

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

Daniel G. Aliaga  
aliaga@cs.princeton.edu  
Princeton University

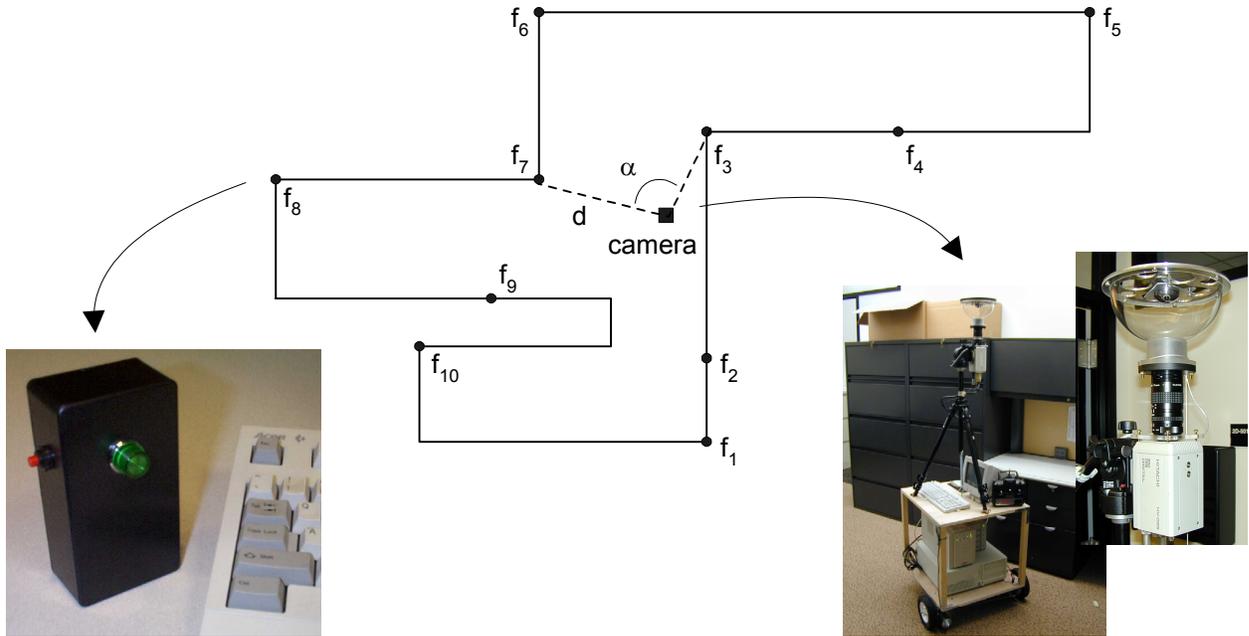
Ingrid Carlbom  
carlbom@lucent.com  
Lucent Bell Labs

## 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 (FOV) cameras, but few algorithms have been proposed to take advantage of the larger FOV of panoramic cameras. Furthermore, while existing camera pose estimation algorithms work well in small spaces, they do not scale well to large, complex 3D spaces 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. Image-based rendering techniques require camera position and orientation to recreate novel views of an environment from a large number of images. Augmented reality systems use camera pose information to align virtual objects with real objects, and robot navigation relies on knowledge of the robot's current location in order to plan a path through 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. To obtain robust and accurate pose estimates, the model or map must contain sufficient detail to ensure correspondences in all frames. Another class of passive methods, self-tracking methods, use optical flow to calculate changes in position and orientation. However, self-tracking approaches are prone to cumulative errors making them particularly unsuited for large environments.



**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.

Active methods utilize 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. For large arbitrarily shaped environments, such as the ones presented in this 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). Camera pose estimates can be

computed in real-time or after image capture. For the latter, we also describe a global optimization procedure to obtain improved estimates as a post process. For example, this is useful to capture a large set of images annotated with camera pose.

