Multi-Camera Positioning to Optimize Task Observability

Robert Bodor, Paul Schrater, and Nikolaos Papanikolopoulos
Department of Computer Science and Engineering
University of Minnesota
Minneapolis, Minnesota, 55455
{*rbodor,schrater,npapas*}*@cs.umn.edu*

Abstract

The performance of computer vision systems for measurement, surveillance, reconstruction, gait recognition, and many other applications, depends heavily on the placement of cameras observing the scene. This work addresses the question of the optimal placement of cameras to maximize the performance of real-world vision systems in a variety of applications. Specifically, 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 multi-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 multi-camera systems in real-world monitoring applications, both indoor and outdoor.

I. Introduction

Resolution is one of the fundamental and primary information bottlenecks for vision applications. The performance of virtually all 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.

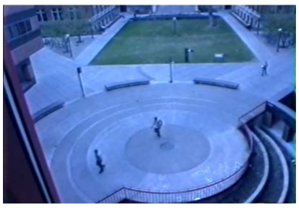


Fig. 1. Scene of people moving outdoors.

For example, if we desire to monitor an area effectively with a vision system (Figure 1), the system may be required to perform a variety of tasks such as general surveillance (detection of loitering behavior, detection of unusual behaviors, detection of abandoned objects, monitoring of crowd movements, etc.), subject tracking [7], activity classification ([2]), gesture recognition [5], along with individual biometrics gathering (face recognition, 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 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 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.

II. Related Work

Proper camera placement for the purpose of optimizing the sensor's ability to capture information about a desired environment or task has been considerably studied. In [16], O'Rourke 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). Fleishman *et al.* further refined the art gallery framework by introducing a resolution quality metric [8]. 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 [9]. Our method differs from the art gallery framework in several important ways. We focus on task observability, and try to capture target

0-7803-9385-6/05/$20.00 ©2005 IEEE

motions. In addition, we do not optimize for minimum coverage, and in fact, to observe these tasks, overlapping camera views are often necessary.

In the field of robotics, vision sensor planning has been studied to aid in task planning and visual servoing tasks. In [1], 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. Nelson and Khosla [12] 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 dynamic 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 ([13] and [14]). Sharma and Hutchinson also introduce a quantitative sensory measure, perceptibility, in order to improve positioning and control of manipulator systems [18].

In [19], Tarabanis *et al.* present a planning method to determine optimal camera placement given task-specific observational requirements such as field of view, visibility, and depth of field. In addition, Yao and Allen [20], 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 [15] 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. In [4], Chen and Davis develop a resolution metric for camera placement considering occlusions. In addition, Denzler *et al.* [6] 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.

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 are not known a priori, and change with time as the subjects move through the scene.

III. Problem Forumlation

A. 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 privileged points $\vec{x}_1(t), \ldots, \vec{x}_n(t)$, represented by a state vector $X(t) = \left[\begin{array}{ccc} \vec{x}_1(t)^T & \cdots & \vec{x}_n(t)^T \end{array} \right]^T$. The distribution of target motions is defined over the ensemble of state vector trajectories $Y_i = \{X(1), \ldots, X(t)\}_i$ where Y_i denotes the i^{th} trajectory in the ensemble. Let $Y = f(\vec{s})$ denote a parametric description of the trajectories.

Finally, 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:

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

where $G\left(\vec{s}, \vec{u}_1, \ldots, \vec{u}_n\right)$ represents a gain function on trajectory states and camera parameters, $\vec{u}_i$ are the parameters for the i^{th} camera and $p\left(\vec{s}\right)$ is the prior probability distribution on the trajectory states. Our system assumes that $p\left(\vec{s}\right)$ 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 sample paths with parameters $\vec{s}_j$, the value function can be approximated:

$$V\left(\vec{u}_1, \ldots, \vec{u}_n\right) = \sum_{j}^{samples} G\left(\vec{s}, \vec{u}_1, \ldots, \vec{u}_n\right). \qquad (2)$$

The choice of a parametrization $\vec{s}_j$ 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 decribed in [3]. 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, in [11], Mizoguchi and Sato 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 [17].

