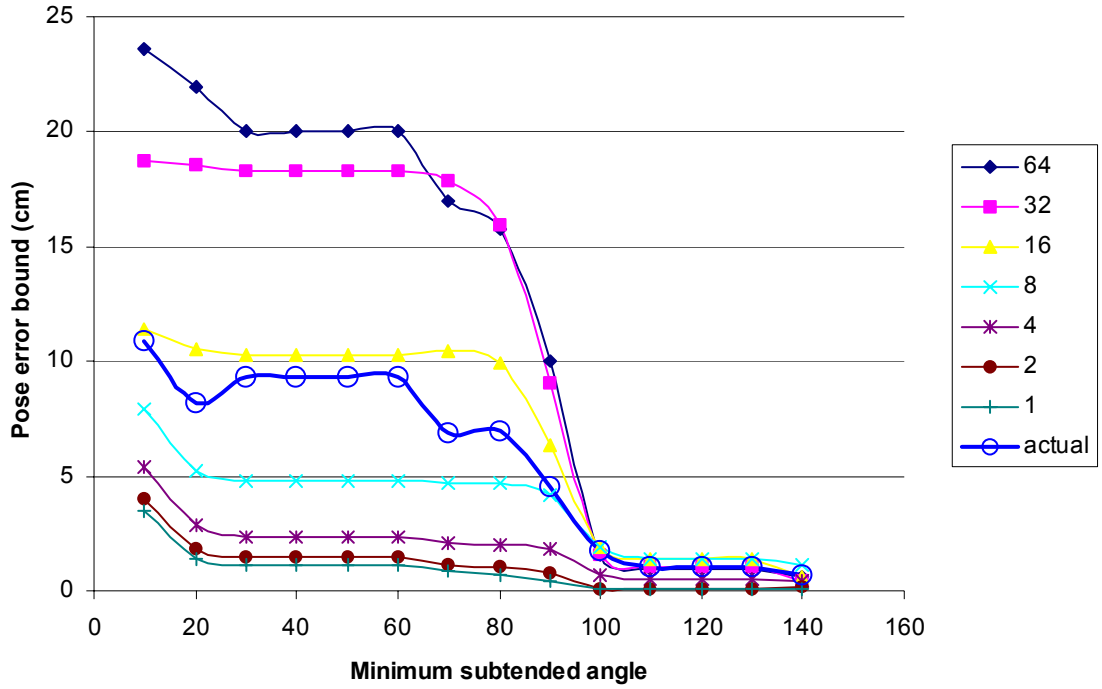


Figure 10. Pose error vs. minimum subtended angle*. This graph shows how varying minimum subtended angles between fiducial pairs and varying fiducial placement accuracies affect pose estimation. The horizontal axis indicates the minimum subtended angle. Each graph line corresponds to different fiducial placement accuracies (in cm). The vertical axis shows a bound on the mean pose estimation error as determined by our error model using the example sequence through a test environment.*

maximum distance *D* to the fiducials is either unbounded, 1/4 of the floor plan diagonal, or 1/8 of the floor plan diagonal. In the third environment (Figure 8), the fiducial placement guarantees for any viewpoint within the floor plan at least two fiducials are visible and the angle *A* between one pair of fiducials and the camera is at least 0, 60, or 120 degrees, respectively.

To better understand how pose estimation is affected by the number of visible fiducials, the angle subtended by pairs of fiducials, and the distance to the fiducials, we plot pose error bounds for several fiducial sets in a single test environment. While the general behavior of these plots is intuitive, we are able to provide insights into the subtle interdependencies among the fiducial constraints and calculate "sweet spots" in the tradeoff of fiducials and accuracy. Figures 9-11 show the relationship between pose error bounds, fiducial position error, and fiducial constraints by varying one constraint at a time. The horizontal axes represent how a fiducial constraint is enforced while the vertical axes represent a conservative bound on the mean pose estimation error obtained by our error model. Each curve in a figure represents the bound for a given fiducial placement accuracy. Tables 1-3 show the

