On Viewpoint Control*

S. Uppala D. R. Karuppiah M. Brewer
S. Chandu Ravela R. A. Grupen
Laboratories for Perceptual Robotics and Computer Vision
Computer Science Department, University of Massachusetts, Amherst, MA 01003
Email: {ravela,uppala,deepak,grupen}@cs.umass.edu

Abstract

In this paper a reactive and concurrent control framework for viewpoint control is developed. The viewpoint control task is decomposed into three control objectives namely; obstacle avoidance, visibility and precision. A moving object is tracked in two panoramic sensors using color, and a scalar uncertainty metric of the object position estimate is introduced. Individual control objectives are accomplished by planning paths using harmonic functions and the task is accomplished using a new *subject-to* composition operator. It is shown that the system is stable under this composition. The system is demonstrated for tracking a human subject using a fixed panoramic sensor and another panoramic sensor mounted on a mobile platform.

1 Introduction

The objective of this paper is to demonstrate a reactive and concurrent control framework for *viewpoint control*. In the context of this paper, viewpoint control is defined as the ability of mobile robots to acquire and maintain a well-conditioned kinematic configuration between themselves and a moving target. As such, viewpoint control is essential for several applications.

For example, an essential element of the emerging *smart rooms* application [1, 9, 16] is the ability of a sensor network to locate and track mobile objects (people, robots, etc). Since indoor environments are populated by several occluding features (such as furniture, partitions, walls) an approach that utilizes a large number of sensors (articulated or stationary) to cover the free space is, often, an expensive proposition. Instead mobile sensors can offer a significant advantage. Mobile platforms, with appropriate sensor payloads, can be used with other mobile or stationary sensor(s) to estimate position of moving objects and, in doing so, can provide sensor services across a large area of space.

Mobile robot teams that search environments for mapping, or for locating objects of interest [5], is another example of viewpoint control. Coordination between robots, which is essential to achieve the search task, assumes that the robots can maintain desired kinematic configurations. In the absence of external sensors to measure robot positions the task of localization falls on the robot themselves while simultaneously accomplishing the team objectives. This calls for some (at least two) of the team members to perform viewpoint control.

Irrespective of whether the target is a leader being triangulated by two others on the team or a person being tracked by an ensemble of mobile and stationary sensors in a smart environment, the mobile observer(s) will have to *keep up* with the target. That is, the mobile observer, in general, will have to plan trajectories that are safe (obstacle avoiding), correct (visibility at all times from target), and converge to a vantage point (kinematically *well-conditioned* configuration).

In this paper, the approach to viewpoint control consists of three steps. First, the task is decomposed into three control objectives namely *obstacle avoidance, visibility, and precision*. Second, individual control objectives are accomplished using harmonic function path planners with boundary conditions appropriate for the objective. Finally, the task is accomplished by composing individual controllers into a stable (in the sense of Lyapunov) task controller. This approach avoids inherent complexities that arise when building a monolithic task controller. Further, individual controllers are reactive, and can handle changes in the environment while the task is being executed [10, 11]. Finally, control composition is done using a systematic framework that is provably stable in contrast to other regimes [4, 7, 18].

Our approach is demonstrated with experiments on a system that consists of a mobile platform with a panoramic camera, a stationary panoramic camera, and a moving target. The objective is to track this moving target position on the ground plane (x, y), and position a robot at a good viewpoint. The contributions of the proposed approach are the following: First, visual correspondence between object features in multiple panoramic cameras using color. Second, a scalar precision metric for a target and observer pair configuration is introduced and used for viewpoint control. Third, a velocity control law that performs well in the presence of shallow gradients typical in harmonic potential fields (with Dirichlet boundary conditions) is introduced. This law is shown to yield asymptotic stability in the

*This work is funded in part by the following grants. DARPA/ITO DABT63-99-1-0004, DABT63-99-1-0022, NSF IIS-0100851 and NSF CDA9703217. Any opinions, findings and conclusions or recommendations expressed in this material are the author(s) and do not necessarily represent those of the sponsors.

sense of Lyapunov. Fourth, a general framework for concurrent control borrowing from priority-based null-space control of redundant manipulators is described. Finally, an implementation of concurrent control as a mapping of constraints between individual controllers is demonstrated.