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:

$$\vec{s}_j = \left[\begin{array}{cccc} \phi_j & x_j & y_j & l_j \end{array} \right]^T \qquad (3)$$

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.

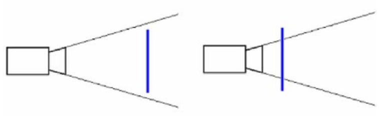
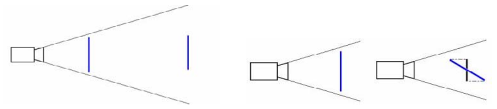


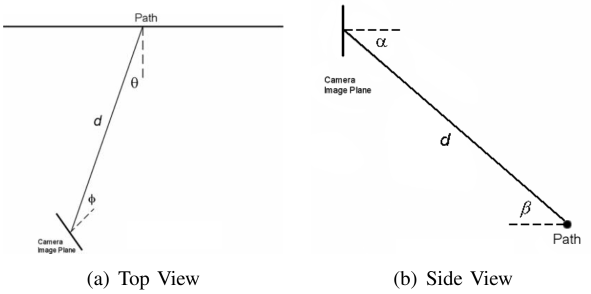
Fig. 2. Effect of $d_{ij} \geq d_o$.



(a) Increasing object-camera distance.

(b) Increasing foreshortening.

Fig. 3. Configurations that decrease observability in pinhole projection cameras.



(a) Top View

(b) Side View

Fig. 4. Variables to minimize for each path.

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.

B. *Specific Formulation*

1) 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, requires the camera to be far enough away from the subject that the entire motion is captured within the camera field of view (Figure 2).

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 3 depicts the reason for this. For a perspective 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 3(b) depicts this situation. As the angle between the camera's view direction and the object decreases, the projection of the object on 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:

$$resolution \propto \frac{1}{d^2} \tag{4}$$

$$resolution \propto cos\left(\theta\right) \tag{5}$$

Thus, optimizing path observability for an individual path corresponds to minimizing the distance between the camera and the path center (d_{ij}), along with minimizing foreshortening effects. In this case, there are two sources of foreshortening: the angle between the path normal and the camera position (θ_{ij}, β_{ij}), and the angle between the path center and the normal to the image plane (ϕ_{ij}, α_{ij}) (Figure 4).

However, the camera must maintain a minimum distance, d_0, from each path to ensure that the full motion sequence is in view (Figure 3):

$$d_0 = \frac{r_a l_j f}{w} \tag{6}$$

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.

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

$$G_{ij} = \frac{d_0^2}{d_{ij}^2} cos\left(\theta_{ij}\right) cos\left(\phi_{ij}\right) cos\left(\alpha_{ij}\right) cos\left(\beta_{ij}\right). \tag{7}$$

Optimizing this function over camera parameters will solve the observability problem for the single camera, single path case.

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

$$V = \sum_{j}^{paths} G_j \tag{8}$$

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 unitless 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.

4) 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 ($O\left(V\right) = \left(km\right)^n$, 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). Thus, for practicality of application, we instead employ an iterative method that closely approximates the optimal approach.

We do this by defining a vector of path observabilities per camera G_i, where each element, G_{ij}, describes the observability of path j by camera i: $G_i = [G_{i1}, \dots, G_{in}]$. In addition, we define constant vectors $G_0 = [0, \dots, 0]$ and $I = [1, \dots, 1]$. By inverting the observability values of the previous camera $(I - G_{k-1})$, the current camera is directed to the regions of the path distribution that have the lowest observability so far. This iterative process works for any number of cameras, and has a computational complexity that is linear ($O\left(V\right) = \left(kmn\right)$).

$$V = \sum_{i}^{cams} \left[\sum_{j}^{paths} \left[\prod_{k=1}^{i} \left(I - G_{k-1j}\left(\vec{u}_k\right)\right) \right] G_{ij}\left(\vec{u}_i\right) \right] \quad (9)$$

Equation (9) gives the total observability value function for the multi-camera system. Note that while this equation is iterative, the order of camera selection does not change the result.