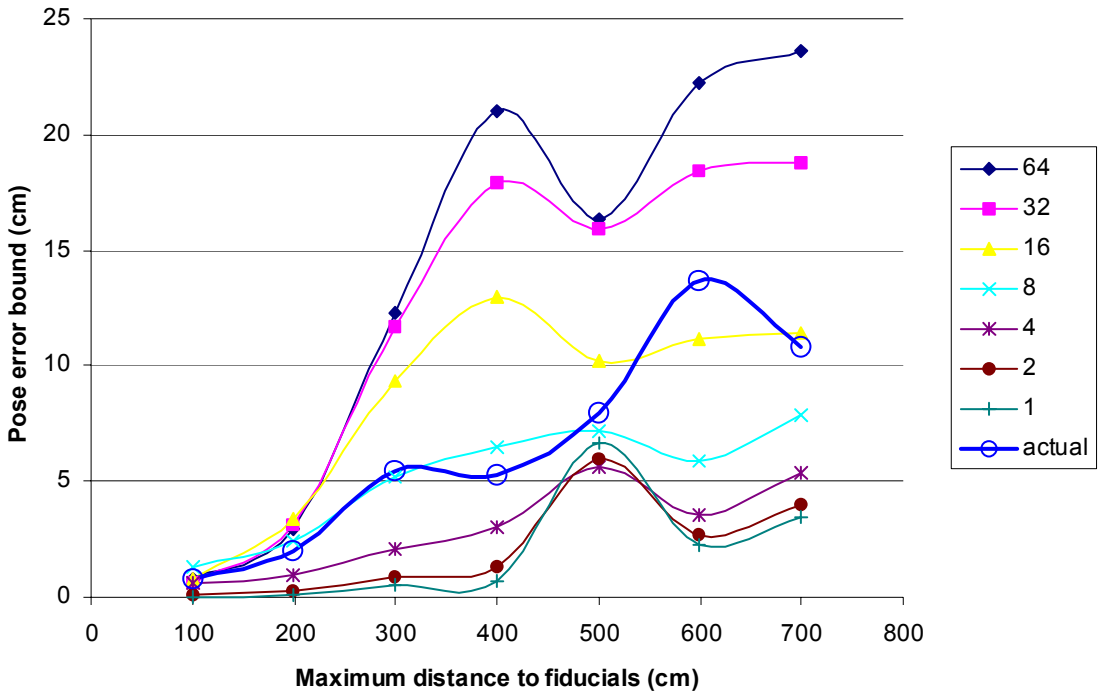


Figure 11. Pose error vs. maximum distance to fiducials*. This graph shows how varying maximum distances to fiducials and varying fiducial placement accuracies affect pose estimation. The horizontal axis indicates the maximum distance to fiducials. Each graph line corresponds to different fiducial placement accuracies (in cm). The vertical axis shows a bound on the mean pose estimation error as determined by our error model using the example sequence through a test environment.*

same data in table form and include the standard deviation of each bound. As expected, the standard deviations of the more accurate solutions are smaller.

The test environment measures 7 by 10 meters in size and contains several interconnected spaces (Figure 12). Rather than capturing a test image sequence in the same environment many times with different fiducial sets, we place a fiducial at every meter along all the walls of the test environment, capture a single long image sequence (4205 images), and then use subsets of these fiducials, computed by our planning algorithm, as the fiducial sets.

Figure 9 depicts the effect of varying the minimum number of visible fiducials. For instance, in order to obtain less than 1 cm of mean pose estimation error with as little as 5 visible fiducials, we need to compute the position of the fiducials to within 2 cm of error. If for the same minimum number of visible fiducials we can compute their position with approximately 32 cm of accuracy, we can only compute pose with up to 8 cm of error on average (and a larger standard deviation).

Figure 10 shows a similar graph for the minimum angle constraint. Using minimum angles beyond 90 to 100 degrees demonstrate a significant improvement. Smaller angles tend to produce pairs of fiducials that are placed

