Proceedings of the 2002 IEEE
International Conference on Robotics & Automation
Washington, DC • May 2002

Waiting with José, a vision-based mobile robot *

Pantelis Elinas Jesse Hoey Darrell Lahey Jefferson D. Montgomery
Don Murray Stephen Se James J. Little
Computer Science Dept.
University of British Columbia
Vancouver, BC, Canada V6T 1Z4
{elinas,jhoey,lahey,jdm,donm,se,little}@cs.ubc.ca

Abstract

José is a visually guided autonomous robotic waiter. He circulates around a room populated by groups of people, politely serving appetizers to humans. The serving task combines elements of robotics with human computer interaction, challenging control architecture with multiple task integration. This paper describes our purely vision-based approach to this task. Methods for mapping, localization and navigation are presented and discussed, including issues of safety for both robots and humans. Our work on human-robot interaction is covered, as well as our solutions to various tasks specific to serving food. We present results of our methods from sample experiments in our laboratory. We further discuss our experiences at the 2001 AAAI mobile robot "Hors D'œuvres Anyone?" competition, at which José took first prize.

1 Introduction

This paper is about using vision for autonomous robotics. Vision provides rich, high bandwidth, two dimensional data containing information about color, texture, depth and optic flow, among others. This multi-modal data source can be exploited universally for the accomplishment of many different tasks. It is a harmonious host of information about a robot's environment, and is an alternative to more specialized sensors such as sonar or laser range finders. Although vision is such a rich data source, it usually requires complex techniques for the extraction of useful information. For example, while sonar data directly estimates depth information, vision data (from multiple cameras) requires a stereo matching algorithm. However, the vision sensors can estimate further properties of environmental structure using the integrated color and texture information.

In this paper, the techniques we have developed for using vision are discussed in the context of a particular robotic task: serving food to a gathering of people. To accomplish this task, a robot must reliably navigate around a room populated by groups of people, politely serving appetizers to humans. The robot must also monitor the food it has available to serve, and return to a home base location to refill when the food is depleted. The serving task involves many basic aspects of mobile robotics, including localization, mapping, navigation and human-robot interaction. In ongoing research, we have developed a solid framework for accomplishing these fundamentals using only vision as a sensor on our autonomous robot, *José* (Figure 1). Problems specific to the serving task were also solved using vision, including finding people to serve and monitoring food.

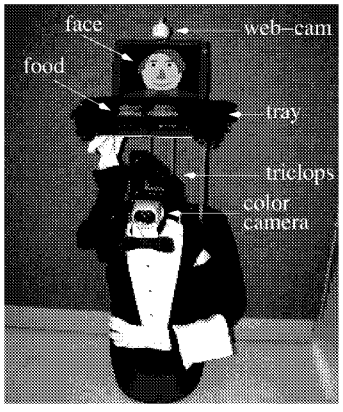


Figure 1: *José*, the robotic waiter

Previous approaches to the autonomous serving task include *Alfred* [10], the winning robot waiter at the 1999 "Hors D'œuvres Anyone?" competition. *José* differs from *Alfred* in three respects. First, *Alfred* relies on sonar for navigation, while *José* uses only vision. Second, *Alfred* focused on speech recognition much more than *José*. Although *Alfred*'s speech recognition worked well in the laboratory environment, it performed poorly in the crowded, noisy conference reception hall typical of "Hors D'œuvres Anyone?" competitions [10]. Commercial speech recognition systems, as used by both *Alfred* and *José*, have not reached the level of accuracy needed for conference reception environments, and we therefore decided not to rely on speech recognition for *José*. Third, *Alfred* needed special landmarks for navigation, and had lighting and scale dependent landmark recognition systems. *José* uses natural landmarks and a scale and illumination invariant recognition system.

*This research was supported by grants from the Natural Sciences and Engineering Research Council of Canada and the Institute for Robotics and Intelligent Systems.

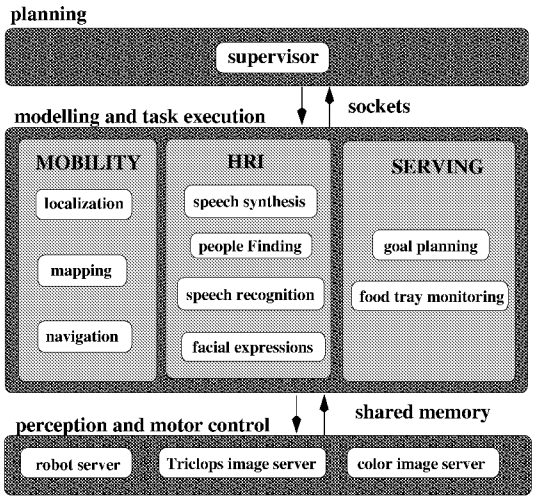


Figure 2: Control Architecture

This paper is structured as follows. The next section presents an overview of our mobile vision-based robot, *José*, including his hardware systems, and his software architecture which separates control into low, mid and high-level behaviors. Sections 3, 4 and 5 present details on these three levels. Section 6 shows results from some sample serving runs, and discusses our experiences at the 2001 AAAI mobile robot competition. Section 7 concludes the paper.