C. Optimization

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

$$\vec{u}_i = \left[\begin{array}{ccccccc} X_{c_i} & Y_{c_i} & Z_{c_i} & \gamma_{x_{c_i}} & \gamma_{y_{c_i}} & \gamma_{z_{c_i}} & f \end{array}\right]^T \quad (10)$$

where $(X_{c_i}, Y_{c_i}, Z_{c_i})$ correspond 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 formulation may be applied to many vision problems. We chose to focus first on the problem of camera placement to maximize general observability of 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 fixed field of view cameras, thus f is held constant.

Our objective function is thus reduced to:

$$G_{ij} = \frac{d_0^2}{d_{ij}^2} cos\left(\theta_{ij}\right) cos\left(\phi_{ij}\right) \quad (11)$$

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

$$\vec{u} = \left[\begin{array}{ccc} X_{c_i} & Y_{c_i} & \gamma_{z_{c_i}} \end{array}\right]^T. \quad (12)$$

Equations (13) through (15) 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.

$$d_{ij} = \sqrt{\left(x_j - X_{c_i}\right)^2 + \left(y_j - Y_{c_i}\right)^2} \quad (13)$$

$$\theta_{ij} = cos^{-1}\left(\frac{T_1 + T_2}{T_3 \cdot T_4}\right) \quad (14)$$

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.

$$\phi_{ij} = cos^{-1}\left(\frac{P_1 - P_2}{P_3 \cdot P_4}\right) \quad (15)$$

where

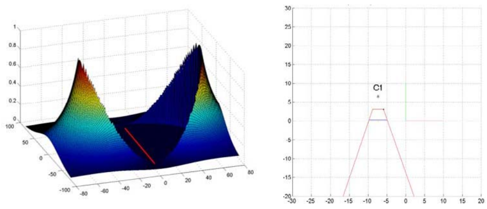
$$\begin{aligned} P_1 &= \left(y_j - Y_{c_i}\right) cos\left(\gamma_{z_{c_i}}\right) cos\left(\gamma_{x_{c_i}}\right) \\ P_2 &= \left(x_j - X_{c_i}\right) sin\left(\gamma_{z_{c_i}}\right) cos\left(\gamma_{x_{c_i}}\right) \\ P_3 &= \sqrt{cos^2\left(\gamma_{z_{c_i}}\right) cos^2\left(\gamma_{x_{c_i}}\right) + sin^2\left(\gamma_{z_{c_i}}\right) cos^2\left(\gamma_{x_{c_i}}\right)} \\ P_4 &= \sqrt{\left(x_j - X_{c_i}\right)^2 + \left(y_j - Y_{c_i}\right)^2}. \end{aligned}$$

We solve for the parameters $(X_{c_i},\ Y_{c_i},\ \gamma_{z_{c_i}})$ through a iterative-refinement based constrained nonlinear optimization process. We evaluate the constrained function 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. We have found this process to work very well, and is faster than Newton-Raphson methods in the presence of complex sets of constraints.

IV. Simulation Results

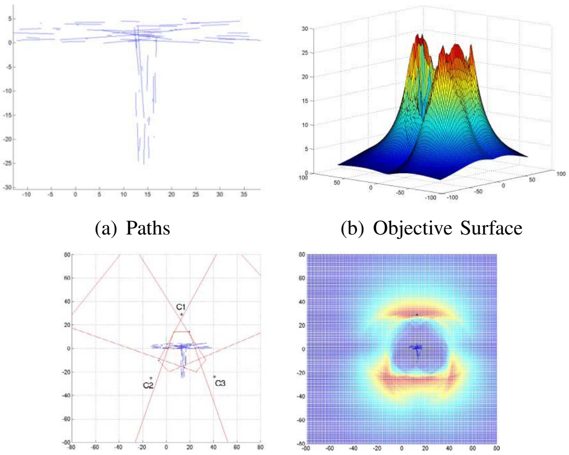
Figures 5 and 6 show the results of the method on simulated path distributions to illustrate the approach. In all cases, the objective surface shown corresponds to the 2D slice of $\gamma_{z_{c_i}}$ where the objective function is optimized.