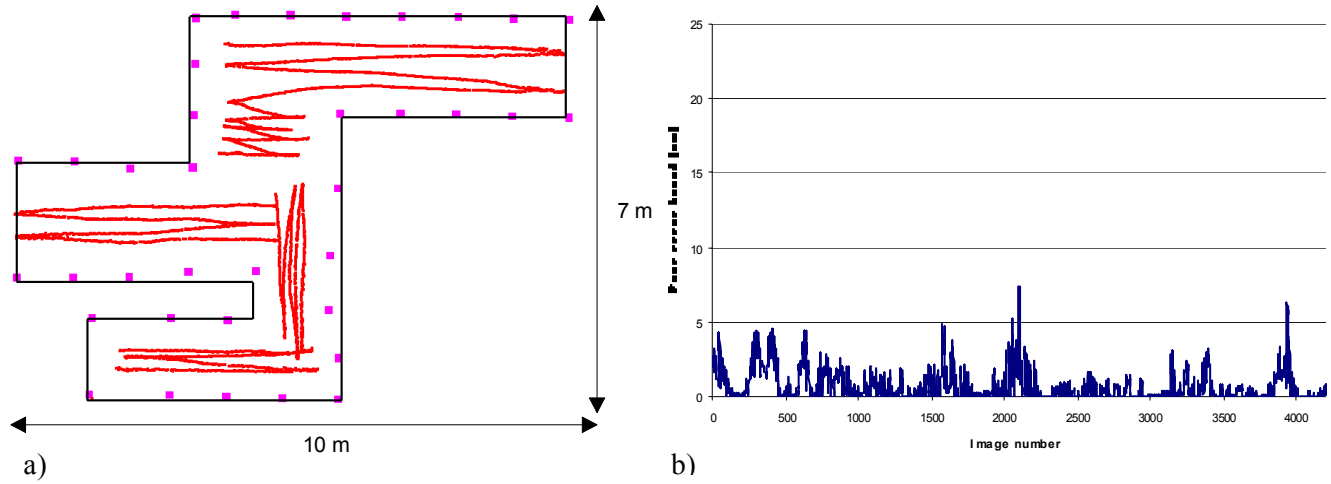


Figure 12. Pose computations in our test environment. *(a) We show the floor plan of our test environment, the location of all the available fiducials (colored boxes), and the computed camera trajectory (using all the fiducials). (b) We show the pose error bounds for the image sequence (actual pose error may be smaller). The mean pose error bound is 0.66 cms and its standard deviation is 0.97 cms.*

nearby and potentially at a far end of an open area. This results in large pose estimation errors. The intersection region of a pair of annuli from two fiducials is smallest when the subtended angle from the camera's COP to the fiducials is near 90 degrees. Enforcing larger subtended angles, increases the number of fiducials, on average, and thus further improves pose estimates.

Figure 11 shows the graph for the maximum distance constraint. For fiducial distances up to 200 centimeters, we get very accurate pose estimates while for fiducial distances approaching 400 centimeters we start to see a large range of pose estimation errors. This result is of course dependent on the resolution of the panoramic sensor. The graph also shows some non-monotonic behavior caused by the fact that the allowable distances to the fiducials is limited by the geometry of the test environment.

We also show in Figures 9-11, the actual pose error bounds obtained by using the fiducial position errors of our global optimization (labeled "actual" in the figures). We estimate the accuracy of our fiducial positions by computing the average re-projection error of the fiducials. In other words, we compute by how much do all images disagree on the global world-space position of each fiducial. (This is the same error term used for bundle adjustment.) In practice, the fiducial positioning accuracy increases as the pose estimation accuracy increases. Our best fiducial positioning accuracy is 7.81 cm. For high accuracy solutions, notice how the curve for the actual pose error bound behaves similar to the theoretical curve for fiducial errors of 8 cm.

Figure 12 presents the pose estimation results for the test image sequence. Figure 12a shows the estimated position of the fiducials in the test environment and the computed trajectory of the camera. Figure 12b shows the actual bounds on pose estimation error for the same test sequence. The apparent noise in this curve is due to the