Our approach is particularly suited for panoramic cameras. Since our fiducials can be viewed from any direction, a panoramic camera can easily see all unoccluded fiducials surrounding the camera. This improves the reliability of pose estimation by increasing the number of fiducials that can be seen and simplifies fiducial planning by allowing the method to ignore camera orientation.

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 all viewpoints within the environment. Exact fiducial locations are not necessary and will be obtained later via the optimization method. At the expense of more fiducials, enforcing tighter constraints increases pose estimation accuracy. Our system has a brief 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.

## 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 in order to obtain sensor measurements.

Structure from motion techniques [1, 2] track environment features during an image sequence and 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 [3] describes a system for computing pose using panoramic images. The user selects points and edges as features in a number of keyframes. 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 [4] 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 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, 3<sup>rd</sup> Tech Inc. developed a commercial version of the UNC Ceiling Tracking project, 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, hardware trackers require complex installations and are very invasive if they are going to yield high precision results.

## FIDUCIAL PLANNING

### 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 [5]. 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) to be viewed by a panoramic camera (instead of an observer). 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  $\alpha$  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.

### 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 all polygons meet the visibility constraint, fiducials are added along polygon edges until at least  $V$  fiducials 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. This best-first approach converges to a set of fiducials that satisfy the visibility constraints.

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 constraints. We calculate the redundancy value for each fiducial 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,  $\alpha_{min}$ , differs from  $A$ , using weights  $w_v$ ,  $w_d$ , and  $w_\alpha$ , respectively:

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

We iteratively remove the fiducial with the next highest redundancy value as long as the constraints are still satisfied. 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  $D$ . For each convex region of viewpoints, we check the constraints from a dense sampling of viewpoints along the perimeter and exploit the following 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.

## 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. In this section, we describe our tracking algorithm, pose estimation method, and global optimization.

### 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 predicted fiducial position defines the location of a search window. Within each search window, the fiducial projection is selected from among all approximately circular blobs of pixels that exceed an intensity threshold.

### 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 tracked fiducials is  $T$ , then we obtain at most  $R = \binom{T}{2}$  valid fiducial pairs and estimates. In the real-time system, we use a simple weighted averaging scheme. Other schemes are also possible [6] but are difficult to scale to thousands of images seeing the same fiducials (as is the case with our test scenarios).

To calculate the camera position and orientation  $(x_{ij}, y_{ij}, \omega_{ij})$  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  $d_j$ , which are the distances from the between the camera’s focal point and the projection of the fiducial positions onto the horizontal plane passing through the camera’s focal point. Then, we compute the angle  $\alpha_{ij}$  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.

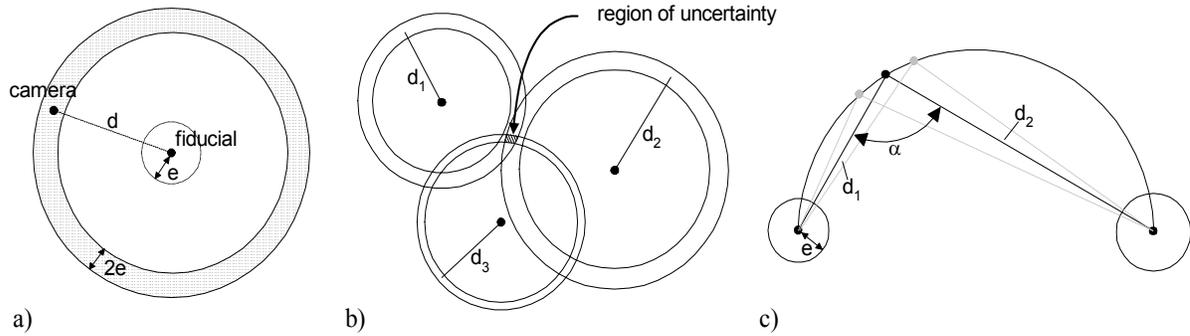
## Global Optimization

To determine a globally consistent set of camera poses and 3D fiducial locations, we use bundle adjustment, 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 fiducial estimates. 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 [7]. 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  $\delta_{ik}$  is 1 when fiducial  $i$  is tracked on image  $k$ ):

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

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



**Figure 2. 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  $2e$ . (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.