2 José

José is a Real World Interface (RWI) B-14 mobile robot equipped with an Intel Pentium PC running the Linux operating system. *José* senses the environment through five cameras, three of which are are encapsulated in a Triclops stereo vision camera module. *José* has a Sony pan-tilt color camera, which is used for locating people to interact with. A Sony laptop computer mounted above *José* supports the food tray and its screen displays *José*'s face. A Logitech web-cam keeps watch over the food tray. *José* has a Compaq wireless ethernet modem which allows his software systems to be distributed. In particular, a second Linux PC serves as a host computer running a supervisor module, and the laptop computer runs the food monitoring and face generation programs.

José is driven by a hierarchical behavior-based control architecture [1], as shown in Figure 2. The system divides the robot's behaviors into three levels, each of which contains simple, independent, modules. The modularity of the system makes implementation and testing simple and efficient. The lowest level involves perception and motor control, and includes servers interfacing with the robot's motors and odometry, the Triclops unit, and the color camera.

The middle level includes modules for the various behaviors the robot needs to perform its tasks. These fall into three categories, as shown in Figure 2.

Mobility: Behaviors which enable the robot to circulate in its environment: mapping, localization, and navigation.

Human-Robot Interaction (HRI): Behaviors for interacting with humans: finding people to interact with, speech recognition and synthesis, and facial expression generation.

Serving: Behaviors specific to the serving task: Planning locations for service and monitoring the food tray.

The highest level is a single supervisor behavior which delegates tasks to middle level modules. To ensure scalability of the system, the supervisor runs on a remote computer, and communicates with the middle level behaviors through sockets. The middle level modules communicate with the lowest level through a shared memory architecture. Middle and low level behaviors must therefore all run on the robot, with the exception of speech recognition, the facial expressions and the tray monitoring, which communicate directly with sensors.

The following three sections will describe each of the three levels in Figure 2.

3 Perception and Motor Control

José's trinocular stereo unit (Triclops) outputs three images. The corresponding dense two-dimensional depth information is used as the primary input for map building, localization, navigation, and people finding behaviors. Triclops was developed at the UBC Laboratory for Computational Intelligence (LCI) and is being marketed by Point Grey Research, Inc. (*www.ptgrey.com*). A Matrox Meteor frame grabber connects the Triclops to *José*. The Triclops stereo vision module has 3 identical wide angle (90° degree field-of-view) cameras, arranged in an L shape. The system is calibrated, and corrected for lens distortion and camera misalignment in software to yield three corrected images that conform to a pinhole camera model with square pixels. The camera coordinate frames are co-planar and aligned so that the the epipolar lines of the camera pairs lie along the rows and columns of the images.

The trinocular stereo approach is based on the multi-baseline stereo developed by Okutomi and Kanade [12]. Each pixel in the reference image is compared with pixels along the epipolar lines in the top and left images. The comparison measure used is sum of absolute differences. The results of the two image pairs (left/right, top/bottom) are summed to yield a combined score. Multi-baseline stereo avoids ambiguity because the sum of the comparison measures is unlikely to cause a mismatch—an erroneous minimum in one pair is unlikely to coincide with an erroneous minimum in another pair. Examples of the stereo results are shown in Figure 3(a) and (b). Further details on the stereo algorithm we use can be found in [11].

A PCTV frame grabber card delivers color images from the Sony pan-tilt unit through the color image server (see Figure 2). The color images are registered with the stereo images from the Triclops using a offline manual calibration. The calibration must be repeated only when the positions (relative to the robot) of the color camera or Triclops unit are adjusted. We are currently replacing the Triclops and

color camera with a single digital color Triclops unit called Digiclops (also from Point Grey Research). The Digiclops unit, with integrated color and stereo information, will circumvent the need for the calibration.

The RWI robot platform has motor controls for rotation and translation, and provides odometry data. Although the odometry is fairly accurate, it can lead to serious errors in mapping and localization over the time period of a typical circulation of the serving robot. Methods for correcting such errors are discussed in Section 4.1.2.

4 Modeling and Task Execution

This section describes the mid-level behaviors which enable *José* to accomplish basic mobility (mapping, localization and navigation), human-robot interaction (people finding, speech synthesis and recognition, and facial expressions), and other behaviors specific to the serving task (goal planning and tray monitoring).

