Mobile Camera Positioning to Optimize the Observability of Human Activity Recognition Tasks *

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

***Abstract*— The performance of systems for human activity recognition 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 these types of recognition tasks. Specifically, our goal is to optimize the quality of the joint 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. A nonlinear optimization approach is used to find the internal and external (mounting position and orientation) camera parameters that optimize the recognition criteria. In these experiments, a single camera is repositioned using a mobile robot. Initial results for the problem of human activity recognition are presented.**

***Index Terms*— Observability, human activity monitoring, tracking, patrol robotics.**

I. Introduction

Recognizing human activity is a very important and active area in computer vision, with applications in human-computer interaction, user interfaces, robot learning, and surveillance, among others.

However, the development of systems that can robustly recognize human behavior in real-world environments remains challenging. People are free to move throughout an area of interest in any direction they like (Figure 1). As a result, positioning a camera to effectively observe the activity of interest is a difficult problem. A single camera position has to take into consideration the observability of all activities in order to optimize the recognition performance.

Our goal is to address the problem of camera placement to optimize the joint 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. This application 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.

Another application of this research is for active vision. For unknown or changing path distributions, it is desirable to be able to move the cameras online, so as to continually optimize camera pose to match the new path configuration.

In this paper, we present an initial experiment that demonstrates such an active vision application. Specifically, we consider the task of human motion recognition. This task defines our observability goal to be: observing the entire path of motion while maximizing the view of the subject in the image.

Observations of this type may be used as input image sequences for a wide variety of human activity recognitions systems such as gait recognition and other articulated motion recognition systems. This field of study applies pattern recognition principles to images of human motion to categorize the articulated motion that is taking place (walking, running, etc.) ([2], [3], [5], [7], [9], [11], [22]). Gait recognition is sometimes considered a category of motion recognition, focusing on walking as a biometric feature. These classifiers generally require input image sequences showing the entire motion path of the subject from a single viewpoint. Moreover, some views of the motion will better disambiguate the activity than others, and are thus better suited for recognition.

Our observability formulation and optimization procedure can find the optimal camera location and pose to observe these motion sequences and provide this input.

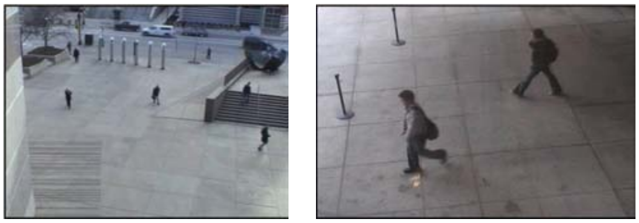


Fig. 1. Two sample scenes of people walking in real-world settings.

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 studied considerably. In [17], O'Rourke provides an in-depth theoretical analysis of the problem of maximizing camera coverage of an area, where

[*] 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.

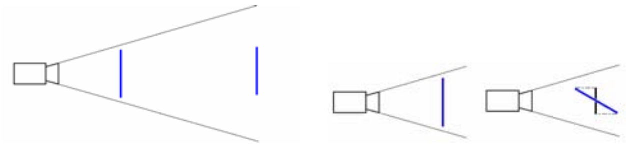
the camera fields of view do not overlap (the so-called "art gallery" problem). Fleishman, Cohen-Or and Lischinski further refined the art gallery framework by introducing a resolution quality metric [8].

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 [13] 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 ([14] and [15]). 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 [21], formulate the problem of sensor placement as an unconstrained optimization problem, and apply tree-annealing to compute optimal camera viewpoints in the presence of noise. Olague and Mohr [16] 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 occlusion. 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. Formulation of Path Observability Metric

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. 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 2(a) 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 ((1) and (2)).



(a) Increasing object-camera distance.

(b) Increasing foreshortening.

Fig. 2. Configurations that decrease observability in pinhole projection cameras.

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

$$resolution \propto \cos(\theta) \,. \tag{2}$$

Another factor that reduces observability of an object is foreshortening. Figure 2(b) depicts this situation. As the angle between the camera's view direction and the object decreases, the projection of the object in camera's image plane also decreases. As a result, the number of pixels that measure the object decreases, and the observability lessens.