2 Related Work

This work is related to four separate related themes in the literature. The first relationship is to the use of visual features to establish correspondence. Visual correspondence is necessary for triangulation and, in this paper, is based on a variation of the histogram intersection technique presented by [19]. However, there are a few differences. First, the histograms are constructed in normalized RGB space, making them more tolerant to color distortions due to illumination changes. Second, instead of using a multi-dimensional histogram, histograms are constructed along each individual dimension and concatenated. This provides significant speed-up and is sufficiently accurate.

The second related thread concerns viewpoint planning [21, 13]. The closest work in this regard is presented by Zhu et. al. [21]. The difference between the two approaches is that the derivation for error uncertainty in their paper is based on a geometric analysis of error in depth measurement that assumes that one observation is accurate and their technique involves some fairly complicated derivations. In contrast, the error uncertainty metric presented here has a simple interpretation in terms of the conditioning of a Jacobian, makes no assumption about observation accuracies, and is quite simple to implement for online planning and control.

Third, there is a strong relation between the existing work, path planning and redundant manipulator kinematics. Individual control objectives are accomplished in this paper using path planning based on harmonic functions [11]. However, the synthesis of a control law is somewhat different and related to an energy based formulation [12]. The formulation for velocity control is also motivated by the generalized pseudo-inverse formulation of inverse kinematics for redundant manipulators [17]. Specifically, the introduction of an *equivalence class* of control actions from a given state is based on the notion that the steepest descent (flowline) path to a goal is not necessarily the only solution and, in fact, by introducing alternate possibilities, concurrent control becomes feasible. In this paper we show that this policy is asymptotically Lyapunov stable.

Finally, there is a relation between this work and other work on reactive robot control architectures. The AuRA system [4] is one such example. The difference from AuRA and other similar architectures [18] is that control composition is not expressed as a linear superposition of individual controllers because, in general, the safety of such compositions, or their correctness cannot be guaranteed. The approach presented here relies on the hypothesis that coordinated control is only feasible when one controller does not violate the stability characteristics of the other. The *equivalence class* formulation of control laws and the subject-to composition operator (see Section 7) are employed to synthesize control compositions and, to the best of our knowledge, such use in mobile robotics is unique. The subject-to formulation is also related to the Subsumption architecture [7] in the sense that there is an implicit prioritization of control. However, there are no hand-crafted implementations of prioritization. Once the task level priorities are specified, a concurrent control expression can be synthesized by automatic analysis of the individual control laws, by using the subject-to operator.

3 Visual Correspondence

A pair of observers can be used to triangulate the position $(x, y)^T$ of the tracked object, given the baseline B and headings θ_1 and θ_2. This is calculated as follows:

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} \frac{B\ c\theta_1\ s\theta_2}{s(\theta_2-\theta_1)} \\ \frac{B\ s\theta_1\ s\theta_2}{s(\theta_2-\theta_1)} \end{pmatrix} \tag{1}$$

However, in order to determine the position of the object the relative pose of the two observers, the pose of at least one observer with respect to the world coordinate frame, and the heading correspondences (θ_1, θ_2) to the tracked object must be known.

In this paper, correspondences are determined using visual features. Panoramic cameras, which offer a 360^o field of view and a straight forward conversion from image coordinates to headings [21], are used. Additionally, one panoramic camera is held stationary and its pose relative to the world coordinate is assumed to be known. The second panoramic camera is mounted on a mobile robot whose position is obtained either through dead reckoning or Monte-Carlo localization using sonar [14]. Thus, the baseline between the two cameras is known. The corresponding headings of a moving object between the two cameras is determined as follows.

In the stationary camera motion detection [3, 16] is used to obtain a region of interest. Then, a one dimensional color histogram in rg (normalized RGB) space is constructed by building the histograms along individual dimensions and concatenating them together. This histogram is transmitted to the mobile camera as a reference feature vector, and is searched within the second image using a windowed histogram intersection approach [19]. Once a match is obtained, the corresponding headings are used in Equation 1 to obtain the position of the target. In Section 8 the performance of this algorithm is discussed.

