J Intell Robot Syst (2007) 50:257–295
DOI 10.1007/s10846-007-9164-7

Optimal Camera Placement for Automated Surveillance Tasks

Robert Bodor · Andrew Drenner · Paul Schrater · Nikolaos Papanikolopoulos

Received: 4 February 2007 / Accepted: 4 June 2007 /
Published online: 3 October 2007

Abstract Camera placement has an enormous impact on the performance of vision systems, but the best placement to maximize performance depends on the purpose of the system. As a result, this paper focuses largely on the problem of task-specific camera placement. We propose a new camera placement method that optimizes views to provide the highest resolution images of objects and motions in the scene that are critical for the performance of some specified task (e.g. motion recognition, visual metrology, part identification, etc.). A general analytical formulation of the observation problem is developed in terms of motion statistics of a scene and resolution of observed actions resulting in an aggregate observability measure. The goal of this system is to optimize across multiple cameras the aggregate observability of the set of actions performed in a defined area. The method considers dynamic and unpredictable environments, where the subject of interest changes in time. It does not attempt to measure or reconstruct surfaces or objects, and does not use an internal model of the subjects for reference. As a result, this method differs significantly in its core formulation from camera placement solutions applied to problems such as inspection, reconstruction or the Art Gallery class of problems. We present tests of the system's optimized camera placement solutions using real-world data in both

R. Bodor · A. Drenner · P. Schrater · N. Papanikolopoulos (✉)
Department of Computer Science and Engineering,
University of Minnesota, Minneapolis,
MN 55455, USA
e-mail: npapas@cs.umn.edu

R. Bodor
e-mail: rbodor@cs.umn.edu

A. Drenner
e-mail: drenner@cs.umn.edu

P. Schrater
e-mail: schrater@cs.umn.edu

indoor and outdoor situations and robot-based experimentation using an all terrain robot vehicle-Jr robot in an indoor setting.

Keywords Camera networks · Robot/camera placement · Observability · Optimization · Sensor networks · Vision-based robotics

1 Introduction

This paper focuses on the problem of task-specific camera placement. We attempt to determine how to place cameras relative to activities being performed by human subjects, in order to provide image input to a system such that it optimizes that system's ability to achieve its task (learn activities, recognize motion, take measurements, etc.). We use information taken from passive measurements of the scene, and in general, do not assume a model of the environment.

The performance of computer vision systems for measurement, automation applications, surveillance, reconstruction, gait recognition, and many other applications, depends heavily on the placement of cameras observing the scene. A standard approach to camera placement for observing a scene is to position the cameras uniformly around the area. This approach tends to do a good job of providing view coverage of an area. However, this is not appropriate for our needs, because such uniform placement ignores the human aspect of the tasks we are interested in. We use the statistics of the motion paths taken in the scene to determine proper camera placement, rather than spread cameras evenly to cover the area, since it is the people, not the area that we are interested in observing. Our goal is to optimize the aggregate observability of the tasks being performed by the subjects in an area. We develop a general analytical formulation of the observation problem, in terms of the statistics of the motion in the scene and the total resolution of the observed actions, that is applicable to many observation tasks and to multiple camera systems. An optimization approach is used to find the internal and external (mounting position and orientation) camera parameters that optimize the observation criteria. We demonstrate the method for multiple camera systems in real-world monitoring applications, both indoor and outdoor.

The paper is organized as follows: Section 2 describes related work, while Section 3 describes our camera placement formulation and experimental results. Section 4 summarizes our conclusions, and Section 5 describes future work we would like to pursue.

2 Related Work

The use of computer vision to automate real-world surveillance and motion tasks is a very rich and interesting field. Some of the application areas include pedestrian and transportation safety [5, 24, 28], surveillance [3, 14, 19, 25, 38], crime prevention [16], and human-computer interaction [12, 51].

An issue that plays a fundamental role in real-world surveillance and motion recognition is that of optimal camera placement for the purpose of maximizing the observability of the motions taking place. Optimal camera placement has been

considered for numerous problem areas, and significant prior literature exists in the areas of scene monitoring, surveillance and tracking, part inspection, three-dimensional reconstruction, robotic sensor motion planning (often for eye-in-hand systems), and for the Art Gallery problem. While all of the methods developed in these areas focus on the problem of optimal camera placement, the methods differ in features of interest, and the objective functions which they optimize. Furthermore, we believe the camera placement method described in this paper is novel and distinct from prior approaches in these areas.

The method focuses on optimizing the view of a set of target motion paths taken through an area of interest. As such, the method considers dynamic and unpredictable environments, where the subject of interest changes in time. Furthermore, the path starting positions and motion directions are not known a priori since targets can come and go within the area of interest. The method focuses on optimizing the aggregate path observability as a whole, based on a probabilistic framework, and does not attempt to optimize the view of any individual path or target motion. While tracking is used as one means of providing path input to the system, the purpose of the method is not tracking, but rather observing the path motions for subsequent applications such as motion recognition. The method does not attempt to measure or reconstruct surfaces or objects, and does not use an internal model of the subjects for reference. As a result of this and the issues described above, this method differs significantly in its core formulation from camera placement solutions applied to problems such as inspection, reconstruction or the Art Gallery class of problems. In the rest of this section, we describe many instances of prior work in these areas, and attempt to elaborate on these distinctions.

O'Rourke [35] provides an in-depth theoretical analysis of the problem of maximizing camera coverage of an area, where the camera fields of view do not overlap (the so-called "Art Gallery" problem).

In recent years, research has been done to extend the framework to include limited field of view cameras and to incorporate resolution metrics into the formulation. Fleishman et al. further refined the art gallery framework by introducing a resolution quality metric [15]. In addition, Isler et al. extended the formulation of the minimum guard coverage art gallery problem to incorporate minimum-set cover. They derived reduced upper bounds for two cases of exterior visibility for two- and three-dimensions [20].

Our method differs from the art gallery framework in several important ways. We focus on task observability, and try to capture target motions, while this framework tries to optimize coverage area. This corresponds to minimizing overlapping views. We do not optimize for minimum view overlap, and in fact, to accomplish these tasks, overlapping camera views are often necessary. However, the work we do can be related to the notion of "strong visibility of guard patrol" as described by O'Rourke. In some ways, this is the dual problem - assuming moving cameras, and fixed points of interest to observe. We assume (largely) fixed cameras, with moving subjects. However, the framework of the art gallery problem makes many fundamental assumptions that differ considerably from our own framework. The art gallery framework assumes that the subject(s) of observation (in O'Rourke's case, the gallery walls) is/are known a priori. They generally also assume omni-directional cameras, and that all non-occluded camera placements are equally good. These assumptions are fundamental to the art gallery problem, and so fundamentally

affect the approach, that they make it a different problem. To our knowledge, no one has addressed these issues as we do.

In the field of robotics, vision sensor planning has been studied to aid in inspection, object recognition, task planning and visual servoing tasks. Several excellent surveys of this area have been written, including Roy et al. and Scott et al. [39, 40], and Tarabanis et al. [43]. Abrams et al. develop a system to perform dynamic sensor planning for a camera mounted on a moving robotic arm in order to compute optimal viewpoints for a pre-planned robotic grasping task [1]. Nelson and Khosla [29] introduce a modified manipulability measure in order to reduce the constraints on the tracking region of eye-in-hand systems while avoiding singularities and joint limits. They also studied dynamical sensor placement within this context, and introduced the concept of the resolvability ellipsoid to direct camera motion in real-time in order to maintain servoing accuracy [30, 31]. Sharma and Hutchinson also introduce a quantitative sensory measure, perceptibility, in order to improve positioning and control of manipulator systems [41].

Tarabanis et al. [44, 45], and [46] present a succession of planning methods to determine optimal camera placement given task-specific observational requirements such as field of view, visibility, and depth of field. They describe optimization approaches to maximize these features for model based and task driven eye-in-hand sensor planning. In addition, Yao and Allen [52] formulate the problem of sensor placement to satisfy feature detectability constraints as an unconstrained optimization problem, and apply tree-annealing to compute optimal camera viewpoints in the presence of noise.

Olague and Mohr [34] consider the problem of optimal camera placement for 3D measurement accuracy of parts located at the center of view of several cameras. They demonstrate good results in simulation for known static objects. Chen and Davis [10], develop a resolution metric for camera placement considering occlusions. In addition, Denzler et al. [13] develop a Kalman filter based approach for selecting optimal intrinsic camera parameters for tracking applications. They demonstrate results for actively adapting focal length while tracking a rigid object. Tarbox and Gottschich [47] present three different planning algorithms for measuring exposed surfaces of an object using a measurability criterion for model based inspection. Chen and Li [9] develop a sensor placement graph for model based inspection, based on a genetic algorithm approach. They refine the graph to a shortest path using a Christofides algorithm.

Our method differs from these because it considers the joint observability of a set of tasks. In addition, our method considers task uncertainty: the locations of the tasks that we are attempting to observe are not known a priori, and change with time as the subjects move through the scene.

Another important set of multi-camera systems are those that incorporate only fixed cameras. These systems generally focus on the problem of tracking subjects across cameras and finding correspondence of subjects between cameras. Two good examples of this work appear in Ukita and Matsuyama [48] and Stauffer and Tieu [42]. These methods do not consider observability, and camera placement is generally ad-hoc. We believe that this group of methods could benefit from the camera placement framework we introduce.

Another interesting area of camera placement research is the area of object or scene reconstruction. Three-dimensional visual hull reconstruction is a process for

creating three-dimensional models of objects from multiple two-dimensional images taken from views around the object. This process has been developed extensively by the computer graphics field and has seen applications in many areas including motion picture special effects, video games and product marketing. Matusik et al. [23] describe a method where reconstruction is accomplished by projecting the silhouettes of each camera's foreground subject into a 3D space. Each silhouette "carves out" a section of the space, and the intersection of the carved-out sections results in the 3D model of the subject. Similar approaches to 3D reconstruction from silhouette can be found in Carranza et al. and Cheung et al. [8, 11], and Weik and Liedtke [49]. Another three-dimensional reconstruction approach that uses tensor voting is described by Mordohai and Medioni [27]. This method does not require computation of silhouettes, or a known background.

Reconstruction quality may be greatly improved by increasing the number of cameras used, or views taken. In addition, improvements to visual hull reconstruction have been developed in recent years [18, 32, 33, 50]. Many of these improve the reconstructed shape by adjusting the camera positioning based on an internal model of the subject. These approaches are very interesting, but differ from the method we describe because they assume a static object or scene, often incorporate a known model, and are formulated to optimize the appearance or minimize the error of the external surfaces of an object, not optimally observe a set of motion paths.

3 Camera Placement to Observe the Scene

Resolution is one of the fundamental and primary information bottlenecks for vision applications. The performance of virtually all single or multiple camera-based vision systems will improve with higher resolution input. For example, measurement, 3D reconstruction, and recognition accuracy all have a dependence on the resolution of the subject. Therefore, it is desirable to position cameras in order to maximize the total resolution of the subject in all cameras (as measured by the fraction of the subject's area in the image). However, how to position the cameras is not always obvious.

For example, if we desire to monitor an area effectively with a multi-camera vision system (Fig. 1), the system may be required to perform a variety of tasks such as general surveillance (detection of loitering behavior, monitoring of crowd movements, etc.), subject tracking [14], activity classification [2], gesture recognition [12], along with individual biometrics gathering (gait recognition, etc.). Each of these tasks may have different requirements and features of interest. As a result, optimal camera placement may vary from task to task. In addition, task location will vary throughout the area since people are free to move throughout the area of interest in any direction they like.

As a result, positioning a camera, or set of cameras, to effectively observe the area of interest is a difficult problem. Each camera position has to take into consideration the observability of all activities in order to optimize the system's performance.

Our goal is to address the problem of camera placement to optimize the aggregate observability of a set of tasks. One possible application of this research is the development of a design tool for surveillance camera placement in areas of high traffic, where each subject may take a different path through the area. This work assumes

Fig. 1 Scene of people moving outdoors

the cameras are statically mounted to view an area. Optimizing the observability of such a system means jointly maximizing the observability of the cameras relative to the expected path distribution for the area of interest.

3.1 Problem Formulation

3.1.1 General Formulation

The general problem is to optimize the aggregate observability of a distribution of target motions, where targets mean anything that we want to track: vehicles, pedestrians, body parts, objects, etc. We assume target motions can be described by the 3D motions of a set of b privileged points $\vec{x}_1(t), \ldots, \vec{x}_b(t)$.

Each camera's state can be parameterized in terms of an "action" that carries the camera parameters from default values to the current values (e.g., the rotation and translation between world and camera coordinates). For every trajectory and set of camera parameters, we will define a gain function that encodes the quality of the view. The problem of finding a good camera location for the set of trajectories can be formulated as a decision theory problem that tries to maximize the expected gain for all cameras, where the expectation is performed across trajectories. In general:

$$V(\vec{u}_1, \ldots, \vec{u}_n) = \int_{\vec{s} \in S} G(\vec{s}, \vec{u}_1, \ldots, \vec{u}_n)\, p(\vec{s})\, d\vec{s} \tag{1}$$

where $G(\vec{s}, \vec{u}_1, \ldots, \vec{u}_n)$ represents a gain function on trajectory states and camera parameters, $\vec{u}_i$ are the parameters for the i^{th} camera, $p(\vec{s})$ is the prior probability distribution on the trajectory states, and S is the set of states of the system (world) we are interested in. Our system assumes that $p(\vec{s})$ is known or can be computed. In fact, we determine the probability distribution of paths by tracking subjects as they move through the area.

Given a set of m sample paths with parameters $\vec{s}_j$, the value function can be approximated:

$$V(\vec{u}_1, \ldots, \vec{u}_n) = \sum_{j=1}^{m} G\left(\vec{s}_j, \vec{u}_1, \ldots, \vec{u}_n\right). \tag{2}$$

The choice of a parameterization $\vec{s}_j$ will depend on the application. In motion studies, $\vec{s}_j$ would describe the aspects of the motion that are relevant for the task. For example, in activity recognition involving multi-segmented rigid bodies connected at joints, $\vec{s}$ could be a time-series of joint angles. One means of recovering these from video is described in Bregler and Malik [7]. Another interesting method for measuring motion trajectories from video is developed by Pless [37]. He demonstrates the use of isomaps to extract motion trajectories from a wide variety of video sequences.