4.1 Mobility

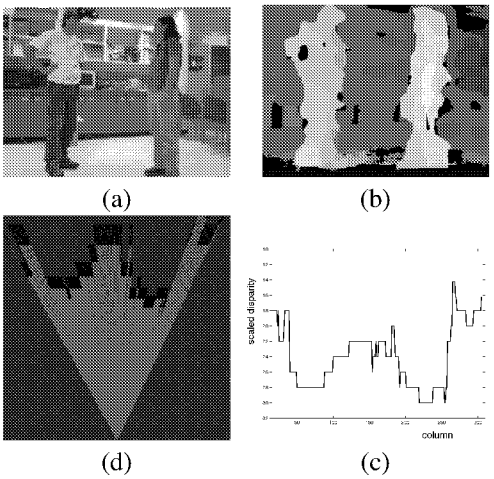
The most fundamental, by no means the simplest, task for a mobile robot is moving around in its environment. This must be accomplished within certain safety limits for the robot. If humans are present (as in the serving task), their safety cannot be jeopardized. These constraints are satisfied by building an accurate map, localizing the robot, and then navigating safely through the mapped environment, as we now describe.

4.1.1 Occupancy Grid Mapping

Occupancy grid mapping, pioneered by Moravec and Elfes [5], is the most widely used robot mapping technique due to its simplicity, robustness and flexibility in accommodating many kinds of spatial sensors. It also adapts well to dynamic environments. The technique divides the environment into a discrete grid and assigns to each grid location a value related to the probability that the location is occupied by an obstacle. Initially, all grid values are set to 50%, indicating equal probability that the grid location is occupied and unoccupied. Sensor readings supply uncertainty regions within which an obstacle is expected to be. Probabilities at grid locations that fall within these regions of uncertainty are increased while those at locations in the sensing path between the robot and the obstacle are decreased.

Although occupancy grids may be implemented in any number of dimensions, most mobile robotics applications (including ours) use 2D grids. Much of the 3D data is lost in the construction of a 2D occupancy grid map. The robot possesses 3 DOF (X, Y, heading) within a 2D plane corresponding to the floor. The robot's field of view sweeps out a 3D volume above this plane. A projection of all obstacles within this volume to the floor uniquely identifies free and obstructed regions in the robot's space.

Figure 3 shows the construction of the 2D occupancy grid sensor reading from a single 3D stereo image. Figure 3(a) shows the reference camera greyscale image (320x240 pixels), and (b) the resulting disparity image. Black regions indicate image areas which were invalidated. Otherwise, brighter areas indicate higher disparities (closer



(a) (b) (d) (c)

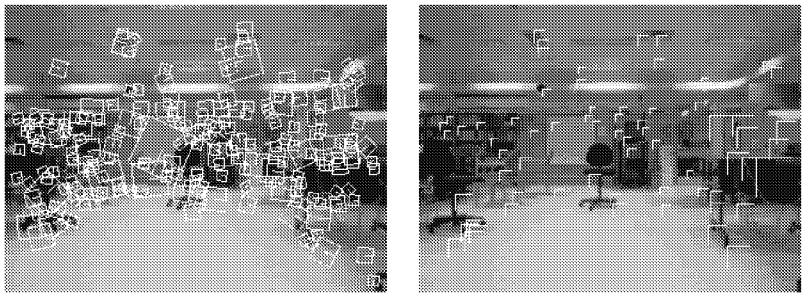
Figure 3: From stereo images to radial maps. (a) greyscale image (b) disparity image (black indicates invalid, otherwise brighter indicates closer to the cameras) (c) depth vs columns graph (depth in cm) (d) the resultant estimate of clear, unknown and occupied regions (light grey is clear, black is occupied and dark grey is unknown)

to the camera). The maximum disparities in each column are converted to depth to produce a *radial map*, as shown in Figure 3(c). Figure 3(d) shows these depth values converted into an occupancy grid representation; light grey indicates clear regions, black indicates occupied, and dark grey indicates unknown areas. The process illustrated in Figure 3 generates the input into our stereo vision occupancy grid. The mapping system then integrates these values over time, to expand the map and keep it current in the changing world. We identify an obstacle at all locations in the occupancy grid where the value is above a threshold. Figure 10 shows examples of occupancy grids generated in this way.

4.1.2 Localization

Safe mobility involves simultaneous localization and mapping (SLAM). The robot must build a map of the environment and track its position relative to that environment. However, accurate localization is a prerequisite for building a good map, and having an accurate map is essential for good localization. This problem has been a central research topic for the past few years [16, 3, 17, 4, 18]. Our vision-based SLAM algorithm uses Triclops stereo data of features detected by the Scale Invariant Feature Transform (SIFT) [9]. Simply put, *José* finds out where he is by recognizing and locating previously observed visual features in his environment. SIFT features are invariant to image translation, scaling, rotation, and partially invariant to illumination changes and affine or 3D projection. These characteristics make SIFT features suitable landmarks for mapping and localization, since when mobile robots are moving around in an environment, landmarks are observed from different angles, distances and under different illuminations. Figure 4(a) shows an example of detected SIFT features, including scale and orientation.

The SIFT features must be located in three dimensions.



(a) (b)

Figure 4: (a) SIFT features found, with scale and orientation indicated by the size and orientation of the squares. (b) Stereo matching result, where horizontal and vertical lines indicate the horizontal and vertical disparities respectively.