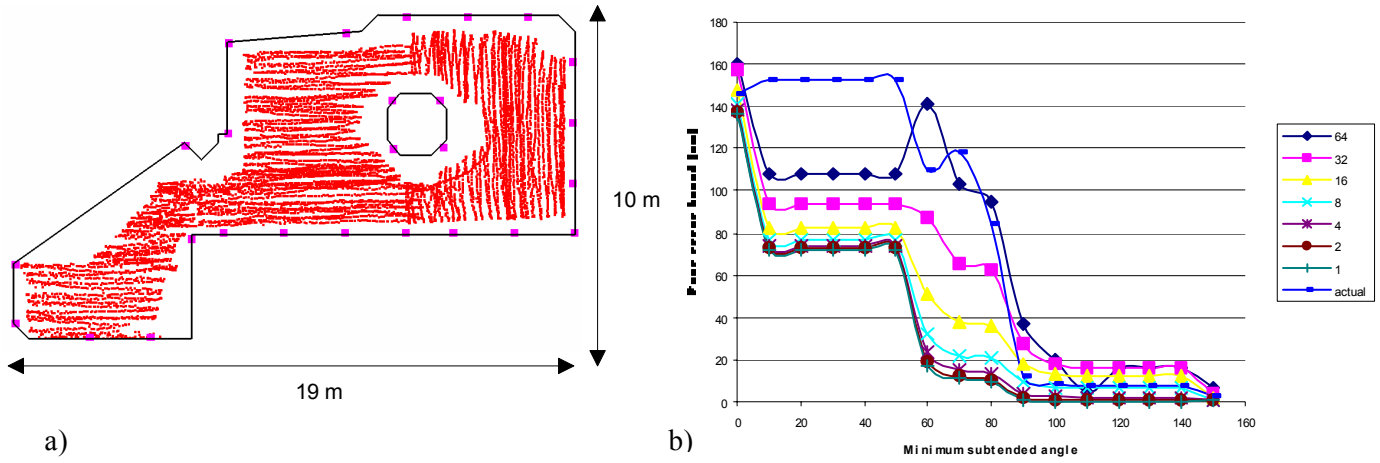


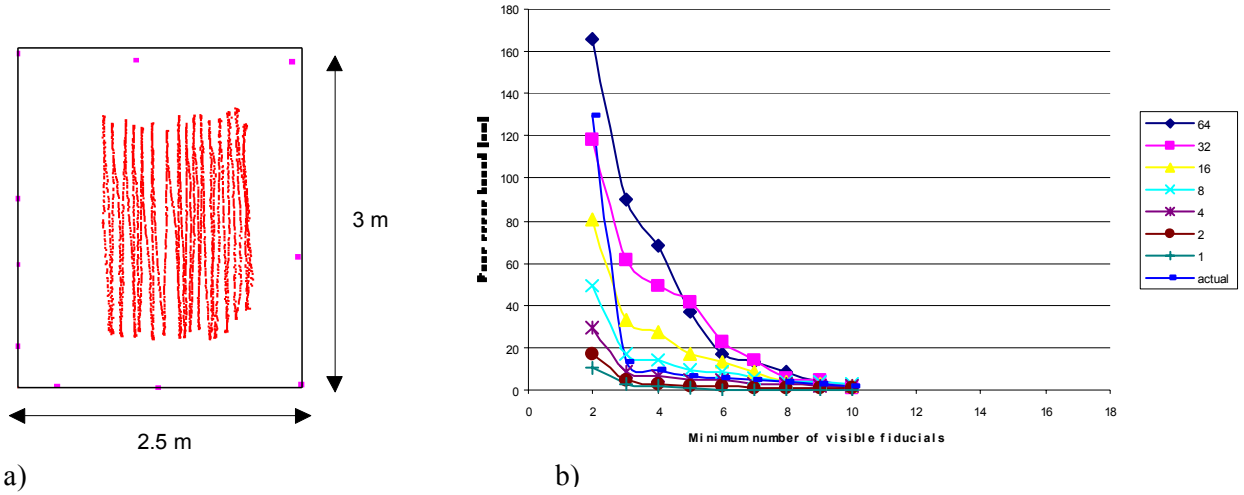
Figure 13. Pose computations in a museum environment. *(a) We show the floor plan of a museum environment, the location of all the available fiducials (colored boxes), and the computed camera trajectory (using all the fiducials). (b) We show the pose error bounds for a range of values for the minimum subtended angle constraint.*

conservative nature of our error bounds and not (necessarily) to noise in the actual pose estimates. In some images, intersecting annuli produce long and slim regions. Our axis-aligned bounding boxes do not approximate these regions very well.

We have used our method to capture several large image databases that have successfully been used to recreate 3D environments [3]. Figures 13 and 14 report the pose estimation bounds for two of these datasets. Figure 13 is a museum measuring 10 by 19 meters in size and with a sequence of 9832 panoramic images. Figure 14 is a small office measuring 2.5 by 3 meters in size and with a sequence of 3475 panoramic images. For the museum, a small angle constraint yields few and distant fiducials. For the office, the error bounds for the solutions with few fiducials are large because the positioning errors are relatively large compared to the size of the environment. Furthermore, given the simple geometry of the room, the angle subtended by fiducials (in particular for the two fiducial case) is often near 180 thus yielding long and slim uncertainty regions.

8. CONCLUSIONS AND FUTURE WORK

We have described an approach to pose estimation particularly suited for panoramic cameras. This approach includes automatic planning of fiducial placements in large complex environments. The fiducial positions are subject to a set of constraints, which in conjunction with an error model enables pose estimation at any desired accuracy. Furthermore, we have presented and analyzed the results for several large indoor environments, consisting of multiple interconnected spaces. Our results have provided us with insights into the complex interplay between fiducial placement and accuracy.



a) b)

Figure 14. Pose computations in a small office environment. *(a) We show the floor plan of a small office environment (i.e., a simple box), the location of all the available fiducials (colored boxes), and the computed camera trajectory (using all the fiducials). (b) We show the pose error bounds for the image sequence using subsets of the fiducials.*

We would like to extend our system to support a self-guided capture mechanism. Using computer-controlled motors, we could guide the capture cart through the environment automatically. Since, we can compute approximate pose in real-time, we could let the system capture images until a desired image trajectory is acquired or a desired image density is obtained. Simultaneously, we could continually perform bundle adjustment to refine the estimated fiducial locations and camera poses.