## ERROR MODEL

We develop an error model in order to bound the uncertainty of the pose estimates computed by our algorithm. Given a particular fiducial configuration 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 [8].

Figures 2(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 2a). For multiple fiducials and distance estimates, the region of uncertainty can be constructed by intersecting the annuli surrounding each fiducial (Figure 2b).

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 distance estimates and the observed angle. For example, as shown in Figure 2c, an observed angle of  $\alpha=90$  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. 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.

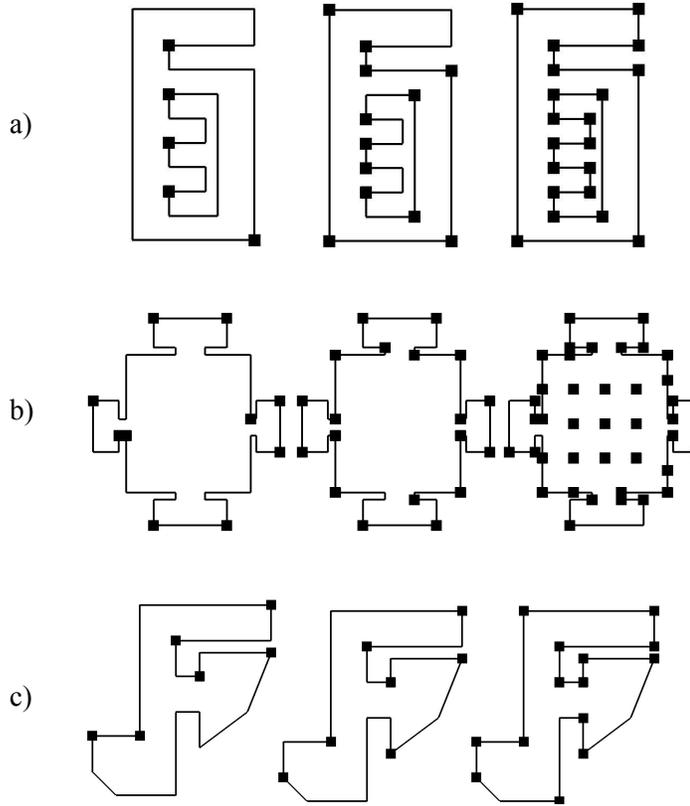
## IMPLEMENTATION

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). 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 [7, 8] 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.

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 where many are visible. For the other