To accomplish this, we match SIFT features in each of the three images delivered by the Triclops system combined with epipolar and disparity constraints. Figure 4(b) shows the final disparities of all consistent SIFT features. From the positions of the matches, and the camera intrinsic parameters, we can compute the 3D world coordinates of each feature relative to the robot. We maintain a database of the located SIFT landmarks and use it to match features found in subsequent views. Once the SIFT features are matched, we can use the matches in a least-squares procedure to compute a more accurate camera ego-motion and hence correct localization errors. This SLAM results in a 3D map of SIFT features, and an accurate position and orientation of the robot in the map. The SIFT map presently is separate from the occupancy grid, but in principle it can be integrated with the grid so that errors due to drift and slippage can be corrected. An example SIFT map with over 2000 landmarks is shown in Figure 5. Readers are referred to [15] for further details of the SLAM technique.

Figure 5: Bird's eye view of the SIFT map around base region: the home location is indicated by the square with the current robot position and view direction shown as a V.

The database of SIFT features can also be used for global localization [14], i.e., determining current position with no prior position information. To tackle this problem, we consider matching a set of SIFT landmarks as a whole. Given a small set of current SIFT features and a large set of SIFT landmarks in the database, we would like to estimate the robot position that would have brought the largest number of landmarks into close alignment, provided that the robot has previously viewed the current scene during the map building stage. We use the RANSAC (RANdom Sample And Consensus) method to generate hypotheses of the form: (X, Z, θ) where X is the sideways displacement, Z is the forward displacement and θ is the orientation. We select the hypothesis with the maximum number of matches and the lowest least-squares error.

Global localization is necessary for serving food when the robot must find its way back to a home base to refill with food. When a refill is required, the robot navigates by dead-reckoning to around 2m away from the home base and carries out global localization there. Figure 5 shows the positions of robot and home base at this stage. Using the localization estimate, the robot can then proceed to the home region successfully for refill.

4.1.3 Navigation

Given a goal location, the robot position, and the occupancy grid map, we want to find the shortest and safest path connecting the two. The path planning algorithm we use is a mixture of shortest path [8] and potential field methods [7, 2]. In clear areas, the method operates as a shortest path planner with a fixed distance constraint from obstacles. In cluttered areas, the method turns into a potential field planner, to avoid getting stuck. The combination of the two allows the robot to navigate efficiently in clear environments without getting stuck in cluttered areas. Our navigator is described more fully in [11].

4.2 Human-Robot Interaction

We are mainly interested in robotic tasks oriented towards people, and devote a significant portion of our research to human-robot interactions. We wish to develop natural interfaces for control of and for social interaction with our robots. Natural interfaces include speech, gesture and facial expression. This section describes our efforts towards enabling *José* with the capacity to find people in his environment (a necessary precursor to interaction) and with natural interaction behaviors.

4.2.1 Finding People

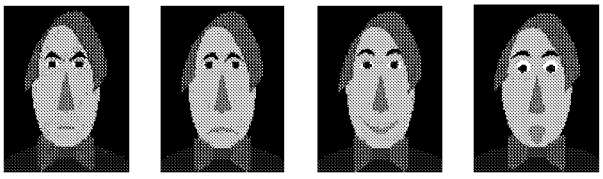
Humans are distinguished from the environment in a two-stage process: skin-color segmentation followed by rejection of false positives using the occupancy grid. One feature that all people have in common is the hue of their skin. The hue of human skin falls in a narrow range which is largely invariant to a person's skin color. The threshold value is decided during a training stage by calculating the mean and standard deviation of the hue of a number of sample skin pixels. We re-train the system for significant changes of the illumination in the operating environment. On average, the hue threshold falls around the value 30 ± 10.

José converts RGB color images from the color camera to HSV color space and segments to select the human skin colored pixels. Since the color images are registered with the Triclops stereo data (see Section 3), 3D locations of skin-colored regions are recovered. These locations are then projected to the floor, and used to build a 2D map of people locations (see Figure 10 for examples). Some objects have hues very similar to human skin (*e.g.*, cardboard and wood). *José* must differentiate people from such obstacles to ensure appropriate serving behavior (*e.g.*, so as not to serve wooden tables). Fortunately, *José*'s map, as described in Section 4.1.1, is built while he is alone in the area he is to operate. Thus, we can compare each selected skin pixel's projected floor location against the occupancy grid and ignore locations that are unoccupied or marked as static obstacles. Two examples of skin-color detection are shown in Figure 9. While the segmentation clearly misses some of the skin colored regions in both images, there are no false positives remaining after comparison with the occupancy grid map. The 2D people location map is integrated over time, resulting in a map, $P_p(\vec{x}, t)$, giving the probability that a person is at location $\vec{x}$ at time t. Figure 10 shows examples of this map during a typical serving run.