The value of a particular view of this motion will depend on how well the relevant (and distinguishing) aspects of the motion survive projection. For example, Mizoguchi and Sato [26] developed "space-time invariants" to encode arm and hand motions taken from video. State vectors of this type would be very appropriate for human-computer interfacing or sign language applications. Alternatively, Parameswaran and Chellappa present several human kinematic motion invariant representations for standard locomotive motion types [36].

In our case, we are concerned with path observability, so $\vec{s}_j$ defines a set of linear paths taken through an area. These are parameterized as follows:

$$\vec{s}_j = \left[\, \phi_j \; x_j \; y_j \; l_j \,\right]^T \tag{3}$$

where ϕ_j is the orientation of the path j, (x_j, y_j) defines the path center, and l_j is the length of the path. Path lengths are not normalized due to variations in individual walking patterns.

In addition, $\vec{u}_i$ defines a vector of actions to be considered. In our case, actions correspond to camera positions, thus $\vec{u}_i$ defines a vector of camera parameters to be optimized. In general, $\vec{u}_i$ may contain any number of variables such as camera focal length, position, orientation, etc.

3.1.2 Specific Formulation

Assumptions In this work, we focus on viewing the locomotive motion paths taken by subjects through a scene in order to best observe those subjects. This choice makes the assumption that the motion path direction captures the principle component of the activity or motion of interest. This assumption is valid for many activity recognition and measurement applications (tracking, gait recognition, articulated motion classification, etc.) because these activities can be found by fixing the dominant orientation of the activity. In fact, many methods in the literature rely on this fact to design their classifiers (PCA, etc.). However, this approach would not be the best choice for applications that benefit from optimizing the front view of the subject, such as face recognition.

In addition, we assume that it is sufficient to view each path from one side only. Because of the symmetry of the human body, we consider viewing a path from either side to have the same value. This is very reasonable for classification, but may not be a valid assumption for other applications such as 3D reconstruction.

We also make the assumption that all paths in our distribution are linear. In fact, we measure (track) all paths making no parametric assumptions, and then fit lines to the tracking data via a least squares approximation. For tracking data that has a high curvature, we split the data into a sequence of linear segments. Since this can be done to an arbitrarily low residual error, this method does generalize to arbitrary path distributions.

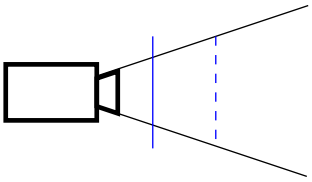


Fig. 2 When d_{ij} (*solid*) $< d_0$ (*dashed*), the camera is unable to observe the full motion sequence and fails the first constraint

One of our most fundamental assumptions lies in the use of the pinhole projection camera model. This model ignores lens distortion effects, among others. We chose this model for its simplicity and efficacy. This model captures the key impact of the camera positions correctly. In addition, for narrow field of view cameras that are far from the subject, lens distortion is minimal. Higher order terms may be added in the future to extend the formulation if this is necessary for a particular application.

Lastly, we use a simplified model of foreshortening. We focus on the first order effects, ignoring the higher order effects. Again, we believe that this captures the key effect of foreshortening on camera placement, and higher order models can be integrated into the formulation in the future if necessary.

Constraints The goal of camera placement for optimal path observability is to position a camera to observe the entire path of motion while maximizing the view of the subject in the image. The first part of that goal, observing the entire path of the j^{th} subject, by the i^{th} camera requires the distance from the camera to the subject (d_{ij}) to be far enough away from the subject that the entire motion is captured within the camera field of view (Fig. 2).

The camera must maintain a minimum distance, d_0, from each path to ensure that the full motion sequence is in view:

$$d_0 = \frac{r_a l_j f}{w} \tag{4}$$

where r_a is the aspect ratio of the image, w is the diagonal width of the imaging sensor, l_j is the length of the path, and f is the focal length of the lens.

In three dimensions, this constraint also requires that the path lies within the view frustum of the camera (Fig. 3) projected on the ground plane. This results in four linear constraints per camera that must be satisfied for a path to contribute to the observability metric for that camera position.

The second part of the goal, maximizing the view of the subject, requires the camera to be as close to the subject as possible, so that the subject is as large as possible in the image. Figure 4 depicts the reason for this. For a perspective

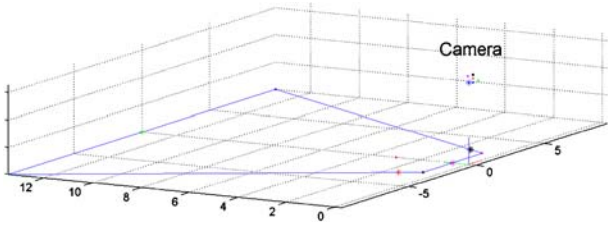


Fig. 3 View constraints for each camera. The view frustum of the camera projected on the ground plane. The path observed must fall within this frustum and remain at least d_0 from the camera

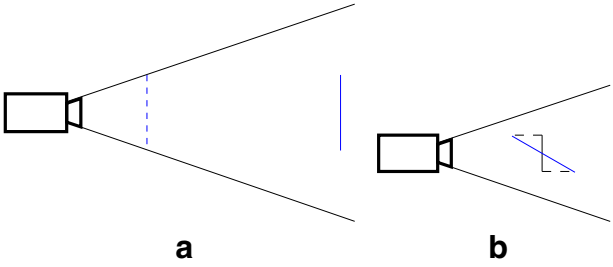


Fig. 4 Configurations that decrease observability in pinhole projection cameras. **a** Increasing object–camera distance. **b** Increasing foreshortening

projection camera with a fixed field of view, the size of an object in an image decreases as the distance to the object increases. In digital imaging, the area of an object in an image corresponds to a number of pixels that measure the object. As a result, we can define observability metrics directly in terms of pixel resolution.

Another factor that reduces observability of an object is foreshortening. Figure 4 depicts this situation. As the angle between the camera's view direction and the object decreases, the projection of the object on the camera's image plane also decreases. As a result, the number of pixels that measure the object decreases, and the observability lessens.

We use the following first-order approximations for these effects:

$$resolution \propto \frac{1}{d^2} \tag{5}$$

$$resolution \propto cos\,(\theta)\,. \tag{6}$$

Thus, optimizing path observability for an individual path corresponds to minimizing the distance between the camera and the path center, along with minimizing foreshortening effects. In this case, there are two sources of foreshortening: the angle between the path normal and the camera position, and the angle between the path center and the normal to the image plane (Fig. 5).

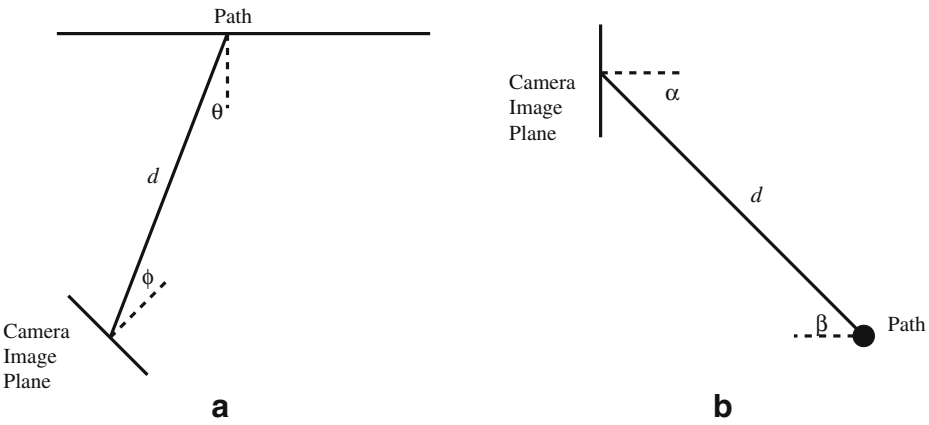


Fig. 5 Variables to minimize for each path. **a** Top view. **b** Side view

Objective function Based on this geometry, we can define an objective function (7) for each path–camera pair:

$$G_{ij} = \begin{cases} 0 & \text{if } d_{ij}^2 < d_0^2, \\ \frac{d_0^2}{d_{ij}^2} cos\left(\theta_{ij}\right) cos\left(\phi_{ij}\right) cos\left(\alpha_{ij}\right) cos\left(\beta_{ij}\right) & \text{otherwise.} \end{cases} \tag{7}$$

Optimizing this function over camera parameters will solve the observability problem for the single camera, single path case. This formulation forces the relationship $0 \leq G_{ij} \leq 1$ to hold, which will be required for our analysis later.

Single camera This single camera method may be further extended to work with multiple paths by simply optimizing over the aggregate observability (Eq. 8) of the entire path distribution.

$$V = \sum_{j=1}^{m} G_j. \tag{8}$$

This gives equal weighting to all paths in the distribution, and ensures that the single camera is placed to maximize the overall path distribution, or equivalently, to optimize the average path observability. Note that this is a unit-less metric. If V is multiplied by the height and width of the image in pixels, it becomes a pixel resolution metric of observability, as discussed above.

Multiple camera systems In the case of multiple camera systems, the formulation is somewhat more complicated. In these instances, we would like to ensure that the camera placements are optimized jointly for the multiple camera system.

To ensure that a single joint optimum is found, it would be necessary to search jointly over all camera parameters $\vec{u}_i$ at the same time. However, this method is very computationally intensive. In fact, the complexity of this method is exponential in the number of cameras:

$$O\left(V\right) = O\left((km)^n\right) \tag{9}$$

where m represents the number of paths, n represents the number of cameras, and k represents the number of camera parameters searched over per camera. (The value of k depends on the particular optimization method used. For example, in a uniform grid-based search method, $k = p^d$, where p is the number of samples per parameter, and d is the number of parameters. See Section 3.2 for more details.) For systems of many cameras, this approach quickly becomes infeasible.

As a result, we employ an iterative method with a computational complexity that is only linear in the number of cameras:

$$O\left(V\right) = O\left(kmn\right). \tag{10}$$

This method is clearly much faster than the previous approach. In addition, an iterative approach allows for the addition of new cameras without re-optimizing, which adds further to its utility.

In addition, we found that the iterative approach results in solutions that closely approximate the optimal approach for cases where the maxima of the objective function are sufficiently separated from one another (Fig. 10). Separated maxima

correspond to path clusters within the overall path distribution that are grouped by position and/or orientation. Such clusters occur naturally for most of the path distributions we observed because of constraints or affordances in the environment. (Sidewalks create clusters because traffic tends to move along them, doorways cause clusters because paths funnel into them, obstacles cause clusters by forcing traffic around them, etc.) If the clusters are sufficiently separated either in position or orientation, then any camera placement solution that observes one cluster will likely have a very low observability of the other clusters, and they can be treated independently.

Because of this finding, we developed an iterative method for camera placement that provides very good results in practice. In this formulation, we define a vector of path observabilities per camera where each element, G_{ij}, describes the observability of path j by camera i.

Now, for each camera i, the objective function becomes:

$$V_i = \sum_{j=1}^{m} \left[G_{ij}(\vec{u}_i) \prod_{k=1}^{i-1} \left(1 - G_{kj}(\vec{u}_k)\right) \right] \tag{11}$$

By inverting the observability values of the previous camera $\left(1 - G_{kj}\right)$, the current camera is directed to the regions of the path distribution that have the lowest observability so far. In other words, a second camera should be positioned to view those path clusters that the first camera did not view well, and so on.

$$V = \sum_{i=1}^{n} \sum_{j=1}^{m} \left[G_{ij}(\vec{u}_i) \prod_{k=1}^{i-1} \left(1 - G_{kj}(\vec{u}_k)\right) \right] \tag{12}$$

Equation 12 gives the total observability value function for the multi-camera system. Maximizing V optimizes the expected value of the observability, and thus provides the best results for the entire path distribution as a whole (assuming distinct clusters). It should be noted, however, that for particular individual paths, it may yield sub-optimal observability results, as this was not our goal. Figure 6a shows a plot of Eq. 11 for a sample path configuration, and Fig. 6b shows a plot of Eq. 12.

We can see from this that the observability asymptotically approaches the maximum as the number of cameras viewing the path distribution increases. The number of cameras necessary depends on the configuration of the particular path distribution, and we do not know it a priori. However, these results indicate that a sufficiency condition on the number of cameras necessary to completely observe any path distribution (to a given residual) may be determined via this approach. In addition, the experimental results shown in Sections 3.3 and 3.4 demonstrate that in practice, the iterative method is able to consistently capture all of the path observability with relatively few cameras.

To illustrate these equations, we develop V for 3 cameras, considering each path as a separate cluster:

$$V_1 = \sum_{j=1}^{m} G_{1j} \tag{13}$$

For convenience, we drop the $\vec{u}_i$ notation, and $G_{ij}(\vec{u}_i)$ becomes G_{ij}.

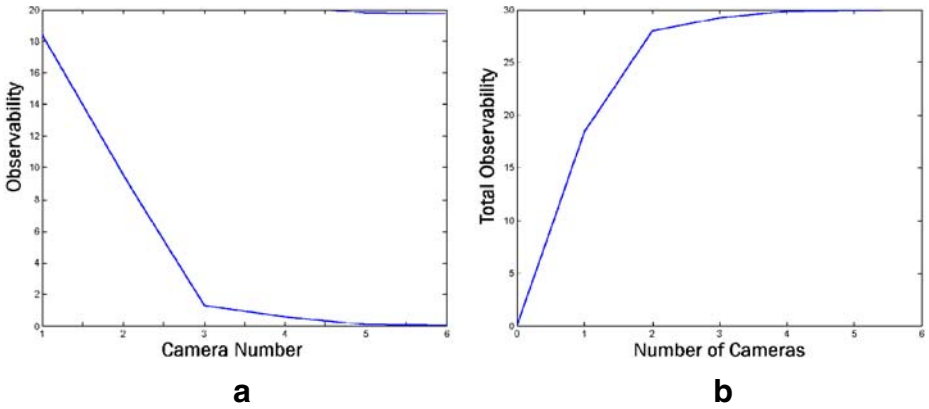


Fig. 6 Observability as a function of cameras placed. **a** Observability per camera V_i. **b** Total observability V vs. number of cameras

Thus,

$$V_1 = G_{11} + \cdots + G_{1m}. \tag{14}$$

For the second camera, V_2 can be determined as follows:

$$V_2 = \sum_{j=1}^{m} \left[G_{2j}(\vec{u}_2) \left(1 - G_{1j}(\vec{u}_1)\right) \right]$$

$$V_2 = G_{21} + \cdots + G_{2n} - G_{11}G_{21} \cdots - G_{1m}G_{2m}. \tag{15}$$

Similarly,

$$V_3 = \sum_{j=1}^{m} \left[G_{3j}(\vec{u}_3) \left(1 - G_{2j}(\vec{u}_2)\right) \left(1 - G_{1j}(\vec{u}_1)\right) \right] \tag{16}$$

or

$$\begin{aligned} V_3 = {} & G_{31} + \cdots + G_{3m} - G_{11}G_{31} \cdots - G_{1m}G_{3m} \\ & - G_{21}G_{31} \cdots - G_{2m}G_{3m} \\ & + G_{11}G_{21}G_{31} + G_{12}G_{22}G_{32} + G_{13}G_{23}G_{33}. \end{aligned} \tag{17}$$

Now, V is simply the sum of the above:

$$V = V_1 + V_2 + V_3, \tag{18}$$

or

$$\begin{aligned} V = {} & \sum_{j=1}^{m} G_{1j} + G_{2j} + G_{3j} - G_{1j}G_{2j} - G_{1j}G_{3j} \\ & - G_{2j}G_{3j} + G_{1j}G_{2j}G_{3j}. \end{aligned} \tag{19}$$

Note that while Eq. 12 is a recursive definition, it yields an objective function that is symmetric in all terms (sets of camera parameters).