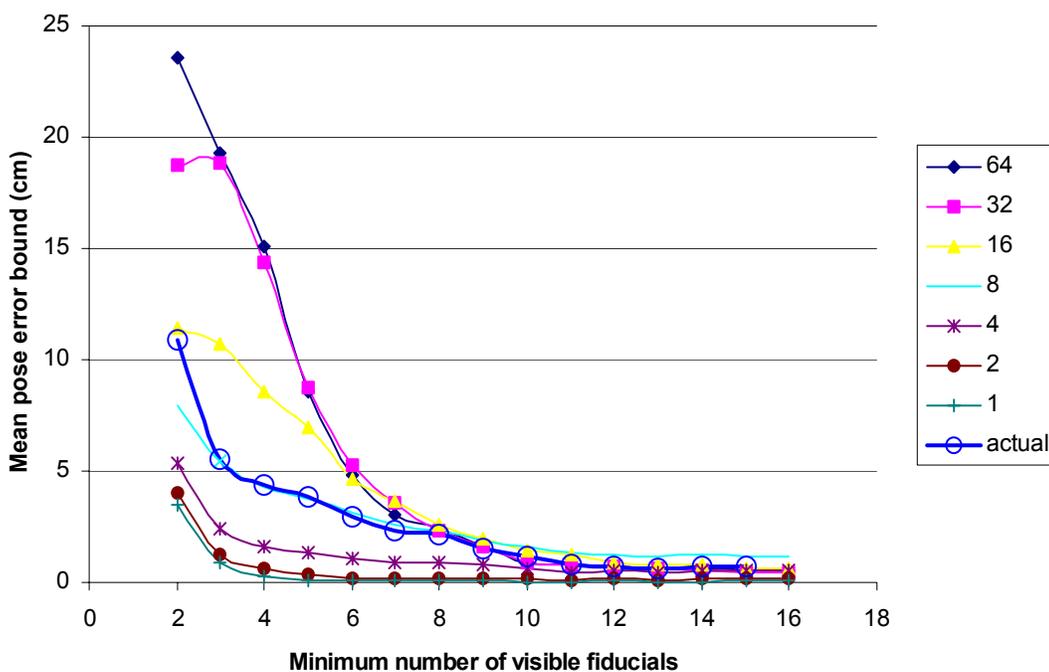


**Figure 3. Fiducial Constraints.** We show fiducial sets for different fiducial placement constraints. a) Visibility constraint: left:  $V=1$ , middle:  $V=2$ , right:  $V=4$  (other placement constraints remain constant at  $D=\infty$  and  $A=0$  degrees). b) Distance constraint: left:  $D=\infty$ , 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  $A=0$ ). c) Angle constraint: left:  $A=0$  degrees, middle:  $A=60$  degrees, right:  $A=120$  degrees (other placement constraints remain constant at  $D=\infty$  and  $V=2$ ).

weights,  $w_d$  and  $w_\alpha$ , we choose values that, on average, scale the corresponding terms to approximately equal values thus equalizing the tradeoff of distance and angle.

## RESULTS

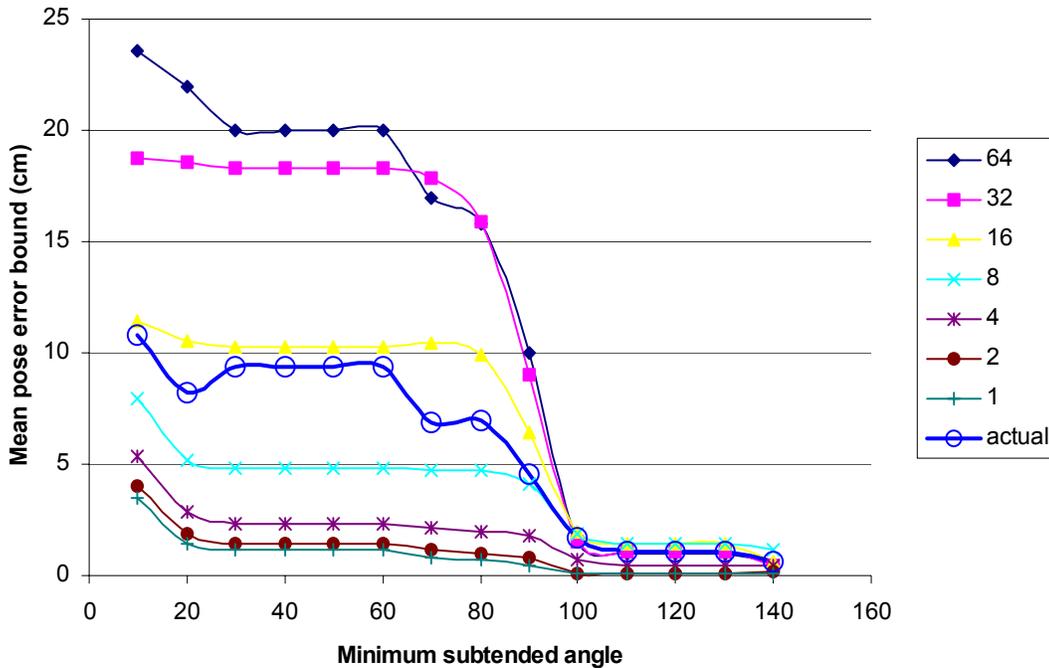
We have used our approach to generate fiducial positions and camera pose for several environments. Computing fiducial positions is relatively fast. On a 1 GHz PC it takes, on average, less than a minute to compute fiducial sets for the environments shown in this article. The floor plans for the test environments



**Figure 4. 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.

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.