4 Harmonic Function Path Planning

Having obtained the target position, the mobile robot must execute a trajectory that is obstacle avoiding, remains visible, and terminates at a well-conditioned vantage point. These individual control objectives are implemented using path planning based on harmonic functions [2, 10, 11] discussed below:

Harmonic functions are solutions to Laplace's equation [11]. Given the state $p = (x, y)$ one can write:

$$\nabla^2 \phi = \frac{\partial^2 \phi}{\partial x^2} + \frac{\partial^2 \phi}{\partial y^2} = 0 \tag{2}$$

Harmonic functions have been widely applied in robotics [2, 10, 11, 12]. In the present implementation, Dirichlet boundary conditions are asserted in the form of fixed potentials for goals and obstacles (or other constraints) and numerical solutions are obtained using successive over-relaxation. Upon convergence a path to a goal configuration is available from anywhere within entire mapped space without the existence of local minima (subject to grid resolution). In contrast to off-line methods [6, 8] Harmonic functions can be computed rapidly making them suitable for reactive planning.

The gradient of the harmonic potential function $\nabla\phi$, though attractive at first glance, makes a poor choice for velocity control. When Dirichlet conditions are used the harmonic function ϕ often possesses shallow gradients in a large portion of the mapped space, and this often results in difficulties in computing derivatives between neighboring points.

4.1 Control Law Synthesis

Instead of using $\nabla\phi$, a different approach to velocity control is adopted. First, heuristically, the potential $\phi(\mathbf{p})$ of a state is made available as kinetic energy to the system. Thus, the reference velocity input depends on the potential itself which varies smoothly between 1 at obstacle boundaries and 0 at goals. Second, instead of choosing the flowline direction as the only available control choice, an *equivalence class* of control actions spanning the entire sub-space of negative gradients is allowed. This flexibility is key for accomplishing multiple control objectives concurrently, and is discussed in detail in Section 7.

Formally, the velocity control law can be written as

$$K = \{\dot{\mathbf{p}}\} = \begin{cases} c\sqrt{\phi(\mathbf{p})}\ \hat{e} & \forall \hat{e} \mid \nabla\phi \cdot \hat{e} < 0 \\ \mathbf{0} & \text{otherwise} \end{cases} \tag{3}$$

Here c is an arbitrary constant, and $\hat{e}$ is a unit vector, and K is a continuous set of permissible control actions, from which any single action may be chosen. It is straightforward to see that this law yields asymptotic stability in the sense of Lyapunov.

Define the scalar function $V(\mathbf{p}, t) = \phi(\mathbf{p})$, and observe that ϕ, the harmonic function, is positive definite. Note that $\dot{V} = c\ \sqrt{\phi}\nabla\phi \cdot \hat{e} \leq 0$. Further note that for any path, $\mathbf{s}(\mathbf{p_0}, t_0; t)$, to the goal configuration, $\dot{V}(\mathbf{s}(\mathbf{p_0}, t_0; t), t)$ does not identically vanish to zero. It only vanishes at saddle points. Thus, one can conclude that the proposed velocity control law in Equation 3 is asymptotically stable.

When only one single control objective needs to be satisfied the steepest descent version of Equation 3 can be implemented. In this case $K^* = \dot{\mathbf{p}} = c\sqrt{\phi}\ \frac{\nabla\phi}{\|\nabla\phi\|}$ if $\| \nabla\phi \| \neq 0$, 0 otherwise. The condition where $\dot{\mathbf{p}} = 0$ is either a goal or a saddle. By checking the potential value when this happens, saddles and goals can be distinguished. To escape from saddles, simply continue using the previous velocity reference command. In practice, saddles rarely occur.

5 Visibility

The visibility control objective requires that the target remain visible from the mobile robot at all times, and it is assumed that in the initial configuration the target is visible from both cameras (otherwise it is impossible to triangulate). A visibility controller that is also obstacle avoiding can be synthesized using a harmonic function ϕ_v by mapping the obstacle boundaries and locations in the configuration space invisible from the target as obstacles. Goal points are selected from a subset of the visible points, and the remaining locations are designated as free space. Not all visible points are mapped as goals because this gives the robot enough free space to keep up with an evolving target trajectory instead of facing with a sudden failure of visibility.

6 Precision

The precision control objective is about selecting vantage points that are kinematically well conditioned in the sense that the triangulation uncertainty is minimum. This can be formalized using the notion of a *viewpoint Jacobian*. The error $(\delta x, \delta y)^T$ in the triangulation estimate of (x, y) due to an observation error $(\delta\theta_1, \delta\theta_2)^T$ can be obtained by linearizing Equation 1 around the current operating point $(\theta_1, \theta_2)^T$. A first order approximation, $(\delta x\ \delta y)^T = J(\delta\theta_1\ \delta\theta_2)^T$, yields the following formula.

$$J = \frac{B}{s^2(\theta_2 - \theta_1)} \begin{pmatrix} c\theta_2 s\theta_2 & -c\theta_1 s\theta_1 \\ s^2\theta_2 & -s^2\theta_1 \end{pmatrix} \tag{4}$$

The expression J, here shown to map observation errors to position errors can, infact, be interpreted as the Jacobian which maps observation velocities to Cartesian velocity of the tracked object. Since the two observers are tracking this target from a certain viewpoint, we call J, the *viewpoint Jacobian*.