4.2.2 Interacting with People

A robot gains acceptance by humans if it allows for natural interaction. We have explored interactions between *José* and his customers using speech and facial expressions. *José* uses a DoubleTalk speech generation engine to utter predefined statements. In conjunction with speech, facial expressions are displayed with an animated face on a laptop screen mounted above the serving tray. Examples are shown in Figure 6. The animated face lends expressiveness to the speech, thus making interactions with *José* more interesting for his customers. While many face generation systems use complex 3D graphics [10], *José*'s face is a simple cartoon. This allows for fast rendering, and does not detract from interaction quality, since humans will interact with even the simplest of generated faces as a real human face [13].

José has speech recognition capabilities, but has not made extensive use of them yet. We chose not to rely on speech recognition, as robustness to environmental factors has not yet emerged in commercially available products. We are also working on facial expression and gesture recognition for *José* [6].



(a) (b) (c) (d)

Figure 6: Faces coincide with speech (a) "Stop stealing food!", (b) "I'm sorry", (c) "Would you like an appetizer?", (d) "I have no food left!"

4.3 Serving

The particular task we have used recently as a testbed for our vision-based robotic system is that of serving food to a gathering of people. This task requires many of the behaviors which have been implemented on our robotic platform, *José*. Serving also necessitates some additional task-specific behaviors: circulating in a room full of people, ensuring coverage (everyone gets served) and making sure there is food in the serving tray. This section describes these two behaviors.

4.3.1 Route planning

José must plan a route through the environment that enables him to offer food to candidate humans and to return to his refilling station when required. This is accomplished using a procedure that dynamically determines the best feasible goal location. At each time, t, the best goal is defined using a dynamic *desirability function*, $\mathcal{D}(\vec{x}, t), \vec{x} \in E$, where E is the spatial extent of the environment. The *desirability* of a location, $\vec{x}$, tells the robot the utility of being at position $\vec{x}$ at time t given that the current robot position is $\vec{x}_r(t)$. A goal is chosen as the maximum of the *desirability* function.

We calculate the *desirability* as a weighted sum of the people probability map, $P_p(\vec{p})$ (Section 4.2.1), and three cost terms C_c, C_o, and C_h.

1. The cost associated with locomotion, $\mathcal{C}_c$, is given by the distance of a path planned to $\vec{x}$ from the current robot position, $\vec{x}_r(t)$: $\mathcal{C}_c = c\left(\vec{x}_r\left(t\right), \vec{x}\right)$.

2. The cost of proximity to obstacles, $\mathcal{C}_o$, is given by $\mathcal{C}_o = \min_{i=0..N_o} \|\vec{x} - \vec{o}_i\|$, where $\vec{o}_i$ is the location of the i^{th} static obstacle (Section 4.1.1). and N_o is the number of obstacles.

3. The cost of serving at previously served sites, $\mathcal{C}_h$, is given by $\mathcal{C}_h = \sum_{\tau=1}^{\tau_{\max}} m^{c(\vec{x}, \vec{x}_r(t-\tau))} h^\tau$ where the constant parameter $m \in (0, 1)$ adjusts *José*'s desire to serve as many locations as possible, and the constant parameter $m \in (0, 1)$ discounts the past. The history is not considered beyond a horizon $\tau_{\max}$.

The *desirability* function is given by:

$$\mathcal{D} = P_p - \omega_o^2 \mathcal{C}_o - \omega_h^2 \mathcal{C}_h - \omega_c^2 \mathcal{C}_c,$$

where the weights, ω, are parameters specified by the designer. Figure 11 shows some example *desirability* maps.

Our experiments in various environments have shown that maximization of the desirability function produces reasonable goals. If no people are detected, *José* will wander the room in an exploratory fashion. If people are detected, *José* will try to serve the closest person. The more people that are detected, the longer *José* will remain in the area to serve before moving on.

Despite this success, we have found that such a primitive motion model to be insuficient in general. The assumption that congregation sites remain relatively stationary in a typical reception setting was found to be misguided. Therefore, a method of dynamically tracking the desired target(s) and appropriately adjusting the goal location is important for general application, and is a subject of ongoing research.

4.3.2 Monitoring appetizers

As shown in Figure 1, *José* carries a food tray monitored by a Logitech web cam. Monitoring the amount of food allows *José* to detect when someone takes food, and when the tray is empty (calling for a return to base). The tray is solid black and has a dull texture so that regions containing food will have a significantly higher intensity than the background, allowing the percentage of food on the tray to be estimated using a simple thresholding operation.

Other objects, such as human hands, occasionally appear in the cameras field of view, causing increases in the percentage of segmented pixels. However the amount of food on the tray should only decrease as people take food from the tray. The amount of food on the tray should increase only when *José* is at home base for refilling. Therefore, if the number of non-black pixels suddenly increases significantly, it is likely that some other object has entered the image. However, a persistent increase indicates new food on the tray.

José keeps a ten second history of the percentages of non-black pixels that it has computed for the images. The percentage of food on the tray is estimated as the minimum of the percentages of non-black pixels in the history. With this strategy, an increase in the percentage of non-black pixels will not affect the food percentage unless the increase persists for the entire length of the history.