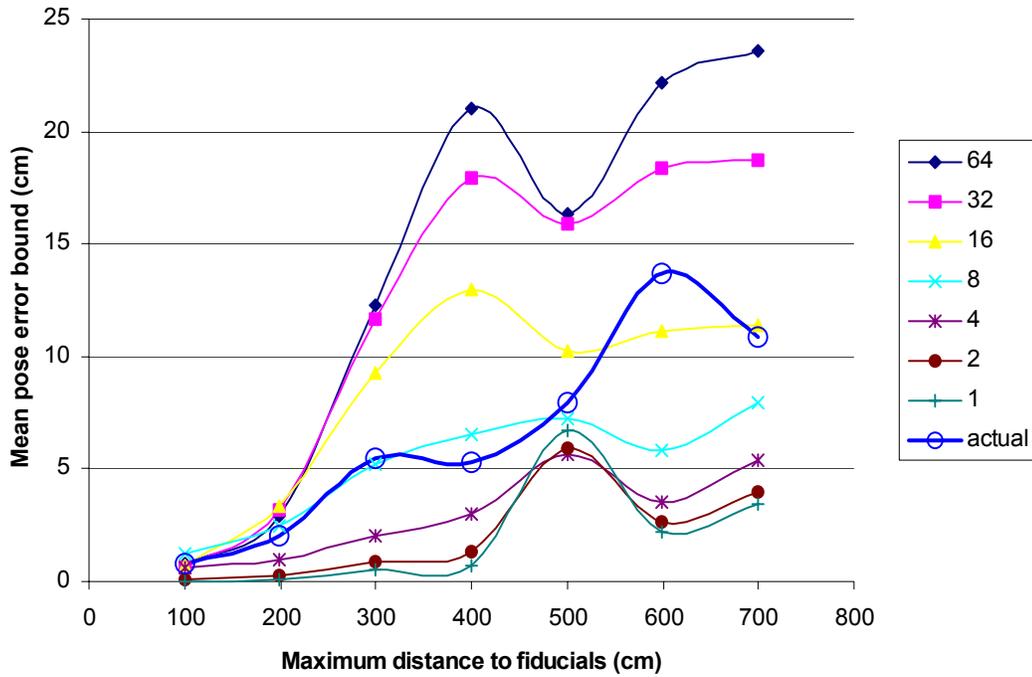
Figure 3 demonstrates the fiducial planning algorithm in three example environments. In the first environment (Figure 3a), the fiducial placement guarantees the minimum number of visible fiducials  $V$  is 1, 2 or 4, respectively. The solution for  $V=1$  corresponds to a set of guard positions for the classical art-gallery problem. In the second environment (Figure 3b), the fiducial placement guarantees at least two fiducials are always visible and the 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 3c), the fiducial placement guarantees for any



**Figure 5. 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.

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.