First, we'll define a function τ as follows:

$$\begin{aligned}\tau(x_1, x_2, \cdots, x_n) = {} & x_1 + x_2 + x_3 + \cdots + x_n \\ & -x_1x_2 - x_1x_3 - \cdots - x_{(n-1)}x_n \\ & +x_1x_2x_3 + x_1x_2x_n \cdots + x_{(n-2)}x_{(n-1)}x_n \\ & (-1)^{n+1}x_1x_2 \cdots x_n \end{aligned} \tag{20}$$

Then, we can re-write Eq. 12 as:

$$V = \tau\left(\sum_{j=1}^{m} G_{1j}, \ldots, \sum_{j=1}^{m} G_{nj}\right). \tag{21}$$

This is very important, because it indicates that the order in which the camera placement is optimized does not effect the outcome of the optimization result for the multi-camera system. Via substitution, we can change the order in which the camera parameters are considered, yet Eq. 18 remains identical.

In addition, this formulation ensures that the maximum gain of any path is 1, regardless of the number of cameras. For example, note that for any path j, the following equation holds (It is important for this analysis to recall that $\forall\ G_{ij} : 0 \leq G_{ij} \leq 1$.):

$$0 \leq G_{1j} + G_{2j} + G_{3j} - G_{1j}G_{2j} - G_{2j}G_{3j} - G_{1j}G_{3j} + G_{1j}G_{2j}G_{3j} \leq 1 \tag{22}$$

This intuition may be further clarified if we consider the following. We can find the optimal value of the objective function by setting the partial derivatives of the function to zero and evaluating. In general:

$$\frac{\partial V}{\partial \vec{u}_k} = \frac{\partial \sum_{j=1}^{m} G_{kj}}{\partial \vec{u}_k} \cdot \left(1 - \tau\left(\sum_{j=1}^{m} G_{1j}, \ldots, \sum_{j=1}^{m} G_{(k-1)j}, \sum_{j=1}^{m} G_{(k+1)j}, \ldots, \sum_{j=1}^{m} G_{nj}\right)\right). \tag{23}$$

In the trivial single camera case, then

$$\frac{\partial V}{\partial \vec{u}_1} = \frac{\partial \sum_{j=1}^{m} G_{1j}}{\partial \vec{u}_1} = \sum_{j=1}^{m} \frac{\partial G_{1j}}{\partial \vec{u}_1} = 0 \tag{24}$$

This equation is maximized when each path is optimally observed by the camera $\left(\forall j : \frac{\partial G_{1j}}{\partial \vec{u}_1} = 0\right)$.

We have found that even for complex cases where clusters are not independent, the iterative method requires only 1 or 2 more cameras in its solution than the optimal method to capture all the path information. Therefore, we have not found a need to use the much more expensive optimal approach in practice.

3.2 Optimization

In the general case, $\vec{u}_i$ might be defined as:

$$\vec{u}_i = \left[\, X_{c_i}\ Y_{c_i}\ Z_{c_i}\ \gamma_{x_{c_i}}\ \gamma_{y_{c_i}}\ \gamma_{z_{c_i}}\ f \,\right]^T \tag{25}$$

where $(X_{c_i}, Y_{c_i}, Z_{c_i})$ corresponds to camera position, $(\gamma_{x_{c_i}}, \gamma_{y_{c_i}}, \gamma_{z_{c_i}})$ correspond to camera orientations (pitch, roll, yaw) about the (X, Y, Z) axes, and f is the focal length of the camera. This would allow optimization over the six extrinsic camera parameters along with focal length, the primary intrinsic parameter.

This formulation may be applied to many vision problems. We chose to focus first on the problem of camera placement to maximize path observability in a scene. In these applications, the camera placement above the ground is often highly constrained. Camera mounting is limited to points such as the roofs of buildings. This does not leave Z_{c_i} as a free parameter. In addition, we found that pitch (γ_x) is highly coupled to the height above the scene, thus we fix these parameters based on the needs of each scene. In addition, we ignore the effect of roll (γ_y), since it simply rotates the image plane about the optical axis, and thus has an insignificant effect on observability. Lastly, we assume systems of identical fixed field of view cameras, thus f is held constant.

Our objective function is thus reduced to:

$$G_{ij} = \begin{cases} 0 & \text{if } d_{ij}^2 < d_0^2, \\ \frac{d_0^2}{d_{ij}^2} cos\left(\theta_{ij}\right) cos\left(\phi_{ij}\right) & \text{otherwise} \end{cases} \tag{26}$$

and the action vector $\vec{u}_i$ becomes:

$$\vec{u}_i = \left[X_{c_i} \; Y_{c_i} \; \gamma_{z_{c_i}} \right]^T. \tag{27}$$

Equations 28 through 30 introduce a change of variables to convert the objective function from relative variables $(d_{ij}, \theta_{ij}, \phi_{ij})$ into absolute variables $(X_{c_i}, Y_{c_i}, \gamma_{z_{c_i}})$ to position and orient each camera in the world coordinate system.

$$d_{ij} = \sqrt{\left(x_j - X_{c_i}\right)^2 + \left(y_j - Y_{c_i}\right)^2} \tag{28}$$

$$\theta_{ij} = cos^{-1}\left(\frac{T_1 - T_2}{T_3 \cdot T_4}\right) \tag{29}$$

where

$$\begin{aligned} T_1 &= \left(y_{sj} - y_j\right)\left(x_j - X_{c_i}\right) \\ T_2 &= \left(x_{sj} - x_j\right)\left(y_j - Y_{c_i}\right) \\ T_3 &= \sqrt{\left(x_j - x_{sj}\right)^2 + \left(y_j - y_{sj}\right)^2} \\ T_4 &= \sqrt{\left(x_j - X_{c_i}\right)^2 + \left(y_j - Y_{c_i}\right)^2} \end{aligned}$$

and $(x_{sj}, \; y_{sj})$ is the starting point of each path.

$$\phi_{ij} = cos^{-1}\left(\frac{P_1 - P_2}{P_3 \cdot P_4}\right) \tag{30}$$

where

$$P_1 = (y_j - Y_{c_i})\, cos\left(\gamma_{z_{c_i}}\right) cos\left(\gamma_{x_{c_i}}\right)$$

$$P_2 = (x_j - X_{c_i})\, sin\left(\gamma_{z_{c_i}}\right) cos\left(\gamma_{x_{c_i}}\right)$$

$$P_3 = \sqrt{cos^2\left(\gamma_{z_{c_i}}\right) cos^2\left(\gamma_{x_{c_i}}\right) + sin^2\left(\gamma_{z_{c_i}}\right) cos^2\left(\gamma_{x_{c_i}}\right)}$$

$$P_4 = \sqrt{\left(x_j - X_{c_i}\right)^2 + \left(y_j - Y_{c_i}\right)^2}.$$

We solve for the parameters $(X_{c_i},\ Y_{c_i},\ \gamma_{z_{c_i}})$ through a iterative-refinement based constrained nonlinear optimization process. The method we used was a modified version of the well-known hill-climbing alorithm, where we evaluate the constrained function Eq. 26 at uniformly spaced intervals of the parameters of $\vec{u}_i$. In regions where the absolute slope $|\frac{\partial V}{\partial \vec{u}}|$ is large, we refine the search by decreasing the spacing interval and iterating. Initial discretization of the parameter space is determined by hand. Furthermore, to decrease computational burden, we limit the parameter space of each camera by removing parameter ranges that correspond to fields of view that do not contain any part of the path distribution.

Of course, this method is quite straightforward, and does not guarantee avoidance of local minima, yet we have found this process to work very well in our numerous simulated and real world experiments. It allows us to be reasonably certain that the process avoids local minima because we maintain a global picture of the objective surface, while providing very accurate estimates of the parameters in the refined regions. In addition, we found this method to be faster than Newton-Raphson methods in the presence of complex sets of constraints. For example, for the real-world scenes described in Section 3.4, we found that all solutions could be computed in between 22 s and 1.7 min using a 2.66-GHz single processor Pentium IV computer running Matlab.

3.3 Simulation Results

Figures 7, 8, and 9 show the results of the method on simulated path distributions to illustrate the approach. In all the result figures shown in this paper, the objective surface shown corresponds to the 2D slice of $\gamma_{z_{c_i}}$ where the objective function is optimized.

Figure 7 shows results for the case of a single camera and single path. Note how the objective surface varies with position. The function has two optima: one on either side of the path. The simulation result confirms the intuition that the optimal camera position to observe a single path is perpendicular to that path, at a distance of d_0. For this single path case, the residual observability that remains after the first camera is placed is near zero.

Figure 8 shows results for the multiple camera system case, where the paths are all parallel. Note that the objective surface is modified for the second camera. The paths that are viewed well by the first camera are discounted when computing the position of the second camera.

Figure 9 shows results for a more complicated path distribution, similar to one that might be observed at a traffic intersection.

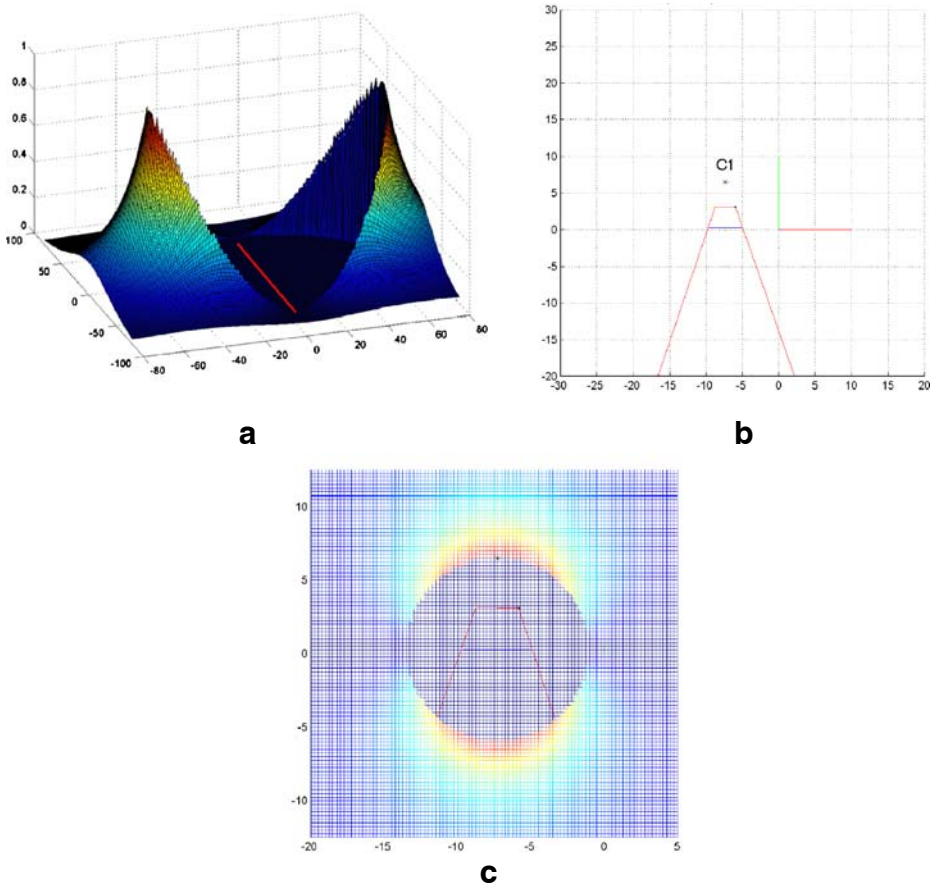


Fig. 7 Objective surface and camera placement solution for the single path, single camera case. *Labeled dots* indicate camera locations. **a** Objective surface. **b** Solution. **c** Camera C1 objective surface (from above). In **c**, *red lines* indicate the view frustum of the camera

For comparison, Fig. 10 shows the results of the optimal multi-camera placement method for 3 cameras on the same traffic intersection path distribution. Note that the objective surface computed by this method is very similar to that of the iterative method, and it yields camera placement results that are almost identical.