With regards to our panoramic sensor, we would like to remove the restriction of moving the camera within a plane and also experiment with higher resolution cameras. Both of these improvements have the potential of further refining camera pose estimates.

	64		**32**		**16**		**8**		**4**		**2**		**1**		**actual**	
	mean	σ	mean	σ	mean	σ	mean	σ	mean	σ	mean	σ	mean	σ	mean	σ
100	0.77	2.37	0.59	1.55	0.74	1.47	1.28	1.48	0.60	0.88	0.10	0.27	0.02	0.07	0.81	1.21
200	2.94	6.10	3.14	4.46	3.36	3.67	2.44	2.13	0.99	1.07	0.23	0.48	0.05	0.13	2.02	1.90
300	12.29	11.80	11.65	9.39	9.32	7.37	5.20	4.07	2.07	2.80	0.89	2.35	0.55	2.14	5.46	4.62
400	21.02	13.06	17.89	10.61	12.99	8.38	6.49	4.80	3.03	4.13	1.33	2.53	0.73	1.76	5.30	4.60
500	16.32	12.25	15.92	12.94	10.22	8.56	7.21	7.63	5.65	10.11	5.93	11.90	6.67	12.98	7.93	8.13
600	22.19	11.15	18.41	11.86	11.15	9.11	5.86	7.04	3.52	6.02	2.65	5.52	2.22	4.86	13.69	9.90
700	23.60	11.50	18.75	11.77	11.40	8.98	7.91	10.25	5.40	9.40	3.98	8.36	3.46	8.06	10.85	9.90

Table 1. Pose error bounds vs. maximum distance to fiducials*. Rows correspond to fiducial sets computed for fiducial distances of 100 to 700 cm in the test environment. Each column contains the mean and standard*

deviation of the pose error bound for fiducial positioning errors of 64, 32, 16, 8, 4, 2, and 1 cm. The last column pair ("actual") refers to the fiducial positioning errors actually obtained via the global optimization.

	64		32		16		8		4		2		1		actual	
	mean	σ	mean	σ	mean	σ	mean	σ	mean	σ	mean	σ	mean	σ	mean	σ
2	23.60	11.50	18.75	11.77	11.40	8.98	7.91	10.25	5.40	9.40	3.98	8.36	3.46	8.06	10.85	9.90
3	19.27	12.47	18.80	10.99	10.75	6.06	5.41	4.09	2.44	3.36	1.27	3.25	0.85	3.22	5.58	4.72
4	15.10	12.65	14.38	9.69	8.56	5.40	4.33	2.61	1.64	1.39	0.60	0.74	0.26	0.43	4.37	3.22
5	8.57	11.00	8.71	7.59	6.93	5.07	3.73	2.58	1.37	1.36	0.38	0.61	0.13	0.33	3.82	2.90
6	4.81	7.98	5.22	5.52	4.63	4.06	3.11	2.48	1.08	1.13	0.20	0.42	0.07	0.20	2.98	2.55
7	3.06	6.33	3.57	4.35	3.67	3.44	2.55	1.98	0.93	0.94	0.21	0.46	0.06	0.21	2.36	2.04
8	2.39	5.83	2.31	3.70	2.58	2.58	2.33	2.00	0.89	0.98	0.20	0.45	0.06	0.21	2.18	2.01
9	1.63	5.17	1.57	3.33	1.94	2.31	1.89	1.86	0.81	0.91	0.19	0.44	0.05	0.27	1.50	1.91
10	0.88	2.44	0.86	1.91	1.46	1.95	1.57	1.69	0.61	0.87	0.16	0.43	0.04	0.21	1.12	1.31
11	0.77	2.32	0.76	1.83	1.26	1.91	1.33	1.56	0.45	0.77	0.10	0.34	0.04	0.21	0.81	1.14
12	0.73	2.23	0.67	1.83	0.92	1.86	1.26	1.56	0.57	0.97	0.19	0.50	0.07	0.30	0.74	1.19
13	0.64	2.05	0.66	1.73	0.83	1.67	1.14	1.31	0.45	0.77	0.12	0.30	0.04	0.10	0.62	0.95
14	0.59	1.95	0.54	1.39	0.79	1.68	1.24	1.47	0.54	0.81	0.14	0.33	0.04	0.11	0.73	1.10
15	0.56	1.86	0.47	1.24	0.60	1.27	1.20	1.42	0.52	0.81	0.16	0.36	0.05	0.13	0.70	1.01
16	0.56	1.86	0.47	1.24	0.60	1.27	1.20	1.42	0.52	0.81	0.16	0.36	0.05	0.13	0.70	1.01