6.1 Conditioning of the Viewpoint Jacobian

The viewpoint Jacobian is useful for viewpoint control where the objective is to place observers (pairs) in a relative geometry that maximizes the precision of the estimated target position. Letting $\delta\theta = (\delta\theta_1\delta\theta_2)^T$ be the observation uncertainty vector the norm of observation uncertainty can be written as:

$$\| \delta\theta \|^2 = (\delta x\ \delta y)\left(JJ^T\right)^{-1} \begin{pmatrix} \delta x \\ \delta y \end{pmatrix} \tag{5}$$

If it is assumed that the observation uncertainties are configuration independent (a reasonable assumption for panoramic sensors) and bounded (*wlog* the unit circle, $\| \delta\theta \|^2 \leq 1$) then Equation 5 shows that $\left(JJ^T\right)^{-1}$ acts as a configuration dependent amplifier of observation uncertainties to position uncertainties. Akin to the manipulability metric [20], $\kappa = \sqrt{| (JJ^T) |}$ serves as a scalar metric that defines, instantaneously, how uncertain a given triangulation is.

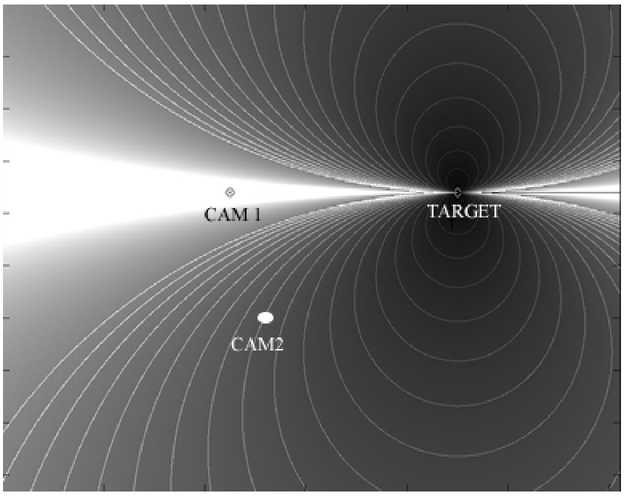

Figure 1: The κ scalar metric for uncertainty for a given camera target configuration

Figure 1 depicts the metric κ for a configuration of the stationary camera(CAM 1) and target over the free space. In this figure, the stationary camera(CAM 1) and the target are separated along the horizontal (X) axis. The gray-scale brightness at any point indicates the scalar uncertainty in estimated target position when the second camera is placed at that location. For example, the brightness at the point marked CAM2 indicates the position uncertainty using the CAM1-CAM2-TARGET configuration. Brighter regions mean greater uncertainty. The brightness function (or equivalently the scalar metric κ) is a bimodal distribution separated by a singularity along the line connecting the stationary camera and the target.

In practice, the singularity is just that, a very thin line of infinite uncertainty joining the camera (CAM1) and the target. It should also be noted that, around the target, the singular line is immediately surrounded by the most precise region. This situation could sometimes lead to a problem, in that it suggests an extremely close distance between the robot and the target as the goal configuration. This issue is addressed by introducing the target (correctly so) as an obstacle from which the robot must remain a certain distance away.

The uncertainty plot suggests at least one method to implement a precision controller; a controller that would servo CAM2 to a location which offers little triangulation uncertainty. In our implementation a small set of N most precise points are mapped as goals, and the singularity line is mapped as an obstacle. Additionally all other obstacles in the C-space are also mapped, so that the precision controller also becomes obstacle avoiding. These boundary conditions upon relaxation produce the harmonic function ϕ_p from which a velocity control law K_p is synthesized.

7 Synthesis of Stable Concurrent Control

The velocity control laws for the visibility K_v and precision K_p are obtained from their corresponding harmonic potential fields ϕ_v and ϕ_p respectively. Each controller is obstacle avoiding because the obstacle boundaries are part of their maps. The viewpoint controller must satisfy, in order of priority, visibility and the precision constraint.