3.4 Experimental Results

We tested this system on motion paths captured at several indoor and outdoor locations. Video was captured by a single fixed camera placed at each scene. In order to estimate the paths of the subjects' motions in the scene, the video was processed to segment each subject from the background, compute the position of the subject in the frame, and track the subject's position over time. Segmentation was achieved through the use of a Gaussian mixture model-based adaptive background

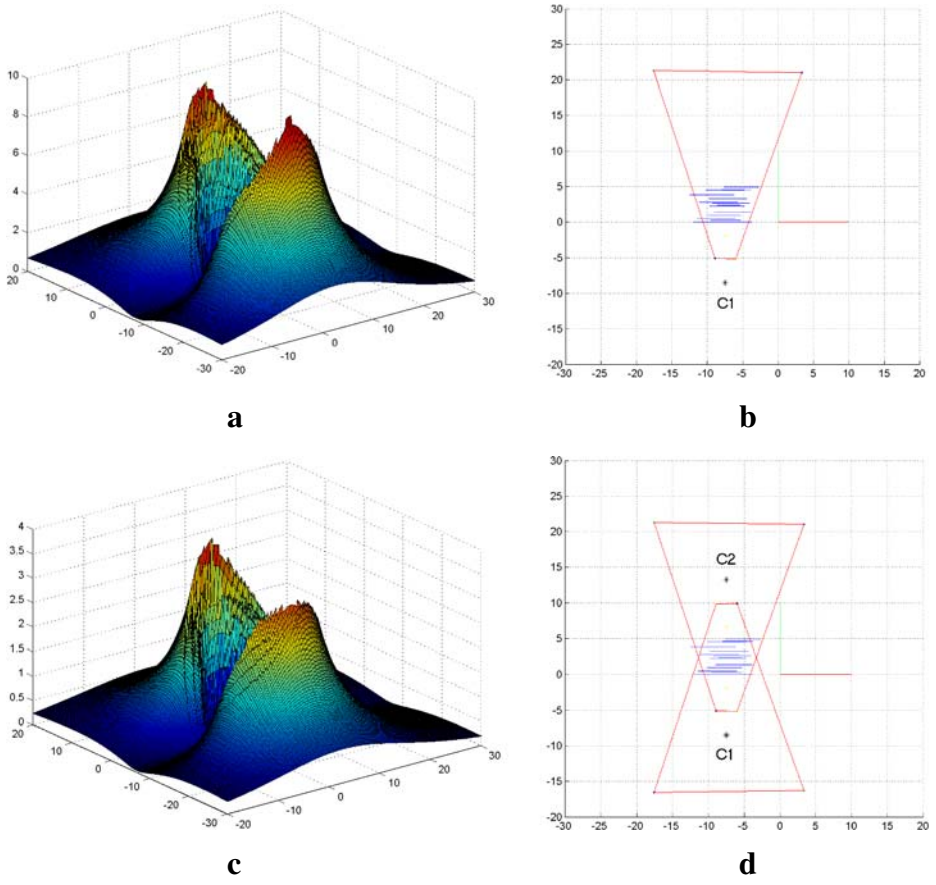


Fig. 8 Objective surface and camera placement solution for multiple parallel paths. *Labeled dots* indicate camera locations. Note how the objective surface changes after the first camera is positioned. **a** Objective surface—first camera. **b** Solution—first camera. **c** Objective surface—second camera. **d** Solution—second camera

segmentation algorithm. In addition, a Kalman filter was used for tracking. Linear paths were estimated from the tracking data using a linear least-squares fitting.

In each experiment, the camera was calibrated relative to the ground plane using the method described in Masoud and Papanikolopoulos [22]. Thus, all path distributions shown are in world coordinates, as are the camera placement solutions.

In each case we show results for the number of cameras necessary to observe the paths well, with negligible residual. Figure 11 shows results for an indoor scene of human motion. For this path distribution, two cameras are sufficient to observe all paths well. Note that cameras placed in the corners of the room (the standard solution) would not be positioned optimally to observe this distribution. Figure 12 shows results for an outdoor pedestrian courtyard and street scene. Here the first two cameras focus attention on the dominant pathways, while the third camera attends to the contribution of the less common paths. Figure 13 shows results for a pedestrian

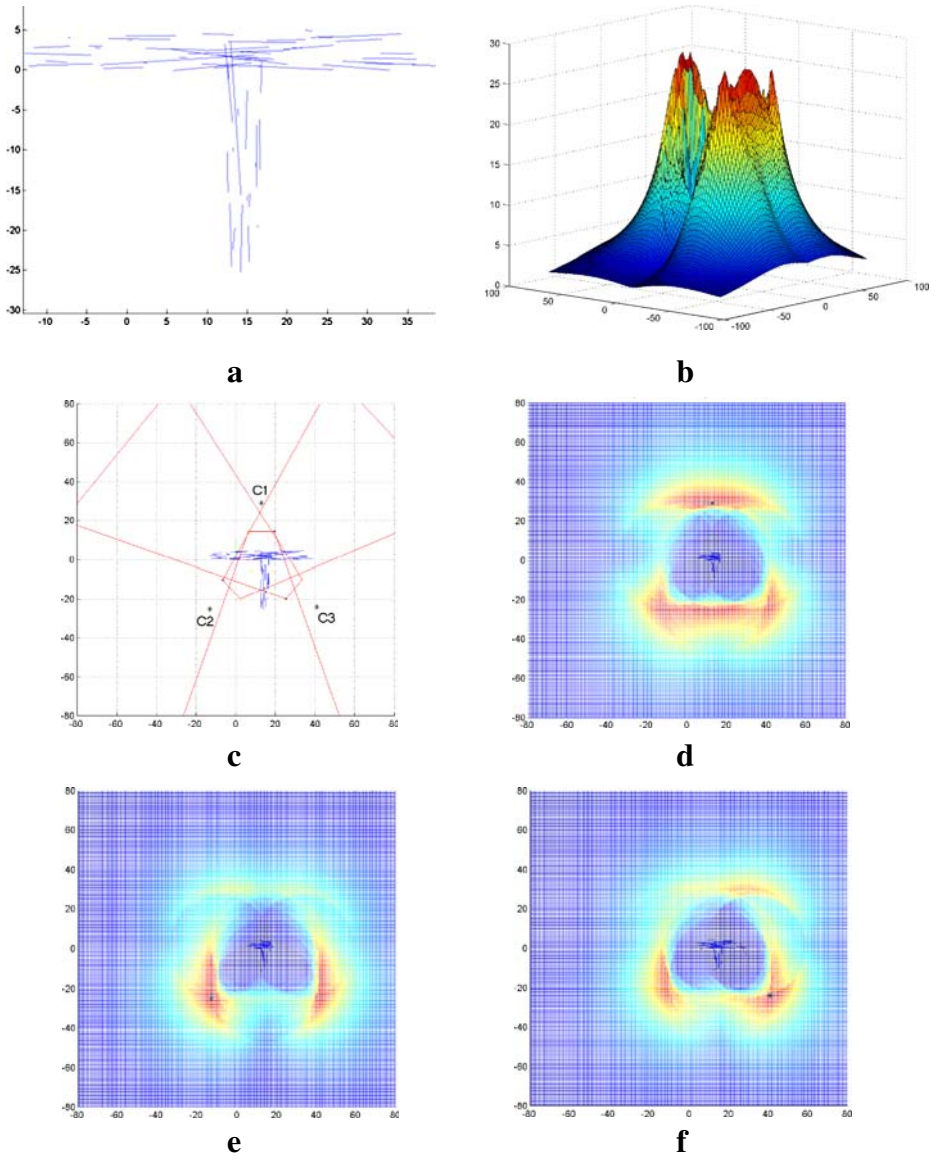


Fig. 9 Results for a simulated traffic intersection path configuration. *Labeled dots* indicate camera locations. Camera placement results are shown for 3 cameras, along with the initial objective surface. **a** Paths. **b** Objective surface. **c** Multiple camera solution. **d** Camera C1 objective surface (from above). **e** Camera C2 objective surface (from above). **f** Camera C3 objective surface (from above)

courtyard scene. Due to the complexity of this path distribution, four cameras were needed to observe all paths well. Figure 14 shows results for a traffic intersection. Note that the camera placement solution is very similar in nature to that predicted by the simulated intersection.

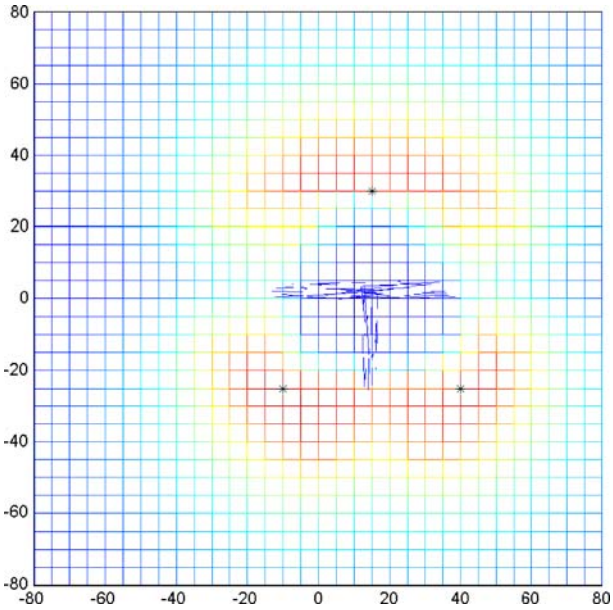


Fig. 10 Camera placement results obtained using an optimal algorithm which searches jointly over all camera parameters at the same time. *Black dots* indicate camera locations

Table 1 summarizes the total observability value of the camera configurations for the experiments. We compare these results with configurations of cameras placed uniformly around a circle, with each camera oriented to look at the center of the path distribution. A circular placement is rather intuitive, and certainly the predominant status quo for multi-camera system configurations. However, it is not optimal for most path configurations. As Fig. 15 shows, peaks may occur at varying distances and

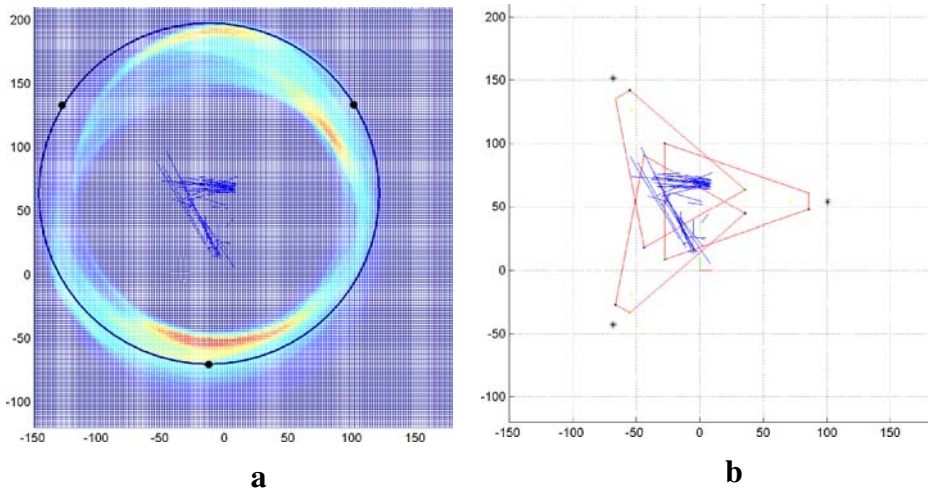
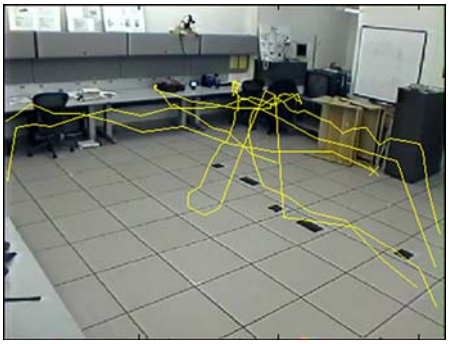
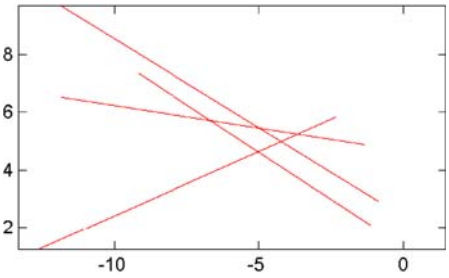


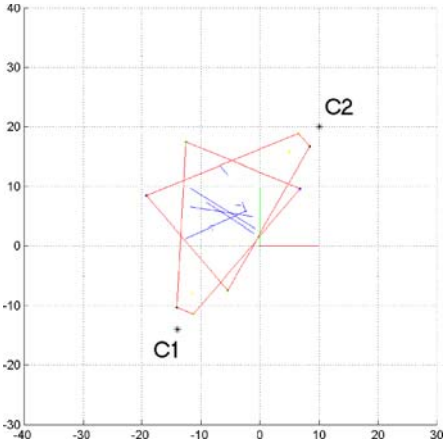
Fig. 11 Uniformly positioned cameras. **a** Circular configuration shown with a sample objective surface. **b** A sample camera solution



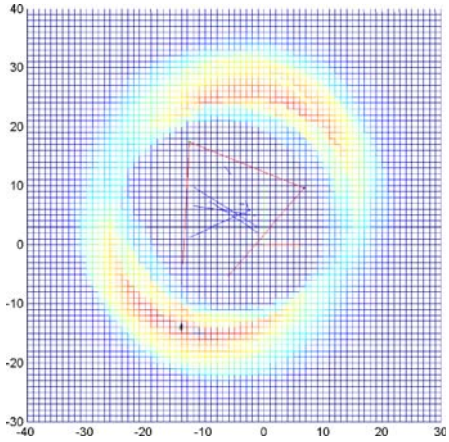
a



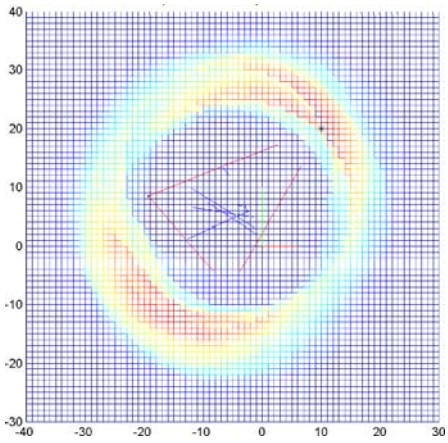
b



c



d



e

◀ **Fig. 12** Results for an indoor motion path configuration. *Labeled dots* indicate camera locations. Camera placement results are shown for 2 cameras, along with the objective surface for each camera. **a** Scene. **b** Tracked paths. **c** Multiple camera solution. **d** Camera C1 objective surface (from above). **e** Camera C2 objective surface (from above)

orientations from the center of the path distribution. As a result, a uniform camera distribution will miss the peaks in most cases. The best radius to choose for placing the cameras is not obvious. If we make the conservative choice of selecting a camera distance such that every path falls within the field of view of every camera, cameras will be farther from the paths than necessary. However, if we bring them too close, then some paths will not be completely viewed by any camera. In addition to the distance from the path center, the locations of the cameras on the circle are not obvious. While they are evenly distributed, this configuration has radial symmetry equal to the inter-camera angle ($2\pi/m$ radians, where m is the number of cameras). A rotation of the entire configuration will result in a different camera placement solution, and a different observability result.