Table 2. Pose error bounds vs. minimum number of visible fiducials*. Rows correspond to fiducial sets computed for 2 to 16 visible fiducials in the test environment. Each column contains the mean and standard deviation of the pose error bound for fiducial positioning errors of 64, 32, 16, 8, 4, 2, and 1 cm. The last column pair ("actual") refers to the fiducial positioning errors actually obtained via the global optimization.*

	64		32		16		8		4		2		1		actual	
	mean	σ	mean	σ	mean	σ	mean	σ	mean	σ	mean	σ	mean	σ	mean	σ
0	23.60	11.50	18.75	11.77	11.40	8.98	7.91	10.25	5.40	9.40	3.98	8.36	3.46	8.06	10.85	9.90
10	23.60	11.50	18.75	11.77	11.40	8.98	7.91	10.25	5.40	9.40	3.98	8.36	3.46	8.06	10.85	9.90
20	21.97	12.24	18.53	10.92	10.57	7.07	5.19	5.10	2.88	5.25	1.84	5.04	1.43	4.82	8.20	8.00
30	20.03	12.12	18.30	11.52	10.29	7.52	4.83	4.26	2.34	3.47	1.47	3.34	1.16	3.31	9.34	8.34
40	20.03	12.12	18.30	11.52	10.29	7.52	4.83	4.26	2.34	3.47	1.47	3.34	1.16	3.31	9.34	8.34
50	20.03	12.12	18.30	11.52	10.29	7.52	4.83	4.26	2.34	3.47	1.47	3.34	1.16	3.31	9.34	8.34
60	20.03	12.12	18.30	11.52	10.29	7.52	4.83	4.26	2.34	3.47	1.47	3.34	1.16	3.31	9.34	8.34
70	16.98	12.24	17.88	11.28	10.47	7.65	4.74	3.98	2.11	2.67	1.17	2.29	0.83	2.10	6.87	6.16
80	15.80	11.41	15.91	10.59	9.90	7.48	4.70	3.81	1.99	2.49	1.01	2.01	0.69	2.01	6.95	6.08
90	10.01	11.37	9.05	9.54	6.39	6.75	4.14	5.18	1.81	3.37	0.79	2.04	0.42	1.39	4.52	6.13
100	1.54	3.70	1.54	2.83	1.92	2.84	1.91	2.13	0.69	1.19	0.13	0.47	0.05	0.25	1.72	2.60
110	0.95	2.67	1.05	2.51	1.42	2.54	1.40	1.85	0.49	0.99	0.11	0.55	0.05	0.44	1.06	1.83
120	0.95	2.67	1.05	2.51	1.42	2.54	1.40	1.85	0.49	0.99	0.11	0.55	0.05	0.44	1.06	1.83
130	0.95	2.67	1.05	2.51	1.42	2.54	1.40	1.85	0.49	0.99	0.11	0.55	0.05	0.44	1.06	1.83
140	0.57	1.87	0.46	1.22	0.61	1.32	1.15	1.37	0.47	0.74	0.14	0.38	0.05	0.17	0.66	0.97

Table 3. Pose error bounds vs. minimum subtended angle*. Rows correspond to fiducial sets computed for angles of 0 to 140 degrees in the test environment. Each column contains the mean and standard deviation of the pose error bound for fiducial positioning errors of 64, 32, 16, 8, 4, 2, and 1 cm. The last column pair ("actual") refers to the fiducial positioning errors actually obtained via the global optimization.*

Acknowledgments

We are thankful to Sid Ahuja, Multimedia Communications Research VP at Bell Labs, for supporting this research. In addition, we thank Bob Holt for his mathematical help.

References

[1] D. Aliaga, I. Carlbom, "Plenoptic Stitching: A Scalable Method for Reconstructing Interactive Walkthroughs", *Proceedings of ACM SIGGRAPH*, pp. 443-450, 2001.

[2] D. Aliaga, "Accurate Catadioptric Calibration for Real-time Pose Estimation in Room-size Environments", *IEEE International Conference on Computer Vision (ICCV),* pp. 127-134, 2001.