Figure 7 shows images of the food tray before, during, and after, respectively, a person takes food from it. The

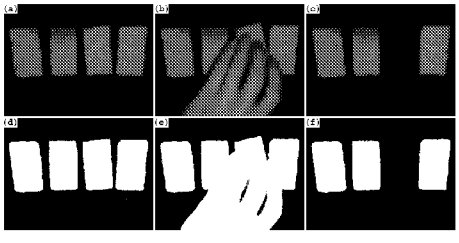


Figure 7: Top row: images taken while a person helps himself to an appetizer. Bottom row: segmented regions.

top set of images are the images taken by the camera. The bottom set of images are the corresponding images resulting from segmentation of food pixels (shown in white) from

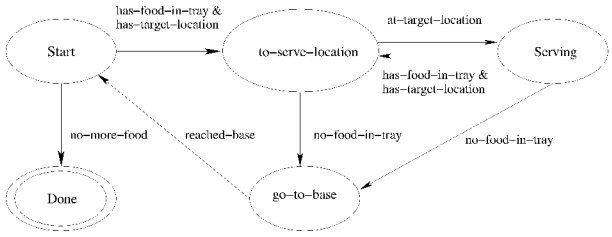


Figure 8: The state diagram for *José*'s serving behaviors

tray pixels (shown in black). Before the person takes food from the tray (leftmost image), the computed food percentage is 34%. When the person moves his hand into the view of the camera (middle image), to take food, the number of non-black pixels increases. However, the computed food percentage remains at 34%. Once the food is removed (rightmost image), the computed food percentage drops to 26%.

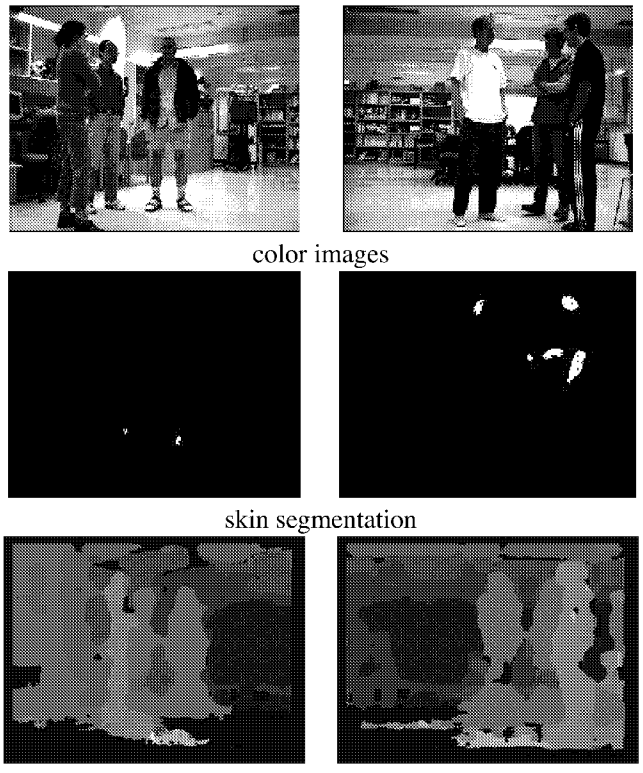
5 Planning

The highest level of control belongs to a supervisor that activates mid-level behaviors to achieve the task at hand. The behavior of the supervisor is modeled with a finite state automaton, as shown in Figure 8. The supervisor remains in the *start* state while the robot waits at the base location for food to be placed in its tray. When the tray is filled and the goal planner returns a location to serve, the supervisor enters the *to-serve-location* state and the supervisor directs the navigator to move to the goal location. Once the robot arrives at the goal, the supervisor enters the *serving* state, and directs the speech and face to offer food. After a brief pause, the supervisor checks the amount of food left. If there is still food in the tray, the goal planner is again invoked. If there is no food left in the tray, the supervisor switches to the *go-to-base* state, and directs the navigator to return back to the base. When the robot arrives near the base, the supervisor invokes the localization behavior, which globally locates the robot and base, allowing a precise move to the base.

6 Results

Figures 9, 10, and 11 show data from an example serving run performed in our laboratory. *José* had previously built an occupancy grid, which is shown in Figure 10. The occupancy grids show unexplored and explored space in dark and light gray, respectively, while obstacles are shown in black. *José* starts at his home base and performs a visual scan of his environment. Color, skin segmentations and stereo data from this scan are shown across two images in Figure 9. The people finding behaviour locates two groups based on the skin segmentations and stereo data, as shown in light-colored pixels overlaid on the occupancy grids in Figure 10(a). The goal planner computes the initial desirability function, as shown in Figure 11(a). The parameters for the desirability function were set to $\omega_o = 3.0$, $\omega_h = 1.0$, $\omega_c = 0.2$, $m = 0.75$ and $h = 0.9$.