To compare our method with this approach, we spread cameras uniformly around a circle of radius r. (The number of cameras was set equal to the optimal number returned by our method for each experiment.) We did an exhaustive search over both radius and angle. We varied the radius of the circle, and for each radius, we rotated each configuration $360/m$ times, and computed the observability of each case. Table 1 shows results for the best case and for the average of these observability results. Note that without a camera placement theory to guide the choice of configuration, such as the one we have presented, the best case results shown here can only be found via an exhaustive search of this parameter space, thus the average results may be a more valid comparison for arbitrarily placed camera configurations viewing a scene.

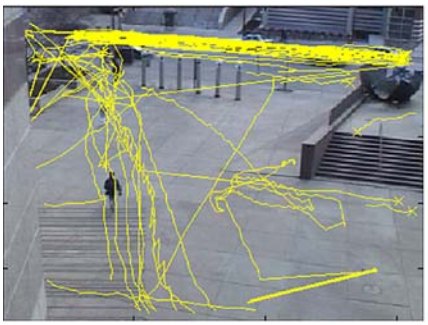
3.5 Extensions

3.5.1 Online Application

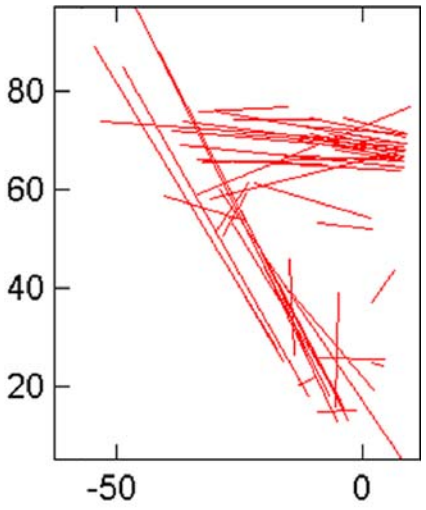
In this section, we present an experiment that demonstrates an active vision application of this work. We relax the constraint that the camera position is fixed, and consider the problem of adapting the camera placement to changing path configurations.

Table 1 Observability of multi-camera placement configurations

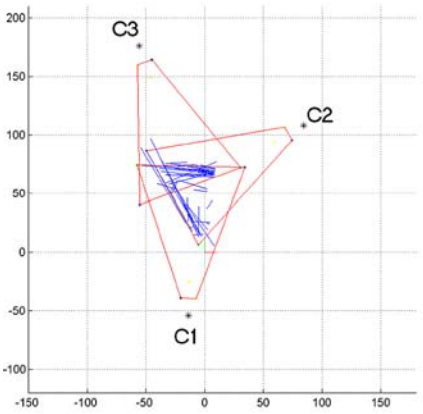
Scene	Theoretical max	Evenly distributed				Proposed method	
		Average	Percent max (%)	Best case	Percent max (%)	Obs based	Percent max (%)
Indoor	8	3.93	49	7.05	88	7.98	99
Pedestrian and vehicle	39	9.44	24	27.71	71	35.79	92
Pedestrian only	21	10.85	52	17.59	84	19.90	95
Vehicle only	43	23.01	54	36.11	84	41.56	97



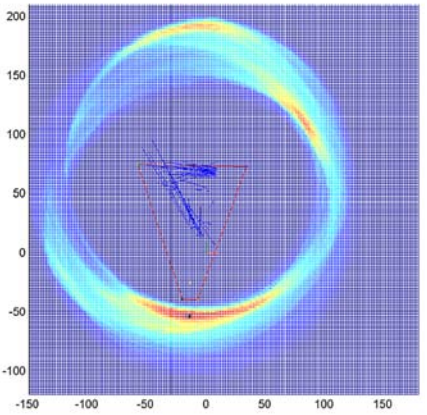
a



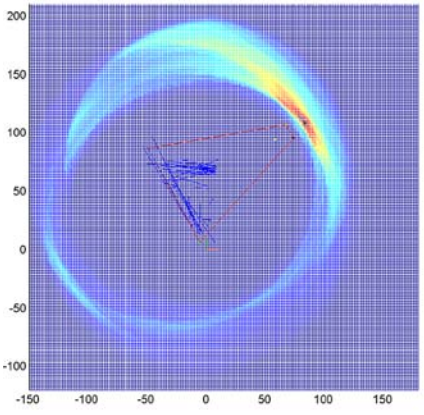
b



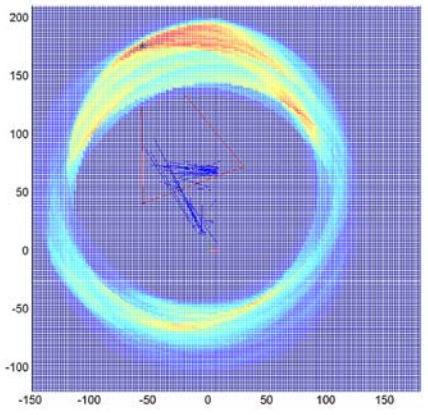
c



d



e



f

◀ **Fig. 13** Results for an outdoor motion path configuration, including pedestrian and vehicle traffic. *Labeled dots* indicate camera locations. Camera placement results are shown for 3 cameras, along with the objective surface for each camera. **a** Scene. **b** Tracked paths. **c** Multiple camera solution. **d** Camera C1 objective surface (from above). **e** Camera C2 objective surface (from above). **f** Camera C3 objective surface (from above)

We use a single mobile camera to observe a moving subject as he walks back and forth between two occluding screens. The camera is initialized to an arbitrary position in the room, while ensuring that all paths fall within the initial view. The system monitors the paths taken through the scene, computes the optimal camera

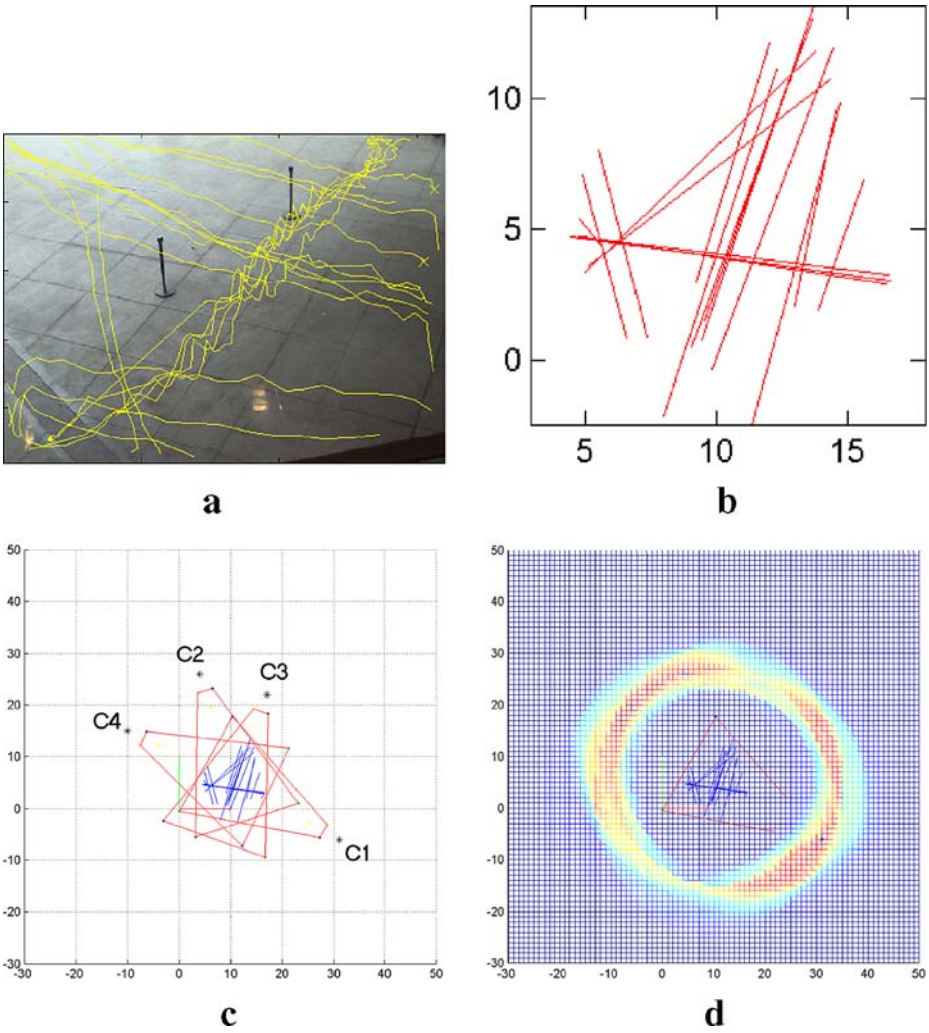


Fig. 14 Results for a pedestrian courtyard path configuration. *Labeled dots* indicate camera locations. Camera placement results are shown for 4 cameras, along with the objective surface for each camera. **a** Scene. **b** Tracked paths. **c** Multiple camera solution. **d** Camera C1 objective surface. **e** Camera C2 objective surface. **f** Camera C3 objective surface. **g** Camera C4 objective surface

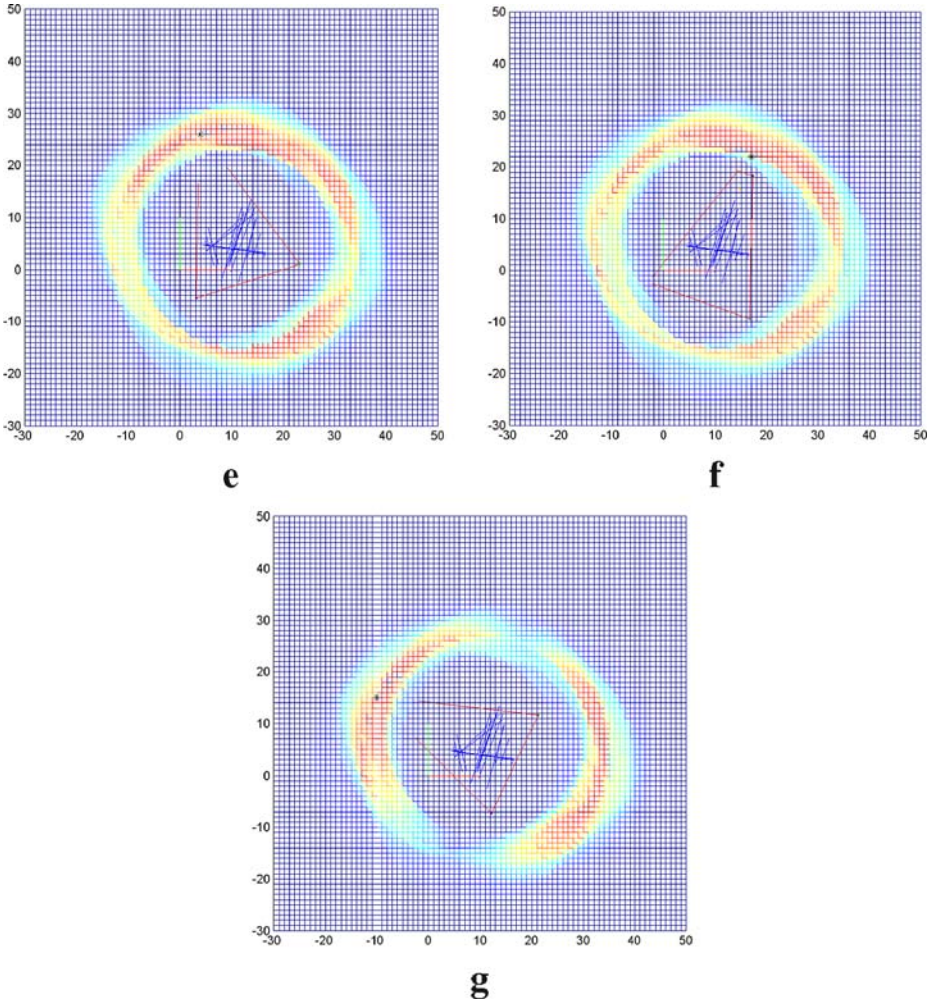


Fig. 14 Continued

position to view this new path distribution, and moves to the new goal position. Figure 16 shows the arrangement of the experiment.

Equipment setup This experiment makes use of a modified all terrain robot vehicle ATRV-Jr robot, shown in Fig. 17a. The modifications include a custom built "sensor mast" which allows the positioning of cameras and other sensors at an elevated vantage point. A laptop is mounted on the rear of the "sensor mast" to perform video processing tasks that are too CPU intensive for the ATRV-Jr's on-board systems.

The experiment required the integration of several systems as shown in the diagram in Fig. 18. Here, a firewire camera (Sony DCR-TRV730) is used to provide a video stream to a software module that captures video and tracks movement.

In order to estimate the paths of the subject's motion, the video was processed to segment the subject from the background, compute the position of the subject in the frame, and track the subject's position over time. These paths are shown in Fig. 19. Segmentation was achieved through the use of a Gaussian mixture model-based adaptive background segmentation algorithm. In addition, a Kalman filter was used for tracking.

Figure 20 illustrates the paths estimated from the position tracking data. The new position is then transmitted as a location goal to the robot. A player-client [17] receives the new location goal, and through an interaction with a player-server, directs the motion of the ATRV-Jr to the optimal location. As the movement of subject through the scene changes, the process repeats continuously to keep the camera in a location that provides the optimal observability of the motion sequences.

System calibration The camera was calibrated to determine its intrinsic parameters, and to determine the extrinsic parameters of the camera with respect to the plane of the ground. We used the method of Masoud and Papanikolopoulos [22]. The calibration involved selecting parallel and perpendicular lines on the ground plane in an image (Fig. 21). The result was a homography transformation matrix between the camera's image plane and the ground plane.

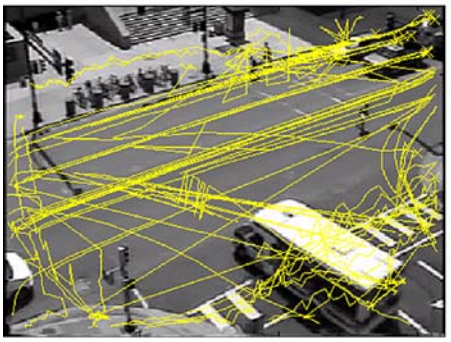
Experimental results Figures 22 and 23 show results for one trial of the experiment. The objective surface for this path distribution is very similar to the simulation result for the multiple parallel path case. Note, however, that the shape of the joint objective surface is not exactly circular, and is somewhat elongated because of the shape of the actual path distribution. The resulting view (Fig. 24b) of the walking motion has significantly improved observability from the initial view (Fig. 24a).

