Towards Semantic Perception for Mobile Manipulation

C. Choi[1], J. Huckaby[1], J. G. Rogers III[1], A. J. B. Trevor[1], J. P. Case[2] and H. I. Christensen[1]

[1] Robotics & Intelligent Machines
College of Computing
Georgia Institute of Technology
Atlanta, GA 30332, USA

[2] Evolution Robotics
1055 E. Colorado Blvd., Suite 410
Pasadena, CA 91106, USA

{cchoi|jgrogers|atrevor|hic}@cc.gatech.edu, huckaby@gatech.edu, pcase@evolution.com

***Abstract*— Personal service robots will need to understand semantic object relationships and task context in order to assist humans in their everyday lives. This paper will demonstrate a technique using keywords, spatial relationships, colors, and other contextual information to assist in the mobile manipulation and object recognition tasks. Preliminary results using a mobile manipulation platform are also presented.**

I. Introduction

Personal service robots will need sophisticated object understanding to be useful in real world domestic environments. When interacting with humans, it will be helpful for robots to be able to understand semantic and contextual information regarding objects. For example, if a robot needs to retrieve a cup from a table that contains many cups, some additional information is needed to determine which one the robot should retrieve. This might take the form of a spatial semantic relationship, such as "get the cup on the left", or a qualitative relationship "get the yellow cup". Enabling a robot to utilize this type of information allows it to receive requests that are sufficiently specific to allow it to complete its task, without needing to resort to requests such as "get cup ID #12". In this paper, we propose a technique using keywords, spatial relationships, colors, and other contextual information which enables the robot to reason about which object it is meant to retrieve or operate on in ambiguous situations.

Some relevant related works will be presented in section II. The mobile robot used in this paper and the algorithms developed for this workshop are explained in section III. Some preliminary experimental results are presented in sections IV, V, and VI. Finally, future works are explored in section VII.

II. Related Work

Several recent mobile manipulation systems from the literature include some reasoning about semantic information. The STanford AI Robot (STAIR) is a personal service robot research platform which makes use of monocular and stereo vision [16]. STAIR uses foveated vision to focus a high resolution pan-tilt-zoom camera on interesting regions seen in the wide angle camera. STAIR also evaluates candidate objects based on spatial semantic properties such as the stapler is on the table, or the door handle should be found in one of a few areas once the door is identified [12] [15].

The domestic service robot "Domo" [6] uses behavior based control and human robot interaction to accomplish object retrieval tasks. Domo is equipped with a vision system which is able to roughly estimate the depth to objects of interest. The use of series-elastic actuation makes this robot able to reach out and touch environmental features in order to refine their depth estimate. Domo also considers visual properties of the objects in the environment in order to figure out how to grasp them.

NASA's Robonaut [2] is a mobile manipulation platform designed to assist astronauts with their duties in space. Through the use of perspective taking [19], this robot is able to be more effective in handling the instructions it is given in multiple frames of reference. Spatial constraints and occlusion offer more cues to disambiguate which object to which the human is referring when the robot takes the person's perspective into account.

In the recognition and tracking of 3D objects, various approaches have relied on natural features such as edges, corner points, or keypoint descriptors. One of the advantages of using edges is that they are relatively computationally efficient and moderately invariant to viewpoint and illumination. In addition, edge features are quite common in human environments, and provide strong tracking cues. With prior knowledge of objects, such as 3D CAD models, the RAPiD style methods have shown good results on textureless objects [10], [5], and have been extended to track an object consisting of 3D curves [18]. Similarly, Vacchetti *et al.* [20] used corner features with keyframes, which are 2D reference images of an object model and its pose parameters, to estimate the object pose. The recent approach [8] went further by using the scale invariant feature descriptors [13], and that approach was recently applied to robotic manipulation [4]. Since those natural features have pros and cons, there have been several attempts to combine them. Rosten and Drummond [17] employed both edge and corner features for robust 3D tracking. While they used corner points to estimate motion between two consecutive frames, Vacchetti *et al.* [21] also used corner points to match input images with the closest keyframes. Panin and Knoll [14] combined

contour-based tracking with an initial pose estimation using scale invariant keypoints [13] for fully automatic tracking.

III. Methods

Our personal service robot consists of a variety of hardware and software components. The mobile base is a Segway RMP-200 balancing platform, which is equipped with a SICK LMS-291 laser range finder for localization and obstacle avoidance. A KUKA KR5-Sixx industrial six axis arm is mounted on the top of the Segway base. The end effector we use is a Schunk PG-70 parallel jaw gripper with custom compliant fingers. A PointGrey Flea firewire camera is mounted on the top of the end effector. The robot is shown in figure 1. We have developed our software in the Microsoft Robotics Studio (MSRS) environment [11], now known as Microsoft Robotics Developer Studio.