The first goal is chosen as the maximum of this function, and is centered on the group of people to *José*'s right. The navigator plans a path to this goal, and *José* offers appetizers



color images

skin segmentation

stereo images

Figure 9: Two views of *José*'s environment from the home base as seen though the color camera (top row) and corresponding skin segmentations (middle row) and stereo depth images (bottow row). Note how the positions of the two groups of people in these images relate to the 2D people locations shown superimposed on the occupancy grids in Figure 10

upon arrival in front of the group of people, as shown in Figure 11(b). The goal planner incorporates this serving location into the desirability function, as shown in Figure 11(b). The new maximum of the function is centered on the second group of people, which becomes the next goal location. Figure 10(b) shows *José*'s path as he navigates to the second group and again offers food. *José* has now run out of food on his tray, and proceeds to navigate back to home base, as shown in Figure 10(c). To ensure accurate global localization, he first navigates to a point 2m in front of the base, performs the localization, and then navigates to the base, arriving within 10cm.

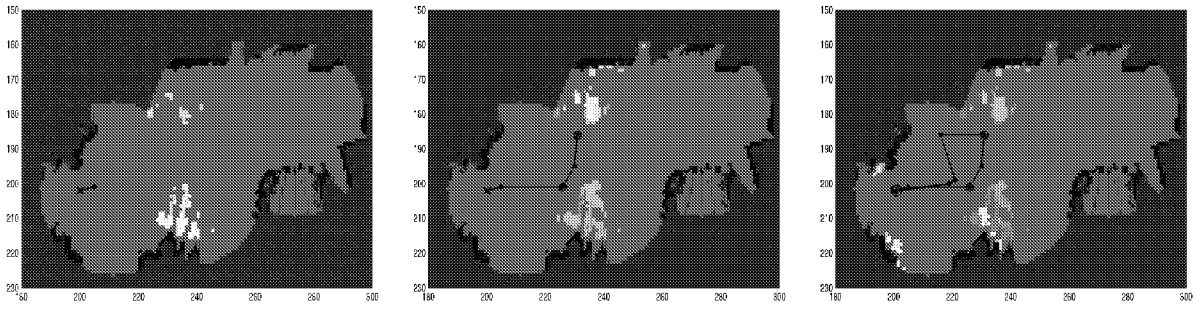
José was deployed at the *Hors D'œuvres Anyone?* mobile robot serving competition in Seattle. He detected and approached groups of people, knew when his tray was empty, and found his way back home to within 10cm. *José*'s face, voice and well-tailored dress were great crowd pleasers, evoking many smiles and laughter. Our experience at the competition uncovered two facts about the robotic serving task. We found that vision alone is sufficient to perform the serving job. We also realized that probabilistic dynamic modeling of people would be a very useful additional component to a system for robotic waiting.

We were initially apprehensive about *José* moving in a room full of people without using sonar. Collisions with humans in the room would lead to disqualification. However, we found that our vision capabilities provided ample real-time feedback about the positions of obstacles to allow the robot to successfully and safely move and serve. The primary reason for these capabilities is the fast, high quality stereo data provided by the Triclops system. However, using vision data alone does impose constraints. First, stereo matching takes time. The result is a bound on the translation and rotation speeds that the robot can achieve. Translation is limited to avoid collisions. The robot cannot react to objects in its path until they appear in the stereo data. Rotation is limited because multiple frames are needed to confirm the presence or absence of an obstacle. If the robot rotates too quickly, the presence and position of an obstacle will not be confirmed, and will not appear in the occupancy grid, possibly leading to a collision. Translation speed was set to 30 cm/s, while rotation speed was set to 10 deg/s. While this gave fairly satisfactory performance, an increase in speed would give a more life-like performance. The second constraint imposed by our stereo vision data is a limited field of view, implying a limited amount of map updates which can be performed in a time interval. Sonar and laser range data avoid this problem with omnidirectional scanning. Additional Triclops units could be used to acheive a larger field of view for a vision based robot.

The *Hors D'œuvres Anyone?* competition showed that our assumption of static groups of people is not often valid in a serving environment. People are dynamic objects, and seem to behave in strange ways in the presence of a robot. Many of the observed human behaviors were attempts to provoke some kind of reaction from the robot: clustering around, waving hands in front of the cameras, attempting to block *José*'s path, etc. These behaviors were interpreted by *José* (perhaps correctly) as attempts to foil his serving task. He would remonstrate with the culprits, sometimes to no avail. *José* currently makes assumptions about the dynamics of people. *José* chooses a group of people to serve, and then makes his way to the location of the group. Once he begins, he does not verify that the group has maintained position, and continues until he reaches his target. In many cases, the chosen group moves, often towards *José*. If they come towards him, he perceives them as an obstacle, asks them to move, and waits for them to do so. We are currently working on simple following behaviors which will avoid these kinds of problems. However, a more general dynamic plan updating scheme would be an asset. We are currently investigating a probabilistic people mapping algorithm. As well, we are combining the occupancy grid navigation and obstacle avoidance with the localization and odometry correction provided by the SIFT map.