Table 2 shows the results of six experimental runs with the mobile robotic platform. The first row shows the observability value of the motion sequences, measured from the initial camera position. The second row shows the observability value of the motion sequences, measured from the final camera position. The third row shows the improvement due to the new camera pose. The fourth and fifth rows show the theoretical maximum observability of the sequences, and the difference between the theoretical and final (optimized) values.

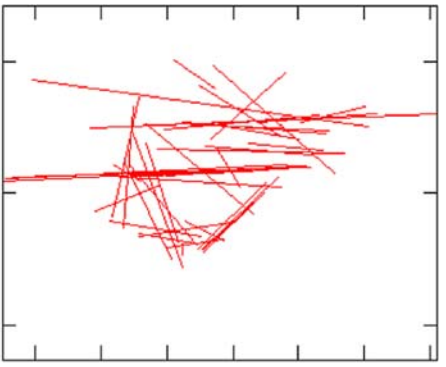
Note that while none of the final positions of the camera reached optimal observability, all runs showed significantly improved observability. We believe that the differences between theoretical maximum observability and our results are due to a combination of robot positioning error and path estimation error. Robot

Table 2 Observability results

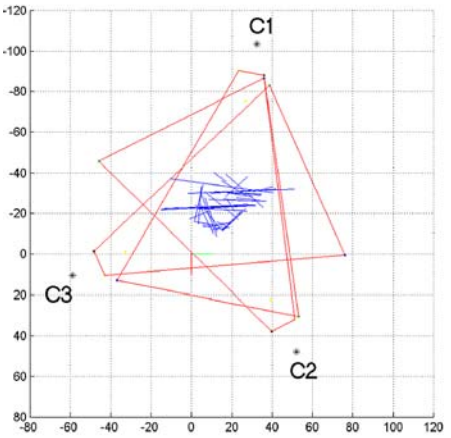
Observability	Trial					
	1	2	3	4	5	6
Initial position	0.55	0.63	0.58	0.54	0.59	0.50
Final position	0.69	0.88	0.85	0.80	0.95	0.78
Percent improvement	25%	40%	47%	48%	61%	57%
Theoretical maximum	1.00	1.00	1.00	1.00	1.00	1.00
Diff. from theoretical	0.31	0.12	0.15	0.20	0.05	0.22



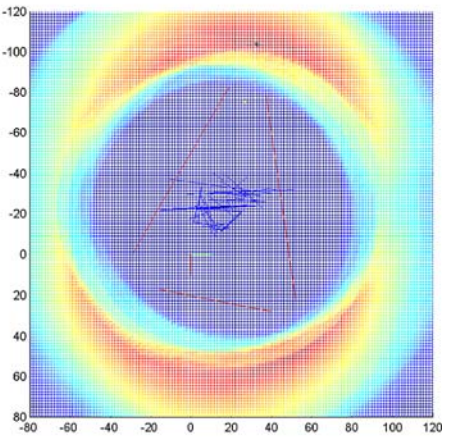
a



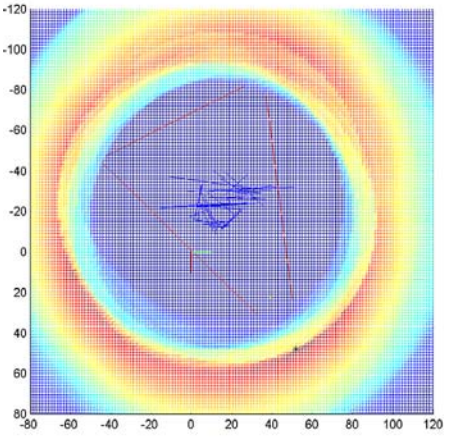
b



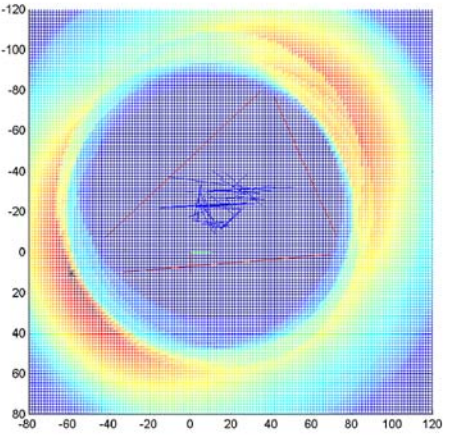
c



d



e



f

◀ **Fig. 15** Results for a traffic intersection path configuration. *Labeled dots* indicate camera locations. Camera placement results are shown for 3 cameras, along with the objective surface for each camera. **a** Scene. **b** Tracked paths. **c** Multiple camera solution. **d** Camera C1 objective surface. **e** Camera C2 objective surface. **f** Camera C3 objective surface

(and therefore camera) positioning error was introduced because of imprecision in the robot's odometry. In our experiments, the robot consistently demonstrated positioning errors of 5 to 25 cm at its final position. We believe this comprised the major source of error for our system.

Path estimation may have also contributed to the overall error. Each path taken by the subject is tracked over time. Small errors in the position estimates of the subject in each image would cause minor errors in the path estimates. In addition, there is some inherent error in estimating the paths by line fitting.

Sign Resolvability To further demonstrate, and better quantify, the improved observability that the proper camera view provides, we repeated the above experiment while the subject carried a sign containing alphanumeric symbols and sets of parallel lines at varying scales (Fig. 25).

The subject moved back and forth between the occlusions and was tracked, as above. The new path distribution was estimated and the robot was given a new goal command to move the camera to the position of optimal path observability. We tested the resolvability of the sign at several positions along the robot's motion path. In this case, sign resolvability is defined as the number of characters that is clearly discernible and unambiguous to a human observer. In the case of lines, we define resolvability as the number of sets of line triplets where all 3 lines are clearly distinct and separate.

Figure 26 shows views of the sign taken at several positions along the robot's motion path. Table 3 displays the results of the resolvability of the sign for both alphanumeric characters and line triplets.

Furthermore, we tested the resolvability of the characters on the sign using an automated character recognition system, in addition to a human observer. Specifically, we used the Cuneiform™ Pro 6.0 optical character recognition software to detect the characters on the sign at each position along the motion path. Results for this test appear in Table 4.

It is important to note that the camera was positioned to optimize for path observability, not sign resolvability. As a result, in no view are all of the characters clearly resolvable. However, there is a very significant improvement in the relative performance of the recognition at the optimized view, indicating the value of the method.

Table 3 Sign resolvability—human observer

	Initial view	Middle view	Final view
Alphanumeric	12	5	24
Lines	3	3	14

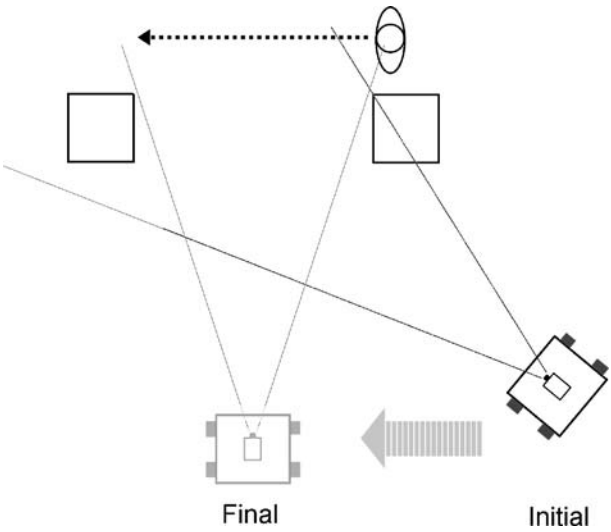


Fig. 16 Overview of the experiment

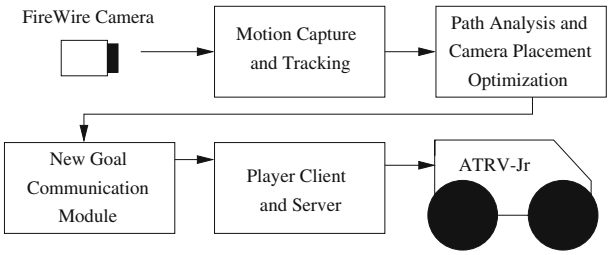


Fig. 17 Block diagram of the ATRV-Jr camera system

a **b**

Fig. 18 The robotic setup consisting of an ATRV-Jr and a "sensor mast." **a** Front view. **b** Side view

Table 4 Sign resolvability—optical character recognition software

	Initial view	Middle view	Final view
Alphanumeric	1	2	10

3.5.2 Incorporating Design Constraints

As we have discussed, in real world scenes, camera placement is often constrained by the physical environment. Indoors, cameras usually need to be placed on or along the ceiling in order to achieve unoccluded views. Outdoors, cameras generally need to be placed on rooftops or light posts in order to observe an area.

In an attempt to address this real-world placement constraint, and provide a better design tool for camera placement, we extended the formulation to include placement constraint regions. The optimization procedure on camera parameters is confined to the user-defined valid camera placement regions. This modification speeds up the analysis, and provides a solution relevant to the physical environment. In addition, these results can be compared to optimal results obtained via unconstrained optimization in order to determine how much of an effect the constraints have on the observability of the scene.

Figure 27 shows the results of this method for the simulated traffic intersection data. In this case, the dark areas of the figure represent building rooftops. Cameras may be mounted at the edges of these areas. Note that the camera placement is optimized with respect to the imposed placement constraints.

3.5.3 Occlusion

We extended the formulation to incorporate static occlusions in the environment. Such occlusions have obvious and potentially significant impact on the observability of a scene. We incorporate occlusion into our model by removing occluded paths from the value calculation for a given set of camera parameters. If an occluding body comes between a camera and a path, that path cannot be completely observed by the camera. As a result, it does not contribute to the value of the observability seen by that camera. The location and dimensions of such occlusions are assumed to be

Fig. 19 Path lines observed from the initial view

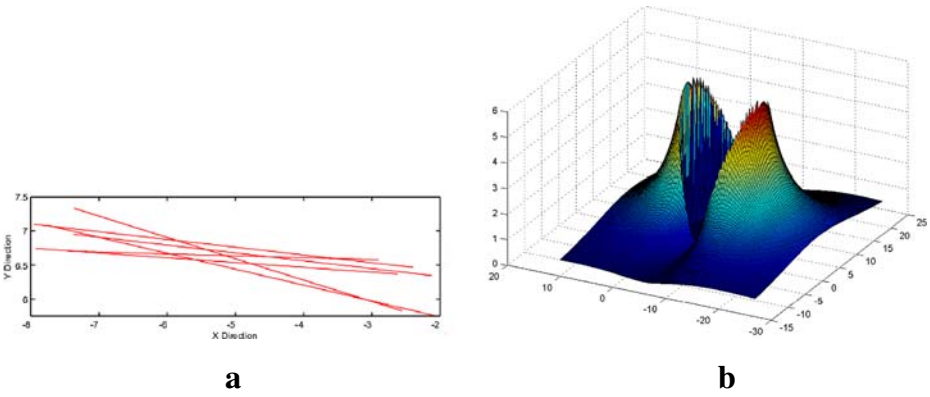


Fig. 20 Path lines and resultant objective surface. **a** Path lines projected onto the ground plane (world coordinates). **b** Objective surface

Fig. 21 Calibration setup. The calibration grid shown here was removed for experiments

Fig. 22 Initial and final view of the camera during a run. **a** Initial view. **b** Final view

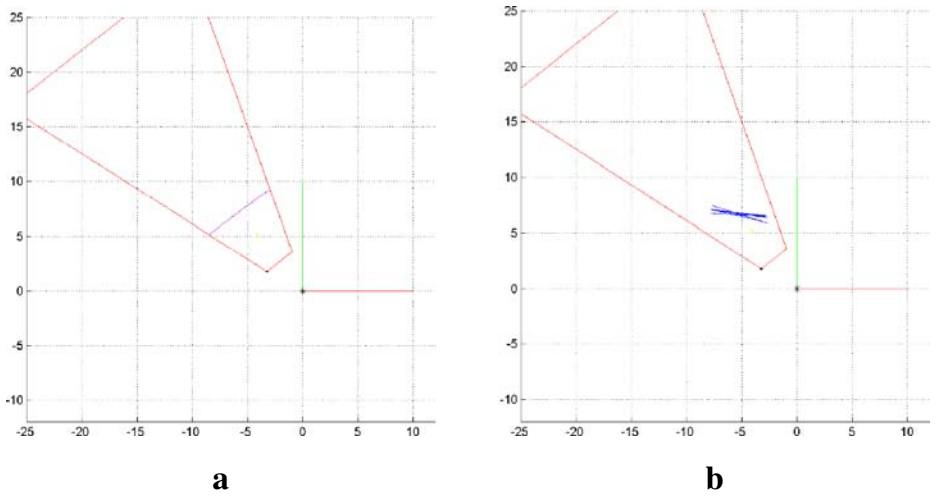


Fig. 23 Initial camera positioning and view of paths. Figures show camera view frustum projected onto ground plane. **a** Path configuration at initialization. Camera is positioned to optimize the view of this single path. **b** Observed path configuration relative to camera pose

known for this analysis. (They may be measured by hand as an initialization step for the scene, or can be estimated online via a method such as Jackson et al. [21])

Figure 28 shows the effect of an occlusion on the single-camera, single-path case. The effect of the occluding body on the resultant solution can be seen clearly when compared to Fig. 7. Note that the occlusion significantly changes the objective surface, effectively eliminating one possible solution from consideration.