A natural framework for expressing such forms of concurrent control is the *subject-to* operator written as $\triangleleft$. The expression $\phi_p \triangleleft \phi_v$ must be read *run precision subject to visibility*. The motivation for $\triangleleft$ composition rule is drawn from the generalized pseudo-inverse formulation for inverse kinematics of a redundant manipulator [17], where the end-effector can be maintained in a Cartesian configuration while introducing joint motions that lie within the null space of the manipulator Jacobian. Likewise, the subject-to constraint is designed to assert the rule that the subordinate controller (precision) can run in the *null space* of the dominant controller. But, how do we formalize this notion?

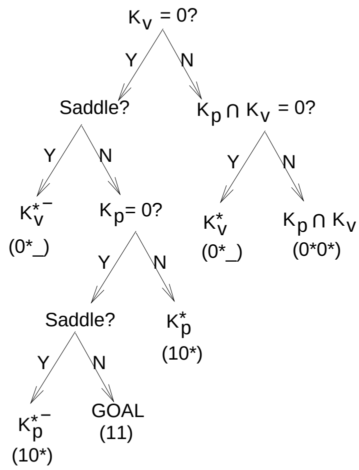

Figure 2: Conditions for composition using the subject-to constraint

In Section 4, a part of this mechanism was described, where an *equivalence class* of control actions were introduced by expanding the possible choices from the flowline to all negative gradient directions from the current state. The intuitive notion is that this allows the dominant controller to accommodate a subordinate controller by picking alternate, albeit instantaneously sub-optimal, control actions but yielding a composition that is guaranteed to result in convergence of both the controllers.

In the context of this paper, $\phi_p \triangleleft \phi_v$ is formalized in the decision tree shown in Figure 2. The leaf nodes of this tree indicate the status of individual controllers and what the convergence properties of the composition may be. For example $0^\star\_$ indicates that the visibility controller is not converged and is running, and the precision controller cannot be run. 11 indicates that the task is accomplished, $0^\star 0^\star$ indicates that both controllers are running, and not converged. The symbol $K_v^\star$ indicates that the visibility controller is running a steepest descent version, and $K_v^{\star -}$ implies that control choices from a previous control cycle are being used. This is essential to escape saddles. Note that this decision tree must be evaluated at every control step.

While the decision tree provides a general basis for priority-based concurrent control for multiple objectives, in practice, the subject-to constraint is implemented by mapping constraints from one control objective onto another. This is accomplished by mapping goals, obstacles, and/or free space. In the $\phi_p \triangleleft \phi_v$ implementation, invisible regions and free space of ϕ_v are mapped as obstacles into the boundary conditions of ϕ_p, and upon convergence ϕ_p is executed. Thus the composition is stable, safe (obstacle avoiding), correct (maintains visibility), and the goal configuration is well-conditioned.

8 Experiments

Experiments are conducted with the following hardware. The mobile panoramic camera [15] is mounted on a Real World Interface(RWI) ATRV Mini platform. Images are digitized using a Leutron Vision PCI framegrabber and processed on-board using the Pentium-II (350MHz) system (robot) and a VMIC 850MHz single board computer (stationary camera). In this section two aspects of our system are demonstrated. The first is the performance of the visual correspondence algorithm and, the second, is the performance the viewpoint controller.

8.1 Visual Correspondence

The model histogram of the most significant motion blob from the stationary camera is transmitted to the mobile robot. At the mobile robot site, at initialization, the model histogram is compared with the acquired image acquired by windowing a 50×50 pixel mask over the entire image. At each step, the histogram of the mobile robot image window is compared with the model histogram using the Swain& Ballard intersection technique [19]. The window with the maximal match value is used to obtain the heading estimate of the object. After initialization, search is limited to a 200×150 pixel area in the neighborhood of the last known image coordinate of the object.

In Figure 3 a snapshot of the matching process is shown. The top image corresponds to the (unwarped) stationary view, and the bottom, to the mobile view. The detected object is used to compute the color histogram and, in this instance, is matched in a 200×150 search window of the mobile camera's view (see bounding box in bottom image). The maximum of the scaled output of the intersection measure (see gray-scale inset, marked *match distribution*) is used to locate the model in the second camera (see white square in window of the bottom image).