[3] D. Aliaga, T. Funkhouser, D. Yanovsky, I. Carlbom, "Sea of Images", IEEE Visualization, 2002.

[4] D. Avis, H. Imai, "Locating a Robot with Angle Measurements", *Journal of Symbolic Computation*, No. 10, pp. 311-326, 1990.

[5] R. Azuma, "A Survey of Augmented Reality", *Presence: Teleoperators and Virtual Environments*, 6(4), pp. 355-385, 1997.

[6] M. Betke, L. Gurvits, "Mobile Robot Localization Using Landmarks", *IEEE Transactions on Robotics and Automation*, Vol. 13, No. 2, pp. 251-263, 1997.

[7] J. Borenstein, B. Everett, L. Feng, *Navigating Mobile Robots: Systems and Techniques*. A.K. Peters, Ltd., Wellesley, MA, 1996.

[8] T. Boult, Remote Reality Demonstration, *Proceedings of IEEE Computer Vision and Pattern Recognition (CVPR),* pp. 966-967, 1998.

[9] R. Chatila, J.P. Laumond, "Position Referencing and Consistent World Modeling for Mobile Robots", *Proceedings of IEEE Intl. Conference on Robotics and Automation*, pp. 138-145, 1985.

[10] I. Cox, "Blanche: Position Estimation for an Autonomous Robot Vehicle", *Proceedings of IEEE Int. Workshop on Intelligent Robots and Systems,* pp. 432-439, 1989.

[11] F. Dalleart, W. Burgard, D. Fox, S. Thrun, "Using the Condensation Algorithm for Robust, Vision-based Mobile Robot Localization", *Proceedings of IEEE Computer Vision and Pattern Recognition (CVPR),* pp. 588-594, 1999.

[12] U. Dhond, J. Aggarwal, "Structure from Stereo – a Review", *IEEE Transactions on Systems, Man, and Cybernetics,* Vol. 19, No. 16, 1989.

[13] C. Geyer and K. Daniilidis, "Catadioptric Camera calibration", *Proceedings Int. Conf. on Computer Vision (ICCV)*, pp. 398-404, 1999.

[14] C. Geyer and K. Daniilidis, "Structure and Motion from Uncalibrated Catadioptric Views", *IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2001.

[15] Gortler S., Grzeszczuk R., Szeliski R., and Cohen M., "The Lumigraph", *Computer Graphics (SIGGRAPH 96)*, pp. 43-54, 1996.

[16] S.B. Kang, R. Szeliski, "3D Scene Data Recovery using Omnidirectional Multibaseline Stereo", *IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, pp. 364-370, 1996.

[17] S.B. Kang, "Catadioptric Self-Calibration", *Proceedings of IEEE Computer Vision and Pattern Recognition (CVPR)*, pp. 201-207, June 2000.

[18] J.J. Leonard, H.F. Durrant-Whyte, “Mobile Robot Localization by Tracking Geometric Beacons”, *IEEE Transactions on Robotics and Automation,* Vol. 7, No. 3, pp. 376-382, 1991.

[19] Levoy M. and Hanrahan P., “Light Field Rendering”, *Computer Graphics (SIGGRAPH 96)*, pp. 31-42, 1996.

[20] McMillan L. and Bishop G., “Plenoptic Modeling: An Image-Based Rendering System”, *Computer Graphics (SIGGRAPH 95)*, pp. 39-46, 1995.

[21] J. Mendelsohn, K. Daniilidis, “Constrained Self-Calibration”, *Proceedings of IEEE Computer Vision and Pattern Recognition (CVPR),* pp. 581-587, 1999.

[22] T. Morita, T. Kanade, “A Sequential Factorization Method for Recovering Shape and Motion from Image Streams”, *Proc. ARPA Image Understanding Workshop*, Vol. 2, pp. 1177–1188, 1994.

[23] S. Nayar, “Catadioptric Omnidirectional Camera”, *Proc. of IEEE Computer Vision and Pattern Recognition*, pp. 482-488, 1997.

[24] J. O'Rourke, *Art Gallery Theorems and Algorithms*, Oxford University Press, New York, 1987.

[25] M. Pollefeys, R. Koch, and L. van Gool, “Self-Calibration and Metric Reconstruction in Spite of Varying and Unknown Internal Camera Parameters”, *Proceedings Int. Conf. on Computer Vision (ICCV),* pp. 90-95, 1998.

[26] K. Simsarian, T. Olson, N. Nandhakumar, “View-Invariant Regions and Mobile Robot Self-Localization”, *IEEE Transactions on Robotics and Automation,* Vol. 12, No. 5, pp. 810-816, 1996.