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 (Figure 3(b)).

The camera must maintain a minimum distance, d_0, (4) from each path to ensure that the full motion sequence is in view:

$$d_0 = \frac{r_a l_j f}{w} \tag{3}$$

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.

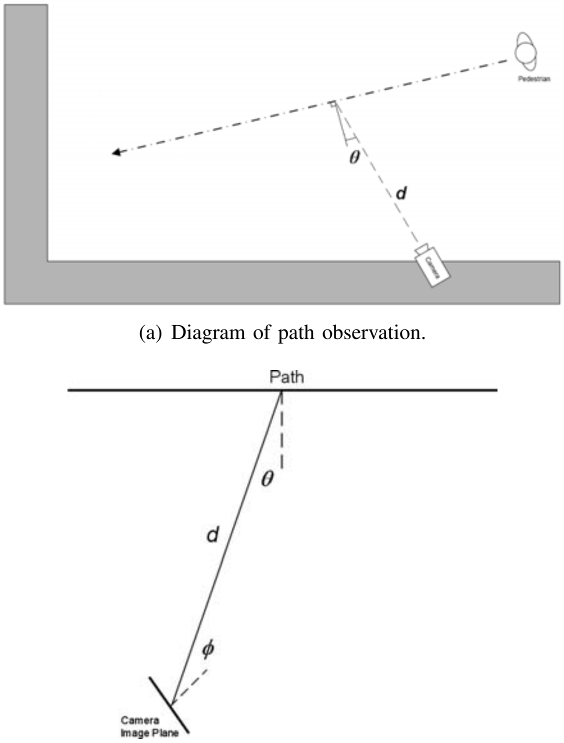
$$d_{ij} \geq d_0 \tag{4}$$

In addition, for multi-path observability, all paths must lie completely within the view frustum of the camera. These two constraints are coupled, as they both depend on the field of view of the camera, yet they are not equivalent. The first constraint applies to each camera-path pair, while the second constraint applies to the path distribution as a whole.

Thus, joint path observability, considering the entire path distribution and multi-camera system, can be described as follows:

$$O_P = \sum_{i}^{cams} \sum_{j}^{paths} \frac{d_0^2}{d_{ij}^2} cos\left(\theta_{ij}\right) cos\left(\phi_{ij}\right) . \tag{5}$$

Note that this is a unitless metric. If O_P is multiplied by the height and width of the image in pixels, it becomes a pixel resolution metric of observability, as discussed above.



(a) Diagram of path observation.

(b) Variables to minimize for each path.

Fig. 3. Camera placement relative to motion path.

IV. Optimization

In order to use this formulation as a tool for camera placement, it is first necessary to change our variables from $(d_{ij}, \theta_{ij}, \phi_{ij})$, which describe the relative position and orientation of the camera to the path, to those that describe the absolute position and orientation of the camera (and path) in the world $(X_{c_i}, Y_{c_i}, \gamma_{z_{c_i}})$. Here (X_{c_i}, Y_{c_i}) correspond to camera position, and $\gamma_{z_{c_i}}$ corresponds to camera orientation (yaw) about the Z axis. We parameterize each motion path as follows:

$$\vec{s}_j = \left[\begin{array}{ccccc} \phi_j & x_{sj} & y_{sj} & x_j & y_j \end{array}\right]^T \tag{6}$$

where ϕ_j is the orientation of the path j, (x_j, y_j) defines the path center, and $(x_{sj},\ y_{sj})$ is the starting point of each path.