8.2 Viewpoint Control

In Figure 8.1 shows a series of snapshots of a trajectory executed by the viewpoint controller. Note that this depiction is not a simulation, but the actual robot and triangulated object trajectory. In the experiments, this data appears on the interface in real-time.

The first (left) image depicts the room, with the stationary camera ($C1$) and obstacles (black). The mapped space is divided into a 21×21 grid with each cell spanning $30 \times 30cm^2$. The stationary camera is placed at (8,3) grid location and, initially, the robot is at (14.5, 3).

A single human subject is tracked along an accurately known path shown as a solid line between the two dotted guide lines and the triangulations produced by the system as the viewpoint controller is executing is shown as the small circles within the two dotted lines. The robot's path itself is shown to the right. As can be seen in the first image, the human path requires the robot to move so it can remain visible, avoid obstacles and converge to a good viewpoint. The remaining three images of Figure 8.1 show snapshots along the robot trajectory. In these images white represents goals, black represents obstacles and invisible locations, and gray represents the harmonic potential (brighter is smaller).

9 Conclusions & Future Work

In this paper a reactive and concurrent control framework for viewpoint control is demonstrated. The task is decomposed into three control objectives namely; obstacle avoidance, visibility and precision. Individual objectives are accomplished by path-planning using harmonic functions, and a task controller is synthesized using an interpretation of the subject-to, $\triangleleft$, priority-based concurrent control composition rule. A general framework for analyzing the rule is described in the form of a decision tree, and an example implementation in the form of $\phi_p \triangleleft \phi_v$ was shown by mapping constraints from one objective into another. The composition is shown to yield stable, safe (obstacle avoiding), correct (maintains visibility), and well-conditioned goal configurations. Tracking is accomplished by establishing visual correspondences using color. A scalar uncertainty metric, similar to the manipulability metric, was introduced to compute goal sets the precision controller. A velocity control law that allows an equivalence class of control actions was introduced. The system was demonstrated using a mobile panoramic sensor, a stationary panoramic sensor and a moving human subject.

One of the avenues that is being explored is to improve the speed of computing visual correspondence using motion estimation in the moving camera. The second direction that this work is being extended is in showing Lyapunov stability by also considering the dynamical parameters of the robot. The third direction involves extending the current framework to multiple moving sensor platforms. Finally, we are exploring ways for learning optimal control sequences for multiple control objectives using the *equivalence class* formulation of the velocity control law.

References

[1] Gregory D. Abowd and Elizabeth D. Mynatt. Charting past, present and future reseach in ubiqitous computing. *ACM Transactions on Computer-Human Interaction*, 7(1):29–58, March 200.

[2] S. Akishita, S. Kawamura, and K. Hayashi. Laplace potential for mobing obstacle avoidance and approach of a mobile robot. In *Japan-USA Symposium on Flexible Automation, A Pacific Rim Conference*, pages 139–142, 1990.