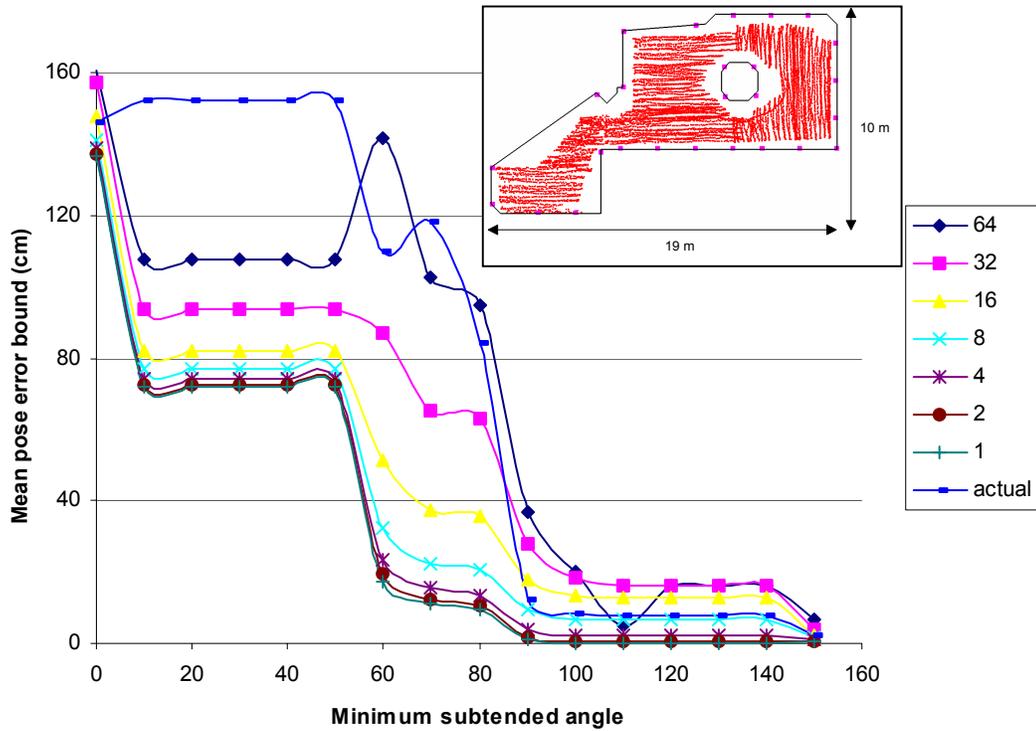
To better understand how pose estimation is affected by the number of visible fiducials, the angle subtended by pairs of fiducials, and the distance to the fiducials, we plot pose error bounds for several fiducial sets in a single test environment. The test environment measures 7 by 10 meters in size and contains several interconnected spaces (Figure 1). 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 6. 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.

Although there is interdependency between the different fiducial constraints, we plot each constraint separately to provide insights into the subtle relationships among the fiducial constraints and to calculate “sweet spots” in the tradeoff of fiducials and accuracy. Figures 4-6 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. For example, the curves for “8” represent the pose error bound assuming that we know the 3D position of the fiducials to within 8 cms of accuracy.

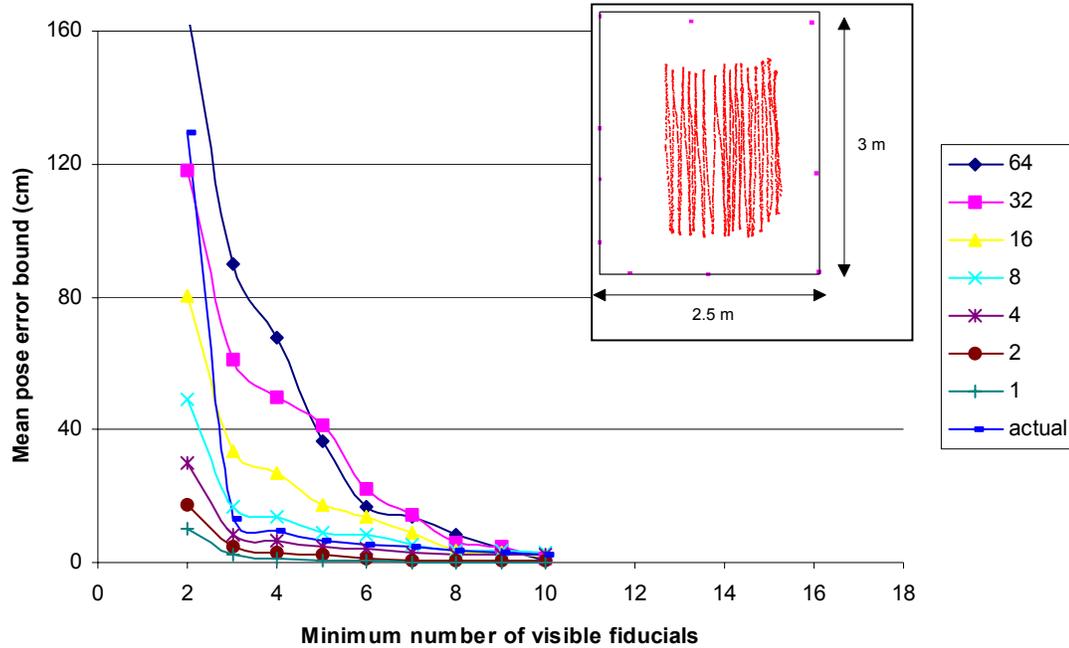
Figure 4 depicts the effect of varying the minimum number of visible fiducials. For instance, in order to obtain less than 1 cm of mean pose estimation error with as little as 5 visible fiducials, we need to compute