Figure 5 shows results for the case of a single camera and single path. Note how the objective surface varies with position. The function has two optimum: one on either side of the path. The simulation result confirms the intuition that the optimal camera position to observe a



(a) Objective Surface (b) Solution

Fig. 5. Objective surface and camera placement solution for single path, single camera case. Labeled dots indicate camera locations. In 5(b), red lines indicate the view frustum of the camera.



(a) Paths (b) Objective Surface

(c) Multiple Camera Solution (d) Objective Surface (from above)

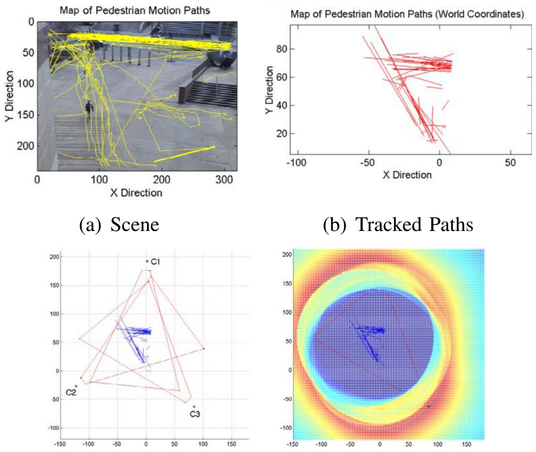
Fig. 6. Results for a simulated traffic intersection path configuration. Labeled dots indicate camera locations. Camera placement results are shown for 3 cameras, along with the initial objective surface.

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. This suggests a sufficiency condition on the number of cameras necessary to completely observe any path distribution may be determined via this approach.

Figure 6 shows results for a more complicatd path distribution, similar to one that might be observed at a traffic intersection.

V. 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 segmentation algorithm. In addition, a Kalman filter was used for tracking. Linear



(a) Scene (b) Tracked Paths

(c) Multiple Camera Solution (d) Objective Surface (from above)

Fig. 7. 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 the third camera.

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 [10]. Thus, all path distributions shown are in world coordinates, as are the camera placement solutions.

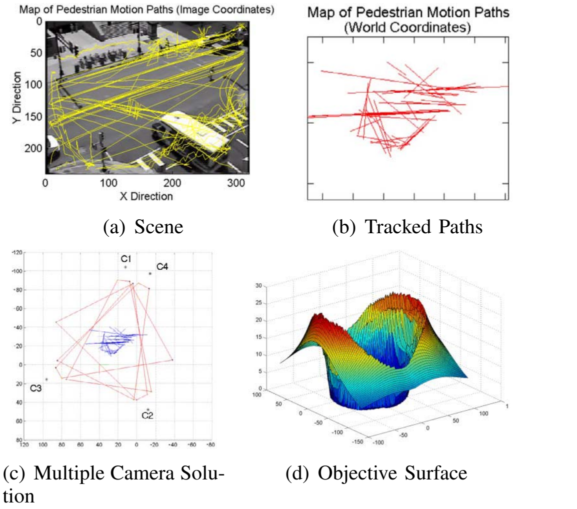
Figure 7 shows results for an outdoor pedestrian courtyard and street scene. Here the first two cameras focus attention on the dominant pathways, while the third camera attends to the contribution of the less common paths. Figure 8 shows results for a traffic intersection. Note that the camera placement solution is very similar in nature to that predicted by the simulated intersection.

VI. Conclusion

We have presented 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. This approach was validated in both simulation and real-world experiments involving multiple camera systems observing many subjects moving through a scene. Our findings indicate that this method may be used to guide proper camera placement in complex and busy scenes.