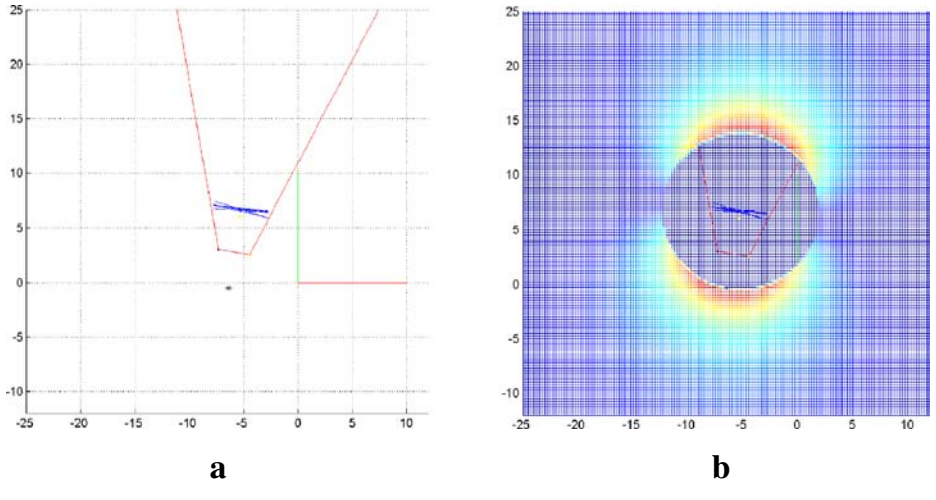


Fig. 24 Camera positioning solution for joint path observability optimization. Figures show camera view frustum projected onto ground plane. **a** Solution shown with path distribution. **b** Solution shown with the objective surface superimposed onto it

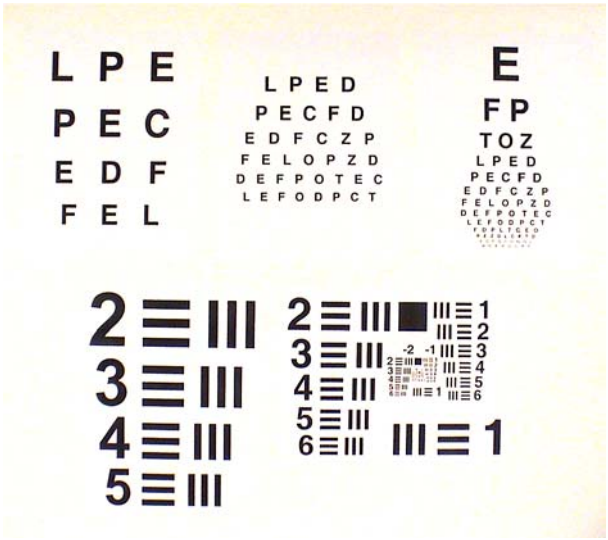


Fig. 25 Sign used for resolvability experiments, containing alphanumeric symbols and sets of parallel lines

Figures 29 and 30 show similar results for actual pedestrian and vehicle traffic scenes. These occlusions suppress portions of the objective surfaces seen in Figs. 12 and 13.

4 Conclusion

In this paper, we have studied the problem of observing humans in largely uncontrolled indoor and outdoor environments. We have described several new approaches to camera placement for automated vision tasks, considering fixed and moving cameras, as well as static occlusions in the scene.

We have developed a novel analytical method for computing optimal camera position and pose for task observability with multiple camera systems. The general formulation was applied to the problem of path observation in wide-area scenes, and extended to consider real-world mounting constraints. This approach was validated in both simulation and real-world experiments involving multiple-camera systems

a **b** **c**

Fig. 26 Views of the sign taken at several positions along the robot's motion path. **a** Initial view (foreshortened). **b** View taken during motion (distant). **c** Final view (optimal)

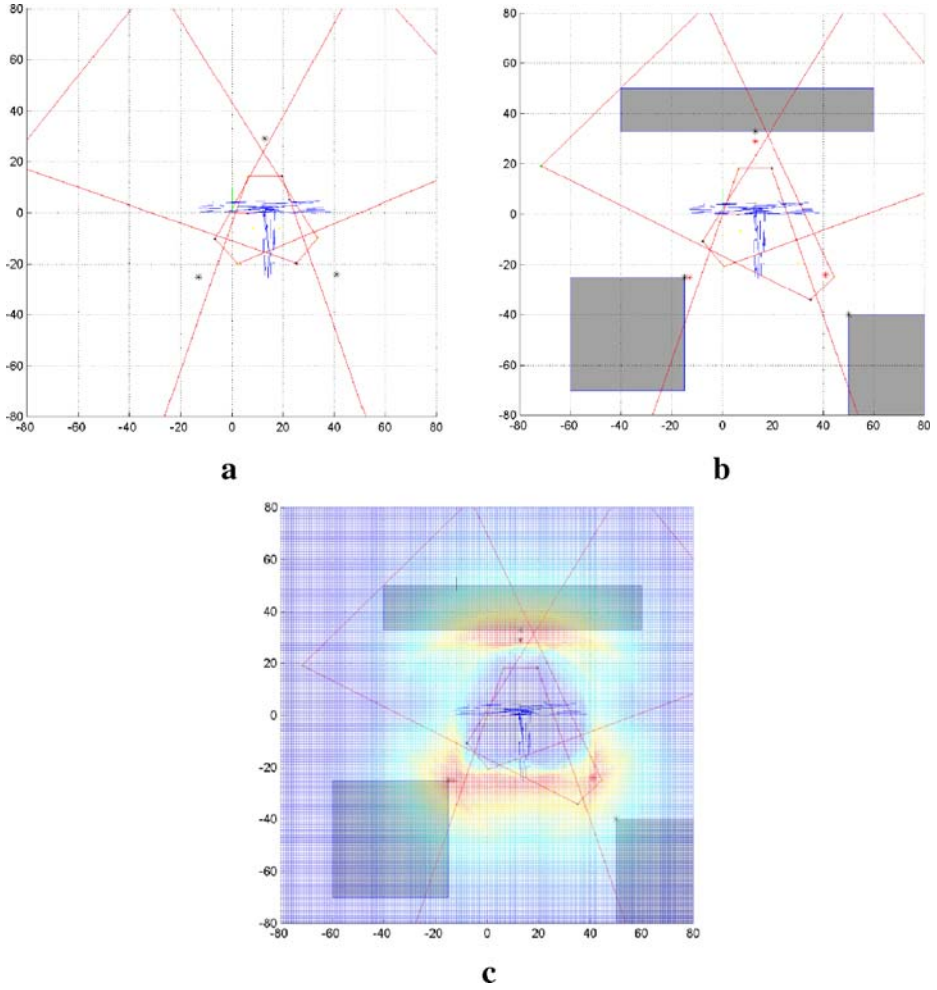


Fig. 27 Constrained camera placement solution. *Grey boxes* indicate building rooftops. *Black dots* indicate camera locations and *lines* indicate the view frustum of each camera. **a** Unconstrained solution. **b** Constrained solution. *Black dots* indicate constrained camera solutions, *red dots* indicate unconstrained solutions. **c** Constrained solutions superimposed on objective surface

observing many subjects moving through a scene. Our findings indicate that this method may be used to guide proper camera placement in complex indoor and outdoor scenes. In addition, the method may be used to guide proper camera placement in dynamically adapting scenes and scenes that contain static occlusions. Furthermore, this approach can aid in the determination of camera placement to provide optimal observability of assembly tasks.

5 Future Work

This is a very rich and interesting problem area, where there is still much work to be done. In this paper, we discuss several natural extensions to the work we

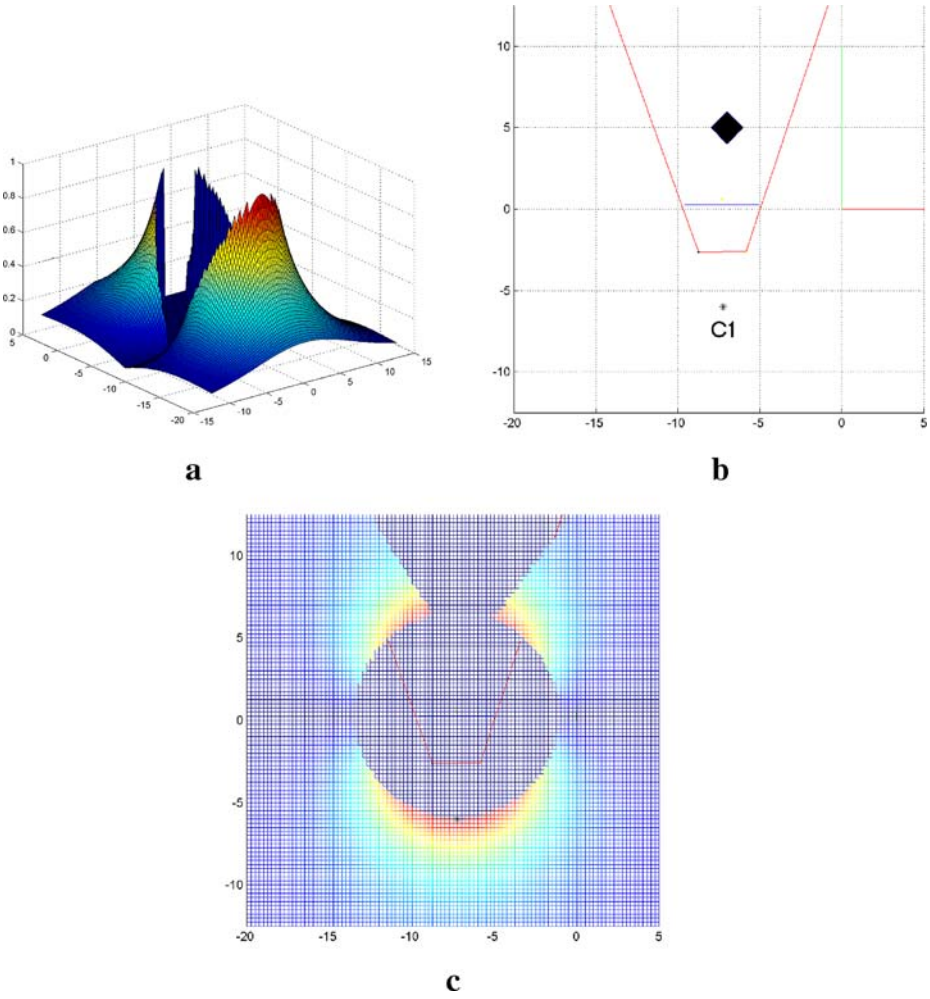


Fig. 28 Objective surface and camera placement solution for the occluded single path, single camera case. *Labeled dots* indicate camera locations. **a** Objective surface. **b** Camera C1 solution. **c** Camera C2 solution. In **b**, lines indicate the view frustum of the camera

have done, along with descriptions of some additional progress we have made in these areas.

5.1 Camera Placement

Our goal with this work was to develop a design tool for camera placement in applications where subject motion varies, and is important to observe. There are many areas into which this work can be extended, including vision-based user interface design and the coordination of multiple robotic systems used to optimally place multiple cameras.

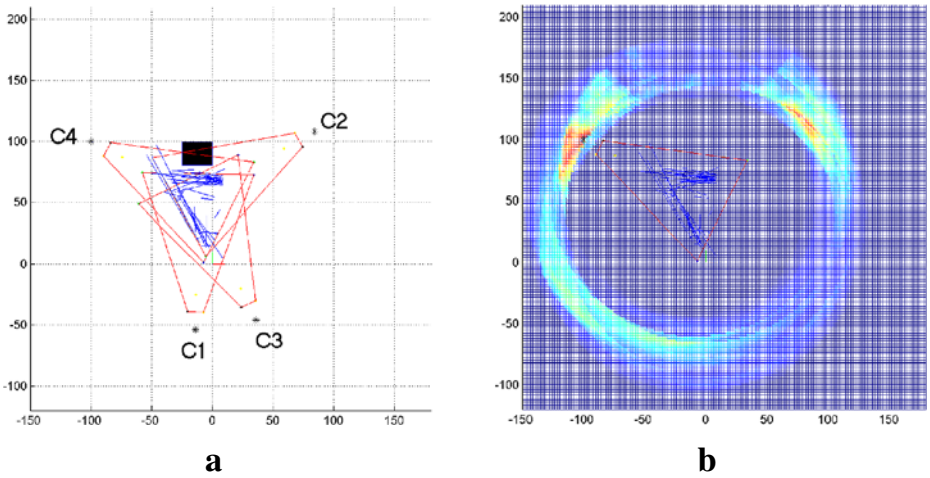


Fig. 29 Results for an occluded outdoor motion path configuration, including pedestrian and vehicle traffic. *Labeled dots* indicate camera locations. Camera placement results are shown for 4 cameras, along with the objective surface of the fourth camera. **a** Multiple camera solution. **b** Objective surface (from above)

A straightforward extension of this work is to extend the state vector of camera parameters to include all extrinsic camera parameters. Optimizing over all extrinsic parameters will provide for general camera placement in cases where camera mounting is completely unconstrained [cameras mounted on unmanned aerial vehicles (UAVs), for instance].