**Figure 7. Pose computations in a museum environment.** 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). The graph shows the pose error bounds for a range of values for the minimum subtended angle constraint.

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 5 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 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 can further improve pose estimates. Nevertheless, the configuration of two fiducials and a subtended angle near 180 degrees can yield unstable results.



**Figure 8. Pose computations in a small office environment.** 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). The graph shows the pose error bounds for the image sequence using subsets of the fiducials.

Figure 6 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 4-6, 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.

We have used our method to capture several large image databases that have successfully been used to recreate 3D environments [9, 10]. Figures 7 and 8 report the pose estimation bounds for two of these datasets. Figure 7 is a museum measuring 10 by 19 meters in size and with a sequence of 9832 panoramic images. Figure 8 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.

## **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.

For future work, we would like to perform a theoretical analysis of the convergence of pose estimation. This will allow us to gauge how close we are to the optimum solution and allow us to determine the error bottlenecks in the overall system.

In addition, we are interested in extending our system to support self-guided capture. Using computer-controlled motors, we could navigate 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.

## **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] C. Geyer, K. Daniilidis, “Structure and Motion from Uncalibrated Catadioptric Views”, *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 279-286, 2001.
- [2] M. Pollefeys, R. Koch, and L. van Gool, “Self-Calibration and Metric Reconstruction in Spite of Varying and Unknown Internal Camera Parameters”, *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, pp. 90-95, 1998.
- [3] C. J. Taylor, “Video Plus”, *IEEE Workshop on Omnidirectional Vision*, pp. 3-10, 2000.
- [4] S. Teller, M. Antone, Z. Bodnar, M. Bosse, S. Coorg, M. Jethwa, N. Master, “Calibrated, Registered Images of an Extended Urban Area”, *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 813-820, 2001.
- [5] J. O'Rourke, *Art Gallery Theorems and Algorithms*, Oxford University Press, New York, 1987.
- [6] P. Sturm, “Mixing Catadioptric and Perspective Cameras”, *Proceedings of IEEE Workshop on Omnidirectional Vision*, pp. 37-44, 2002.
- [7] S. Nayar, “Catadioptric Omnidirectional Camera”, *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 482-488, 1997.

- [8] D. Aliaga, "Accurate Catadioptric Calibration for Real-time Pose Estimation in Room-size Environments", *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, pp. 127-134, 2001.
- [9] D. Aliaga, I. Carlbom, "Plenoptic Stitching: A Scalable Method for Reconstructing Interactive Walkthroughs", *Proceedings of ACM SIGGRAPH 2001*, pp. 443-450, 2001.
- [10] D. Aliaga, T. Funkhouser, D. Yanovsky, I. Carlbom, "Sea of Images", *Proceedings of IEEE Visualization*, pp. 331-338, 2001.