Equations (7), (8), and (9) show $(d_{ij}, \theta_{ij}, \phi_{ij})$in terms of $(X_{c_i}, Y_{c_i}, \gamma_{z_{c_i}})$ for each camera-path pair. When Equations (7), (8), and (9) are substituted into (5), the value of the observability objective function can be determined for a given camera position and pose in world coordinates.

$$d_{ij} = \sqrt{\left(x_j - X_{c_i}\right)^2 + \left(y_j - Y_{c_i}\right)^2} \tag{7}$$

$$\theta_{ij} = cos^{-1}\left(\frac{T_1 + T_2}{T_3 \cdot T_4}\right) \tag{8}$$

where

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

$$\phi_{ij} = cos^{-1}\left(\frac{P_1 - P_2}{P_3 \cdot P_4}\right) \tag{9}$$

where

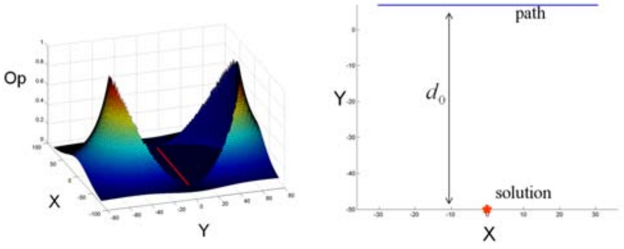
$$\begin{aligned} P_1 &= \left(y_j - Y_{c_i}\right) cos\left(\gamma_{z_{c_i}}\right) cos\left(\gamma_{x_{c_i}}\right) \\ P_2 &= \left(x_j - X_{c_i}\right) sin\left(\gamma_{z_{c_i}}\right) cos\left(\gamma_{x_{c_i}}\right) \\ P_3 &= \sqrt{cos^2\left(\gamma_{z_{c_i}}\right) cos^2\left(\gamma_{x_{c_i}}\right)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

We solve for the parameters $(X_{c_i},\ Y_{c_i},\ \gamma_{z_{c_i}})$ through a constrained nonlinear optimization process.

V. Simulation Results

We have implemented the above formulation in Matlab. We use the built-in function `fmincon` as the constrained nonlinear optimizer. The optimizer is seeded with an initial guess that is determined by sampling the objective surface.

Figure 4(a) shows the objective surface for the single path case. The objective function increases as the position approaches the path, up to the distance d_0. (The objective function is uniformly zero for all positions within d_0 of the path center.) In addition, the objective function decreases as the position moves away from perpendicular to the path.



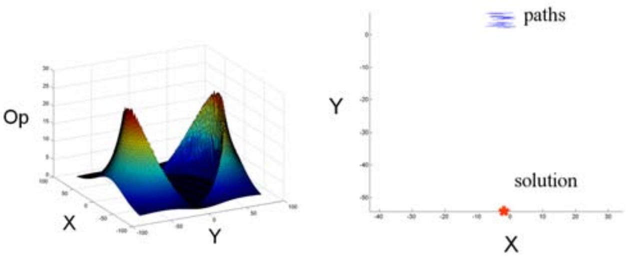
(a) Objective Function.

(b) Solution.

Fig. 4. Objective function and solution for single path optimization.

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 single path is perpendicular to that path, at a distance of d_0.

Note that the multiple path objective surface in Figure 5(a) looks very similar to the single path case. In this case, the joint objective surface can be thought of as a sum of the individual objective surfaces stacked together. Also note that the height of the objective surface is much greater than for the single path case.



(a) Objective Function.
(b) Solution.

Fig. 5. Objective function and solution for multiple path optimization in the case when all paths are parallel.

VI. Experimentation

A. Equipment Setup

In this paper, we present an initial experiment that demonstrates an active vision application of this work. 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 position to view this path distribution, and moves to the new goal position. Figure 6 shows the arrangement of the experiment.

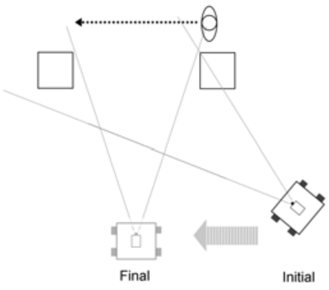
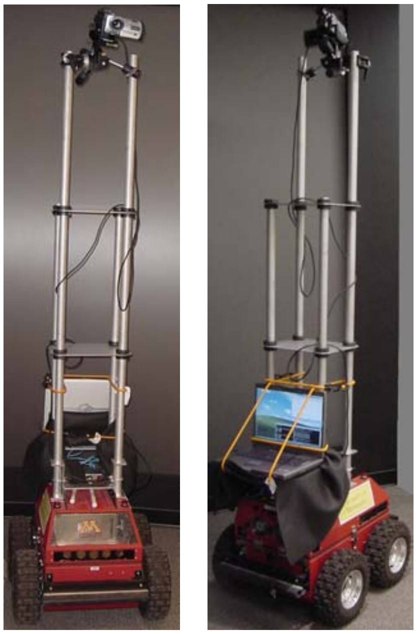


Fig. 6. Overview of the experiment.