In addition, motions other than locomotive motion may be studied within this framework, if the appropriate changes are made to the state vector s. For example,

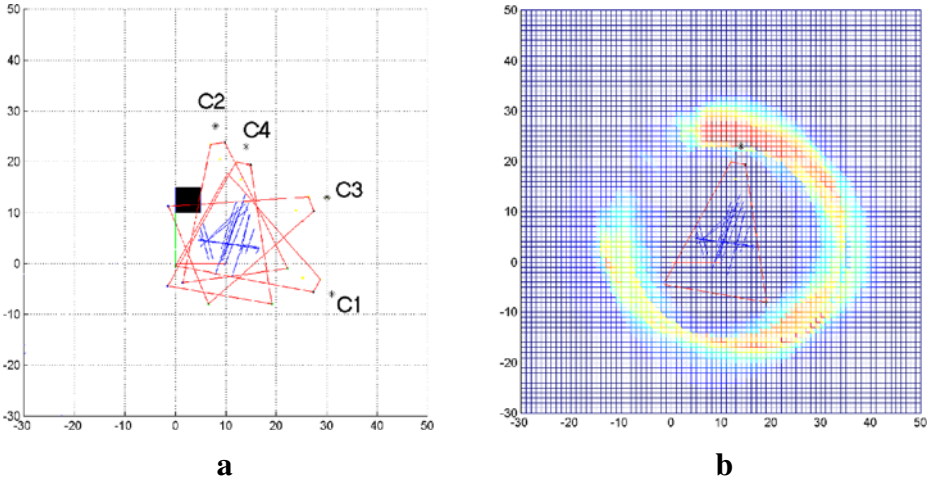


Fig. 30 Results for an occluded outdoor pedestrian motion path configuration. *Labeled dots* indicate camera locations. Camera placement results are shown for 4 cameras, along with the objective surface of the fourth camera. **a** Multiple camera solution. **b** Objective surface (from above)

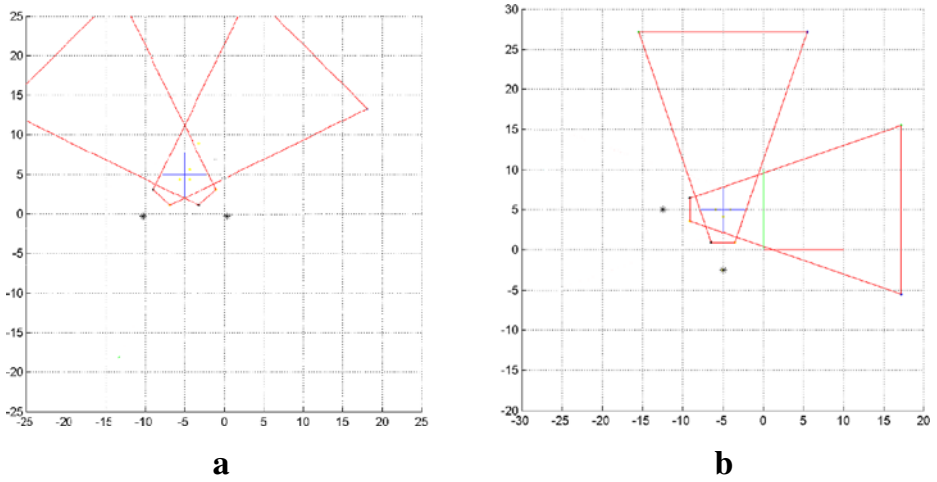


Fig. 31 The effect of varying ω. *Black dots* indicate camera locations. **a** Solution that optimizes observability ($\omega = 1$). **b** Solution that optimizes perpendicularity for recognition ($\omega = 11$)

in the area of user interfacing, the motion of the hands could be tracked and represented in a path distribution.

We developed a camera placement method given static occlusions. An important extension is to study camera placement in the context of moving occlusions. Principally, this would require a means of moving the cameras online to adapt to the varying occlusion conditions. This could be done outdoors using UAV-mounted cameras as we have mentioned, or indoors using cameras mounted on tracks in the ceiling, for example.

5.1.1 The Effect of Focal Length

Another interesting extension is to include intrinsic camera parameters in the state vector. We touched on this by looking at the effect of varying focal length on the resultant camera placement solution. For most of our experiments, the action vector $\vec{u}_i$ was three-dimensional:

$$\vec{u}_i = \left[X_{c_i} \; Y_{c_i} \; \gamma_{z_{c_i}} \right]^T, \tag{31}$$

and did not include focal length f. This was justified because for focal lengths in the linear region (where radial distortion effects are negligible), changing the focal length only changes the minimum distance d_0, and does not generally change the shape of the solution.

5.1.2 Family of Objective Functions

Another intriguing extension of the work is to consider other objective functions. The general problem framework we propose can be used to solve many problems in camera placement. We focused on the most generally applicable problem of

optimizing observability. Other problems include optimal placement for the purposes of three-dimensional reconstruction, activity recognition, face recognition, etc.

We began an initial look at this problem. One can define a family of objective functions on ω as follows:

$$G_{ij} = \begin{cases} 0 & \text{if } d_{ij}^2 < d_0^2, \\ \frac{d_0^2}{d_{ij}^2} \left(cos\left(\theta_{ij}\right) cos\left(\phi_{ij}\right) cos\left(\alpha_{ij}\right) cos\left(\beta_{ij}\right) \right)^{\omega} & \text{otherwise.} \end{cases} \tag{32}$$

Now, by varying the ω parameter, the camera placement solution may be altered to meet different criteria. For example, our analysis focused on the $\omega = 1$ case, which was appropriate for observability. However, the objective function resulting from $\omega = 0$ would be more appropriate for three-dimensional reconstruction applications, because in this case it is important to spread cameras evenly around a subject, and not favor any single view of a path. In contrast, for articulated motion recognition based on image sequences taken from a single viewpoint, it is critical to favor that particular viewpoint (often the view perpendicular to the motion path). In this case, an $\omega > 1$ would tend to drive camera placement toward the paths' perpendiculars. Figure 31 illustrates the different solutions produced by $\omega = 1$ and $\omega = 11$. We believe that the study of varying the objective function used is a very promising area of future research.

Acknowledgements This work was supported by the National Science Foundation through grant #IIS-0219863, #CNS-0224363, #CNS-0324864, and #CNS-0420836, the Minnesota Department of Transportation, the ITS Institute at the University of Minnesota, and an NSF Graduate Research Fellowship. Portions of this work have appeared in works by Bodor et al. [4, 6]. The authors would like to recognize and thank Osama Masoud for his editorial assistance with this paper, and Michael Janssen for his assistance with the ATRV-Jr robot.

References

1. Abrams, S., Allen, P.K., Tarabanis, K.A.: Dynamic sensor planning. In: Proceedings of the IEEE International Conference on Intelligent Autonomous Systems, pp. 206–215. IEEE, Pittsburgh, PA, February 1993
2. Ben-Arie, J., Wang, Z., Pandit, P., Rajaram, S.: Human activity recognition using multidimensional indexing. IEEE Trans. Pattern Anal. Mach. Intell. **24**(8), 1091–1104, August 2002
3. Beymer, D., Konolige, K.: Real-time tracking of multiple people using continuous detection. In: International Conference on Computer Vision (1999)
4. Bodor, R., Drenner, A., Janssen, M., Schrater, P., Papanikolopoulos, N.: Mobile camera positioning to optimize the observability of human activity recognition tasks. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (2005)
5. Bodor, R., Jackson, B., Papanikolopoulos, N.: Vision-based human tracking and activity recognition. In: Proceedings of the 11th Mediterranean Conference on Control and Automation, June 2003
6. Bodor, R., Schrater, P., Papanikolopoulos, N.: Multi-camera positioning to optimize task observability. In: Proceedings of the IEEE International Conference on Advanced Video and Signal-Based Surveillance (2005)
7. Bregler, C., Malik, J.: Tracking people with twists and exponential maps. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, June 1998
8. Carranza, J., Thebalt, C., Magnor, M., Seidel, H.: Free-viewpoint video of human actors. In: Proceedings of ACM SIGGRAPH (2003)
9. Chen, S., Li, Y.: Automatic sensor placement for model-based robot vision. IEEE Trans. Syst. Man Cybern. Part B Cybern. **33**(1), 393–408 (2004)

10. Chen, X., Davis, J.: Camera placement considering occlusion for robust motion capture. Technical Report CS-TR-2000-07. Stanford University (2000)
11. Cheung, G., Baker, S., Kanade, T. Shape-from-silhouette of articulated objects and its use for human body kinematics estimation and motion capture. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, June 2003
12. Cutler, R., Turk, M.: View-based interpretation of real-time optical flow for gesture recognition. In: Proceedings of the Third IEEE Conference on Face and Gesture Recognition, Nara, Japan, April 1998
13. Denzler, J., Zobel, M., Niemann, H.: On optimal camera parameter selection in Kalman filter based object tracking. In: Proceedings of the 24th DAGM Symposium on Pattern Recognition, pp. 17–25. Zurich, Switzerland, (2002)
14. Fablet, R., Black, M.J.: Automatic detection and tracking of human motion with a view-based representation. In: European Conference on Computer Vision, May 2002
15. Fleishman, S., Cohen-Or, D., Lischinski, D.: Automatic camera placement for image-based modeling. In: Proceedings of Pacific Graphics 99, pp. 12–20 (1999)
16. Gasser, G., Bird, N., Masoud, O., Papanikolopoulos, N.: Human activity monitoring at bus stops. In: Proceedings of the IEEE Conference on Robotics and Automation, April 2004
17. Gerkey, B., Vaughan, R.T., Howard, A.: The player/stage project: Tools for multi-robot and distributed sensor systems. In: Proceedings of the 11th International Conference on Advanced Robotics, pp. 317–323. Coimbra, Portugal, June 2003
18. Grauman, K., Shakhnarovich, G., Darrell, T.: A bayesian approach to image-based visual hull reconstruction. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2003)
19. Haritaoglu, I., Harwood, D., Davis, L.: W4: Real-time surveillance of people and their activities. IEEE Trans. Pattern Anal. Mach. Intell. **22**(8), 809–831 (2000)
20. Isler, V., Kannan, S., Daniilidis, K.: Vc-dimension of exterior visibility. IEEE Trans. Pattern Anal. Mach. Intell. **26**(5), 667–671, May 2004
21. Jackson, B., Bodor, R., Papanikolopoulos, N.P.: Learning static occlusions from interactions with moving figures. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems. Japan, October 2004
22. Masoud, O., Papanikolopoulos, N.P.: Using geometric primitives to calibrate traffic scenes. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems. Japan, October 2004
23. Matusik, W., Buehler, C., Raskar, R., Gortler, S., McMillan, L.: Image-based visual hulls. In: Proceedings of ACM SIGGRAPH, July 2000
24. Maurin, B., Masoud, O., Papanikolopoulos, N.: Monitoring crowded traffic scenes. In: Proceedings of the IEEE International Conference on Intelligent Transportation Systems. Singapore, September 2002
25. McKenna, S., Jabri, S., Duric, Z., Wechsler, H.: Tracking interacting people. In: Proceedings of the Conference on Automatic Face and Gesture Recognition. Grenoble, France, March 2000
26. Mizoguchi, M., Sato, J.: Space-time invariants and video motion extraction from arbitrary viewpoints. In: Proceedings of the International Conference on Pattern Recogition. Quebec, August 2002
27. Mordohai, P., Medioni, G.: Dense multiple view stereo with general camera placement using tensor voting. In: Proceedings of the 2nd International Symposium on 3D Data Processing, Visualization and Transmission (2004)
28. Mori, H., Charkari, M., Matsushita, T. On-line vehicle and pedestrian detections based on sign pattern. IEEE Trans. Ind. Electron. **41**(4) (1994)
29. Nelson, B., Khosla, P.K.: Increasing the tracking region of an eye-in-hand system by singluarity and joint limit avoidance. In: Proceedings of the IEEE International Conference on Robotics and Automation vol. 3, pp. 418–423. IEEE, Piscataway (1993)
30. Nelson, B., Khosla, P.K.: Integrating sensor placement and visual tracking strategies. In: Proceedings of the 1994 IEEE International Conference on Robotics and Automation vol. 2, pp. 1351–1356. IEEE, Piscataway (1994)
31. Nelson, B., Khosla, P.K.: The resolvability ellipsoid for visual servoing. In: Proceedings of the 1994 IEEE Conference on Computer Vision and Pattern Recognition, pp. 829–832. IEEE, Piscataway (1994)
32. Nicolescu, M., Medioni, G.: Motion segmentation with accurate boundaries—a tensor voting approach. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. June 2003

33. Niem, W.: Error analysis for silhouette-based 3D shape estimation from multiple views. In: Proceedings of the International Workshop on Synthetic-Natural Hybrid Coding and Three Dimensional Imaging (1997)
34. Olague, G., Mohr, R.: Optimal 3D sensor placement to obtain accurate 3D point positions. In: Proceedings of the Fourteenth International Conference on Pattern Recognition vol. 1, pp. 16–20, August 1998
35. O'Rourke, J.: Art Gallery Theorems and Algorithms. Oxford University Press, New York (1987)
36. Parameswaran, V., Chellappa, R.: View invariants for human action recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, June 2003
37. Pless, R.: Image spaces and video trajectories: using isomap to explore video sequences. In: Proceedings of the International Conference on Computer Vision, pp. 1433–1440 (2003)
38. Rosales, R., Sclaroff, S.: 3D trajectory recovery for tracking multiple objects and trajectory guided recognition of actions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, June 1999
39. Roy, S., Chaudhury, S., Banerjee, S.: Active recognition through next view planning: a survey. Pattern Recogn. **37**, 429–446 (2004)
40. Scott, W., Roth, G., Rivest, J.-F.: View planning for automated three-dimensional object reconstruction and inspection. Comput. Surv. **35**(1), 64–96, March 2003
41. Sharma, R., Hutchinson, S.: Motion perceptibility and its application to active vision-based servo control. IEEE Trans. Robot. Autom. **13**(4), 607–617 (1997)
42. Stauffer, C., Tieu, K.: Automated multi-camera planar tracking correspondence modeling. In: Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition, vol. 1., June 2003
43. Tarabanis, K.A., Allen, P., Tsai, R.Y.: A survey of sensor planning in computer vision. IEEE Trans. Robot. Autom. **11**(1), 86–104, February 1995
44. Tarabanis, K.A., Tsai, R.Y., Allen, P.: Automated sensor planning for robotic vision tasks. In: Proceedings of the IEEE International Conference on Robotics and Automation, pp. 76–82, April 1991
45. Tarabanis, K.A., Tsai, R.Y., Allen, P.: The mvp sensor planning system for robotic vision tasks. IEEE Trans. Robot. Autom. **11**(1), 72–85, February 1995
46. Tarabanis, K.A., Tsai, R.Y., Kaul, A.: Computing occlusion-free viewpoints. IEEE Trans. Pattern Anal. Mach. Intell. **18**(3), 273–292, March 1996
47. Tarbox, G., Gottschlich, S.: Planning for complete sensor coverage in inspection. Comput. Vis. Image Underst. **61**(1), 84–111, January 1995
48. Ukita, N., Matsuyama, T.: Incremental observable-area modeling for cooperative tracking. In: Proceedings of the International Conference on Pattern Recognition, September 2000
49. Weik, S., Liedtke, C.: Hierarchical 3D pose estimation for articulated human body models from a sequence of volume data. In: Proceedings of the International Workshop on Robot Vision, pp. 27–34 (2001)
50. Wong, K., Chang, M.: 3d model reconstruction by constrained bundle adjustment. In: Proceedings of the International Conference on Pattern Recognition, August 2004
51. Wren, C., Azarbayejani, A., Darrell, T., Pentland, A.: Pfinder: Real-time tracking of the human body. IEEE Trans. Pattern Anal. Mach. Intell. **19**(7), 780–785, July 1997
52. Yao, Y., Allen, P.: Computing robust viewpoints with multi-constraints using tree annealing. In: IEEE International Conference on Systems, Man, and Cybernetics, vol. 2, pp. 993–998. IEEE, Piscataway (1995)