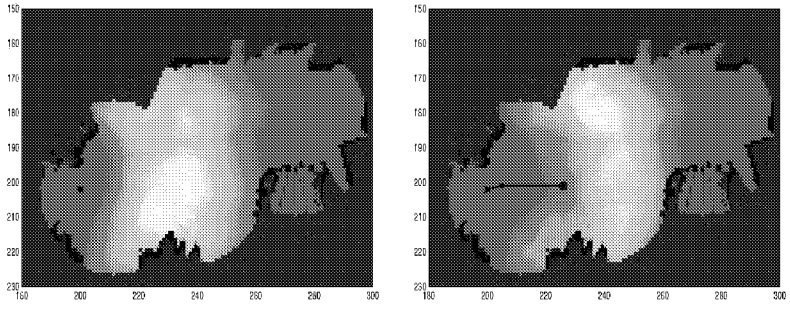
7 Conclusions

We have presented our visually guided autonomous serving robot, *José*. Mapping, localization and navigation issues which have been the focus of recent research in our laboratory were discussed. Human-robot interaction, and serv-



(a) (b) (c)

Figure 10: Occupancy grids at three times during a serving run. Also shown are the accumulated skin-color maps, and *José*'s trajectory, with an 'x' marking the home base, and 'o's marking serving locations.



(a) (b)

Figure 11: Desirability maps at two times during a serving run.

ing issues were also covered. Our results show that *José* is capable waiter, combining effective robotic techniques with panache and wit, and the delicate savoir-faire of an élite waiter. Our experiences at the 2001 AAAI *Hors D'œuvres Anyone?* competition uncovered issues which we are currently looking into. These include more dynamic modeling of people, better navigation techniques, and more integrated speech and facial expression interactions.

References

[1] R. C. Arkin. *Behavior-based Robotics*. MIT Press, 1998.

[2] J. Barraquand, B. Langlois, and J. Latombe. Numerical potential field techniques for robot path planning. *IEEE Trans. on Systems, Man, Cybernetics*, 22(2):224–241, March/April 1992.

[3] W. Burgard, A. Cremers, D. Fox, D. Hahnel, G. Lakemeyer, D. Schulz, W. Steiner, and S. Thrun. The interactive museum tour-guide robot. In *AAAI-98*, Madison, Wisconsin, July 1998.

[4] F. Dellaert, W. Burgard, D. Fox, and S. Thrun. Using the condensation algorithm for robust, vision-based mobile robot localization. In *CVPR'99*, Fort Collins, CO, June 1999.

[5] A. Elfes. Using occupancy grids for mobile robot perception and navigation. *IEEE Computer*, 22(6):46–67, June 1989.

[6] P. Elinas. Interactive directed exploration for mobile robots. Master's thesis, University of British Columbia, 2001.

[7] O. Khatib. Real–time obstacle avoidance for manipulators and mobile robots. *Intl. Journal of Robotics Research*, 5(1):90–98, Spring 1986.

[8] J. Lengyel, M. Reichert, B. Donald, and D. Greenberg. Real–time robot motion planning using rasterizing computer graphics hardware. In *SIGGRAPH*, Dallas, Texas, Aug. 1990.

[9] D. Lowe. Object recognition from local scale-invariant features. In *ICCV-99*, Kerkyra, Greece, September 1999.

[10] L. Meeden, B. Maxwell, N. S. Addo, L. Brown, P. Dickson, J. Ng, S. Olshfski, E. Silk, and J. Wales. Alfred: The robot waiter who remembers you. In *AAAI Workshop on Robotics*, Orlando, FL, 1999.

[11] D. Murray and J. Little. Using real-time stereo vision for mobile robot navigation. *Autonomous Robots*, 8:161–171, 2000.

[12] M. Okutomi and T. Kanade. A multiple–baseline stereo. *IEEE PAMI*, PAMI–15(4):353–363, Apr. 1993.

[13] B. Reeves and C. Nass. *The media equation*. Cambridge University Press, 1996.

[14] S. Se, D. Lowe, and J. Little. Local and global localization for mobile robots using visual landmarks. In *IROS-01*, Maui, Hawaii, October 2001.

[15] S. Se, D. Lowe, and J. Little. Vision-based mobile robot localization and mapping using scale-invariant features. In *ICRA-01*, Seoul, Korea, May 2001.

[16] R. Simmons and S. Koenig. Probabilistic robot navigation in partially observable environments. In *IJCAI-95*, San Mateo, CA, 1995. Morgan Kaufmann.

[17] S. Thrun, W. Burgard, and D. Fox. A probabilistic approach to concurrent mapping and localization for mobile robots. *Machine Learning and Autonomous Robots (joint issue)*, 31(5):1–25, 1998.

[18] S. Thrun, W. Burgard, and D. Fox. A real-time algorithm for mobile robot mapping with applications to multi-robot and 3d mapping. In *ICRA-00*, San Francisco, CA, April 2000.