This experiment makes use of a modified ATRV-JR robot, shown in Figure 7(a). 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 onboard systems.

The experiment required the integration of several systems as shown in the diagram in Figure 8. Here, a firewire camera (Sony DCR-TRV730) is used to provide a video stream to a software module that captures video and tracks movement.



(a) Front View.
(b) Side View.

Fig. 7. The robotic setup consisting of an ATRV-JR and a "sensor mast".

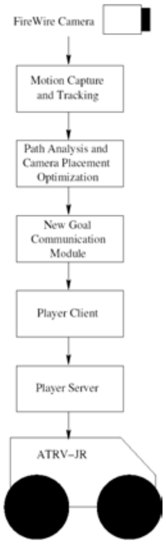

FireWire Camera
Motion Capture and Tracking
Path Analysis and Camera Placement Optimization
New Goal Communication Module
Player Client
Player Server
ATRV-JR

Fig. 8. Block diagram ATRV-JR camera system.

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 Figure 9. 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 [20]. However, any resonably reliable tracker would work well for this.

The tracking information is then passed to a Matlab program which extracts path trajectories, estimates the parameters of each motion path, and uses the method described in Section III to compute the optimal camera pose. Figure 10 illustrates the paths estimated from the position tracking data. This is done using a linear least-squares fitting. The new position is then transmitted as a location goal to the robot. A player-client [10] 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 to continually keep the camera in a location that provides the optimal observability of the motion sequences.

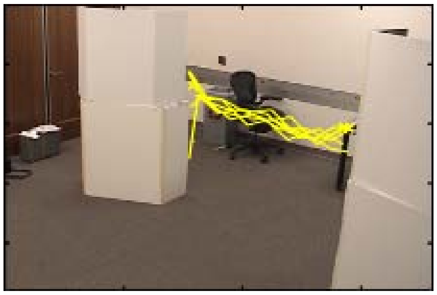
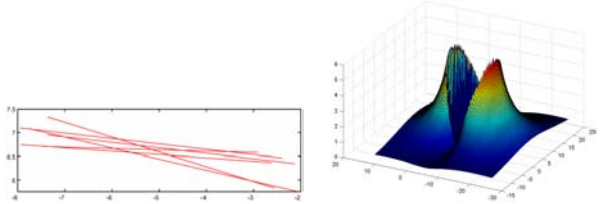


Fig. 9. Paths of motion tracked in the image.



(a) Path lines projected onto the ground plane (world coordinates).

(b) Objective Surface.

Fig. 10. Path lines and resultant objective surface.

B. 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 *et al.* [12]. The calibration involved selecting parallel and perpendicular lines in the ground plane in an image (Figure 11). The result was a homography transformation matrix between the camera's image plane and the ground plane.

C. Experimental Results

Figures 13 and 14 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 (Figure 12(b)) of the walking motion has significantly improved observability from the initial view (Figure 12(a)).