VII. Future Work

Our goal is to develop a camera placement design tool based on this approach. We plan to extend this work to optimize over all extrinsic camera parameters. Furthermore, to provide for camera placement in arbitrary



(a) Scene

(b) Tracked Paths

(c) Multiple Camera Solution

(d) Objective Surface

Fig. 8. Results for a traffic intersection path configuration. Labeled dots indicate camera locations. Camera placement results are shown for 4 cameras, along with the initial objective surface.

real-world scenes, we intend to incorporate static occlusion constraints into the observability formulation.

In addition, we plan to investigate the possibility of fast optimization of the objective function via basis function approximations to the value surface.

Acknowledgments

This work was supported by the National Science Foundation through grant #IIS-0219863, #CNS-0224363, #CNS-0324864, and #CNS-042083.

References

[1] S. Abrams, P. K. Allen, and K. A. Tarabanis. Dynamic sensor planning. In *IEEE International Conference on Intelligent Autonomous Systems*, pages 206–215, Pittsburgh, PA, Feb 1993.

[2] J. Ben-Arie, Z. Wang, P. Pandit, and S. Rajaram. Human activity recognition using multidimensional indexing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(8):1091–1104, Aug 2002.

[3] C. Bregler and J. Malik. Tracking people with twists and exponential maps. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, June 1998.

[4] X. Chen and J. Davis. Camera placement considering occlusion for robust motion capture. Technical Report CS-TR-2000-07, Stanford University, 2000.

[5] R. Cutler and M. Turk. 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, Apr 1998.

[6] J. Denzler, M. Zobel, and H. Niemann. On optimal camera parameter selection in Kalman filter based object tracking. In *Proceedings of the 24th DAGM Symposium on Pattern Recognition*, pages 17–25, Zurich, Switzerland, 2002.

[7] R. Fablet and M. J. Black. Automatic detection and tracking of human motion with a view-based representation. In *European Conference on Computer Vision*, May 2002.

[8] S. Fleishman, D. Cohen-Or, and D. Lischinski. Automatic camera placement for image-based modeling. In *Proceedings of Pacific Graphics 99*, pages 12–20, 1999.

[9] V. Isler, S. Kannan, and K. Daniilidis. Vc-dimension of exterior visibility. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(5):667–671, 2004.

[10] O. Masoud and N. P. Papanikolopoulos. Using geometric primitives to calibrate traffic scenes. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Japan, Oct 2004.

[11] M. Mizoguchi and J. Sato. Space-time invariants and video motion extraction from arbitrary viewpoints. In *Proceedings of the International Conference on Pattern Recognition*, Quebec, August 2002.

[12] B. Nelson and P. K. Khosla. 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*, volume 3, pages 418–423, 1993.

[13] B. Nelson and P. K. Khosla. Integrating sensor placement and visual tracking strategies. In *Proceedings of the 1994 IEEE International Conference on Robotics and Automation*, volume 2, pages 1351–1356, 1994.

[14] B. Nelson and P. K. Khosla. The resolvability ellipsoid for visual servoing. In *Proceedings of the 1994 IEEE Conference on Computer Vision and Pattern Recognition*, pages 829–832, 1994.

[15] G. Olague and R. Mohr. Optimal 3d sensors placement to obtain accurate 3d points positions. In *Proceedings of the Fourteenth International Conference on Pattern Recognition*, volume 1, pages 16–20, Aug 1998.

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

[17] V. Parameswaran and R. Chellappa. View invariants for human action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, June 2003.

[18] R. Sharma and S. Hutchinson. Motion perceptibility and its application to active vision-based servo control. *IEEE Trans. on Robotics and Automation*, 13(4):607–617, 1997.

[19] K. A. Tarabanis, R. Y. Tsai, and A. Kaul. Computing occlusion-free viewpoints. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 18(3):273–292, March 1996.

[20] Y. Yao and P. Allen. Computing robust viewpoints with multi-constraints using tree annealing. *IEEE Intl. Conf. on Systems, Man, and Cybernetics*, 2:993–998, 1995.