[27] K. Sugihara, “Some Location Problems for Robot Navigation using a Single Camera”, *Computer Vision, Graphics, and Image Processing*, Vol. 42, pp. 112-129, 1988.

[28] H. Shum, L. He, “Concentric Mosaics”, *Proceedings of ACM SIGGRAPH*, pp. 299-306, 1999.

[29] R. Talluri, J. K. Aggarwal, “Mobile Robot Self-Location Using Model-Image Feature Correspondence”, *IEEE Transactions on Robotics and Automation*, Vol. 12, No. 1, pp. 63-77, 1996.

[30] C. J. Taylor, “Video Plus**”,** *IEEE Workshop on Omnidirectional Vision*, pp. 3-10, 2000.

[31] S. Teller, M. Antone, Z. Bodnar, M. Bosse, S. Coorg, M. Jethwa, N. Master, “Calibrated, Registered Images of an Extended Urban Area", *IEEE Computer Vision and Pattern Recognition (CVPR)*, 2001.

[32], B. Triggs, P. McLauchlan, R. Hartley, A. Fitzgibbon", *Bundle Adjustment - A Modern Synthesis, in Vision Algorithms: Theory and Practice*, Springer-Verlag, 2000.

[33] M. Ward, R. Azuma, R. Bennett, S. Gottschalk, and H. Fuchs, "A Demonstrated Optical Tracker with Scalable Work Area for Head-Mounted Display Systems.", *ACM Symposium on Interactive 3D Graphics (I3D 92)*, pp. 43-52, 1992.

[34] Y. Yagi, Y. Nishizawa, M. Yachida, “Map-Based Navigation for a Mobile Robot with Omnidirectional Image Sensor COPIS”, *IEEE Transactions on Robotics and Automation,* Vol. 11, No. 5, pp. 634-648, 1995.

APPENDIX A

In this appendix, we briefly describe an optional method for improving the estimate of the 3D fiducial locations before tracking and image capture. Using a tape measure, we measure the distances between pairs of mutually visible fiducials and fit the resulting rigid configuration to the original floor plan. So long as the graph of mutual fiducial visibility is at least biconnected, the distances alone provide a rigid fiducial configuration. To fully support floor plans containing loops, a sufficient condition is that the fiducial visibility graph must be biconnected using only local edges (i.e., if an edge is removed, the affected vertices must remain connected using only local edges and

not via a sequence of edges on the other side of the loop). To enforce this, we could modify the original planning algorithm so that when searching for the most redundant fiducial to remove, we also verify that the resulting mutual fiducial visibility graph is still biconnected.

Using the N fiducial locations $f_1=(x_1, y_1)$ through $f_N=(x_N, y_N)$ computed by the planning algorithm and given a set of fiducial distance measurements, we create a distance matrix. Then, using minimization, we obtain a *new* set of fiducial locations $f_1'=(x_1', y_1')$ through $f_N'=(x_N', y_N')$ that produce a distance matrix with the same values as the measured distances. We wish to minimize the function $g_{placement}$ consisting of the sum of the squared differences between the measured distances m_{ij} and the corresponding distances of the current fiducial set. (We use the Cronecker delta to ignore terms for which we have no distance measurements.) The function and its partial derivatives are given below:

$$g_{placement}(x_1', y_1', \ldots, x_N', y_N') = \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} K_{ij} E_{ij}^2 \qquad (3)$$

$$\begin{aligned}
\frac{\partial g}{x_i'} &= \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} 2K_{ij}E_{ij}(-2x_i' + 2x_j') \\
\frac{\partial g}{y_i'} &= \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} 2K_{ij}E_{ij}(-2y_i' + 2y_j') \qquad (4) \\
E_{ij} &= (m_{ij}^2 - ((x_i' - x_j')^2 + (y_i' - y_j')^2)) \\
K_{ij} &= (1 - \delta_{m_{ij},0})
\end{aligned}$$

We fit the resulting rigid fiducial configuration to the original ideal fiducial configuration by finding a transformation that best aligns the two. We compute a translation (t_x, t_y) and rotation r of the configuration that minimizes the sum of the squared distances between the fiducial locations in the floor plan and the fiducials of the actual configuration. The function g_{fit} to minimize is given below:

$$g_{fit}(t_x, t_y, \omega) = \sum_{i=1}^{N} (x_i'(t_x, t_y, r) - x_i)^2 + (y_i'(t_x, t_y, r) - y_i)^2 \quad (5)$$