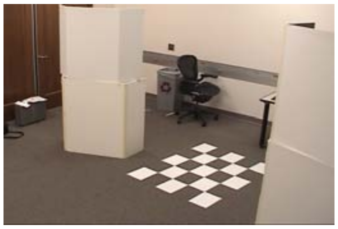
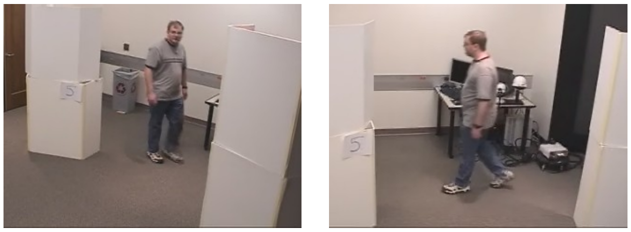


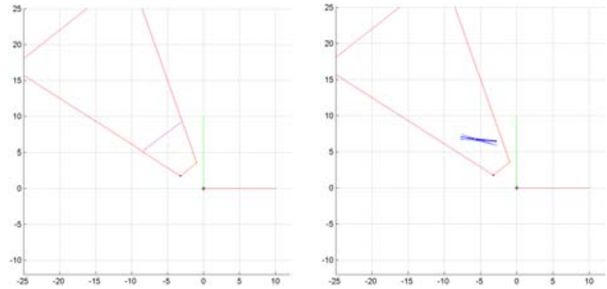
Fig. 11. Calibration setup. The calibration grid shown here was removed for experiments.



(a) Initial View.

(b) Final View.

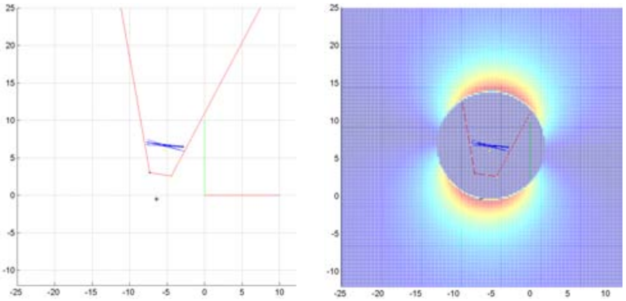
Fig. 12. Initial and final view of the camera during a run.



(a) Path configuration at initialization. Camera is positioned to optimize the view of this single path.

(b) Observed path configuration relative to camera pose.

Fig. 13. Initial camera positioning and view of paths. 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. 14. Camera positioning solution for joint path observability optimization. Figures show camera view frustum projected onto ground plane.

Table I 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 (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.

TABLE I
OBSERVABILITY RESULTS.

	Trial					
Observability	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

VII. CONCLUSIONS

We have presented a novel analytical method for computing optimal camera position and pose for task observability. The general formulation was applied to the problem of human activity recognition. This approach was validated in both simulation and real-world experiments involving a single camera observing a subject moving through a scene. Our findings indicate that this method can significantly improve task observability for human motion studies, and may be used to guide proper camera placement in dynamically adapting scenes.

VIII. FUTURE WORK

There are many areas in which this work can be extended. One area of particular interest is the coordination of multiple robotic systems used to optimally place multiple cameras. In addition, we plan to extend the formulation to study observability in the presence of occlusions. We also plan to test our approach in many other application areas such as traffic monitoring, vision-based user interface design, and indoor and outdoor surveillance.

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] D. Cao, O. Masoud, D. Boley, and N. Papanikolopoulos. Online motion classification using support vector machines. In *IEEE International Conference on Robotics and Automation*, Apr 2004.

[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] D. M. Gavrila. The visual analysis of human movement: A survey. *Computer Vision and Image Understanding*, 73(1):82–98, 1999.

[10] B. Gerkey, R. T. Vaughan, and A. Howard. The player/stage project: Tools for multi-robot and distributed sensor systems. In *Proceedings of the 11th International Conference on Advanced Robotics*, pages 317–323, Coimbra, Portugal, Jun 2003.

[11] O. Masoud and N. P. Papanikolopoulos. A method for human action recognition. *Image and Vision Computing*, 21(8):729–743, 2003.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] 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.

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

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

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

[20] H. Veeraraghavan, O. Masoud, and N. Papanikolopoulos. Computer vision algorithms for intersection monitoring. *IEEE Transactions on Intelligent Transportation Systems*, 4(2):78–89, June 2003.

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

[22] L. Zhao. Dressed human modeling, detection, and parts localization. Technical Report CMU-RI-TR-01-19, Robotics Institute, Carnegie Mellon University, Jul 2001.