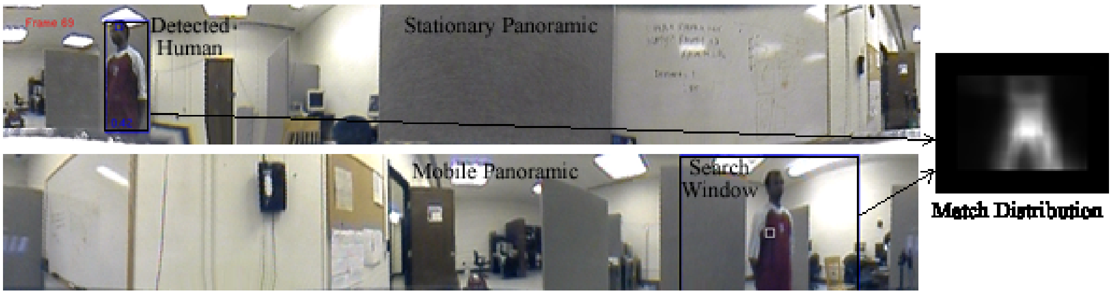

Figure 3: Establishing Visual Correspondence

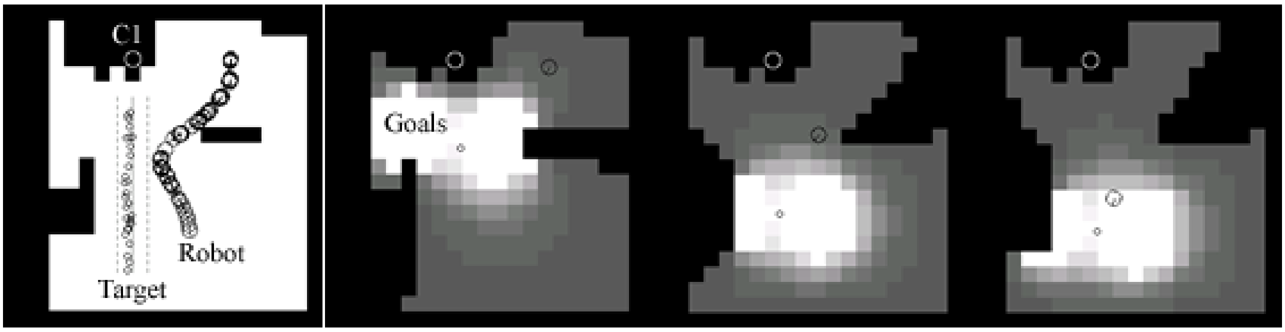

Figure 4: Composite Viewpoint controller in action.

[3] E. Araujo, D. Karuppiah, Y. Yang, R. Grupen, P. Deegan, B. Lerner, E. Riseman, and Z. Zhu. Software mode changes for continuous motion tracking. In *International Workshop on Self Adaptive Software*, Oxford, England, April 2000.

[4] R. C. Arkin and T. Balch. Aura: Principles and practice in review. *Journal of Experimental and Theoretical Artificial Intelligence*, 1997.

[5] T. Balch and R. C. Arkin. Behavior-based formation control for multiagent robot teams. *IEEE Transactions on Robotics and Automation*, December 1998.

[6] Jerome Barraquand and Jean-Claude Latombe. Robot motion planning: A distributed representation approach. *International Journal of Robotics Research*, 10(6):628–649, 1991.

[7] R. A. Brooks. A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation*, 2(1):14–23, March 1986.

[8] J. F. Canny. *The Complexity of Robot Motion Planning*. MIT Press, 1988.

[9] Michael H. Cohen. Design principles for intelligent environments. In *Proceedings of the National Conference on Artificial Intelligence*, Madison, Wisconsin, July 1998.

[10] C. I. Connolly, J. B. Burns, and R. Weiss. Path planning using laplace's equation. In *Proceedings of the IEEE Conference on Robotics and Automation*, pages 2102–2106, 1990.

[11] C. I. Connolly and R. A. Grupen. The applications of harmonic functions to robotics. *Journal of Robotic Systems*, 10, 1993.

[12] C. I. Connolly, R. A. Grupen, and K. Souccar. A hamiltonian framework for kinodynamic planning. In *Proceedings of the IEEE Conference on Robotics and Automation*, 1995.

[13] T. Matsuyama et. al. Dynamic memory: Architecture for real time integration of visual perception, camera action, and network communication. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, Hilton Head Island, SC, June 2000.

[14] D. Fox, W. Burgard, F. Dellaert, and S. Thrun. Monte carlo localization: Efficient position estimation for mobile robots. In *Proceedings of AAAI*, 1999.

[15] P. Greguss. Panoramic imaging block for three-dimensional space. U.S. Patent 4,566,763, January 1986.

[16] G. Holness, D. Karuppiah, S. Uppala, S. C. Ravel, and R. A. Grupen. A service paradigm for reconfigurable agents. In *Proc. of the 2nd Workshop on Infrastructure for Agents, MAS, and Scalable MAS*, Montreal, Canada, 2001.

[17] Y. Nakamura. *Advanced Robotics: redundancy and optimization*. Addison-Wesley, 1991.

[18] J. Rosenblatt. Damn: A distributed architecture for mobile navigation. In *Proceedings of the AAAI Spring Symposium on Lessons Learned from Implemented Software Architectures for Physical Agents*, Stanford, CA, Match 1995.

[19] M. Swain and D. Ballard. Color indexing. *Int. Jrnl. Comput. Vision*, 7(1):11–32, 1991.

[20] T. Yoshikawa. Manipulability of robotic mechanisms. *International Journal of Robotics Research*, 4(2):3–9, 1985.

[21] Z. Zhu, K. Rajasekar, E. Riseman, and A. Hanson. Panoramic virtual stereo vision of cooperative mobile robots for localizing 3d moving objects. In *Proceedings of IEEE Workshop on Omnidirectional Vision*, pages 29–36, 2000.