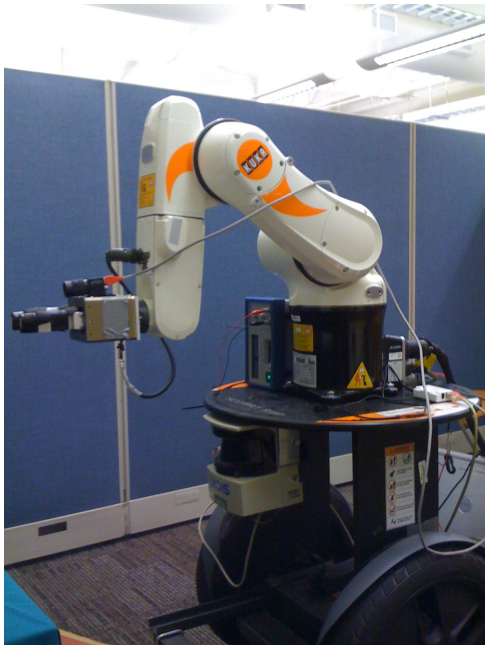


Fig. 1: The robot system used for this work.

A. Localization and Mapping

Localization is a key component for a mobile robot to work and function in the real world. For both mapping and localization, the approach that we use is a grid-based Rao-Blackwellized particle filter, based on the system proposed by Hähnel et. al [9]. The system works by using a particle filter to generate grid-based maps from wheel odometry and laser scans. These are used to estimate the map that best fits the data. Several consistent map hypotheses are maintained by the filter, while inconsistent maps are pruned.

B. Manipulation

The KUKA KR5-Sixx arm is controlled through KUKA's Remote Sensor Interface (RSI), which allows for velocity control in operational space. The manipulation task is carried out in the global frame of reference. Our mobile robot base is dynamically balanced; this results in dynamic motion at the end effector. This motion can be compensated for with tight visual feedback as long as the object is in view of the wrist camera. Unfortunately, our gripper configuration often occludes the camera's view of the object before the grasp is completed, so the final reach must be completed without visual feedback. The robot has typically been able to achieve some moderately reliable performance despite this shortcoming. To improve the performance, the motion of the mobile base is incorporated into the visual servoing computation. By maintaining the pose of the object in the global reference frame, our robot can now compensate for the slight motion of the base to keep the grasp on target even when the visual tracking is no longer possible.

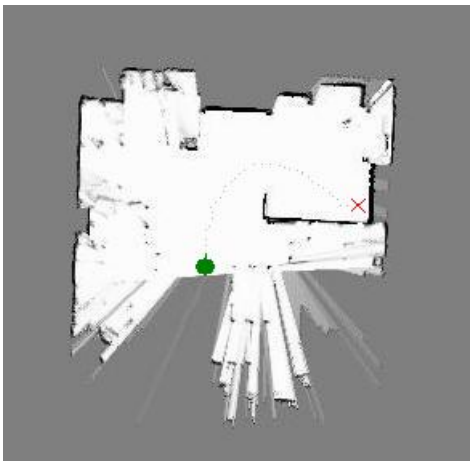


Fig. 2: A map of the environment showing the robot, target location, and path traveled.

Equation 1 shows the transformation for the pose of the object in world coordinates (P_w),

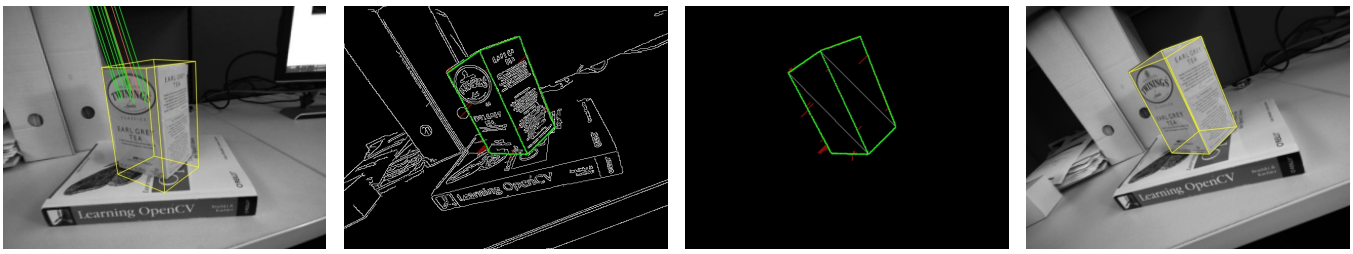
$$P_w = T_s^w * T_b^s * T_s^k * T_c^t * P_c \tag{1}$$

where (P_c) is the object pose in the camera frame, (T_c^t) is the end effector frame transform, (T_s^k) is the robot arm kinematic configuration, (T_b^s) is the platform base frame transform, and (T_s^w) is the world coordinate frame transform. (P_w) is then used in Equation 2 to determine the desired configuration of the arm.

$$T_s^{k*} = T_b^{s-1} * T_s^{w-1} * P_w * T_c^{t-1} \tag{2}$$

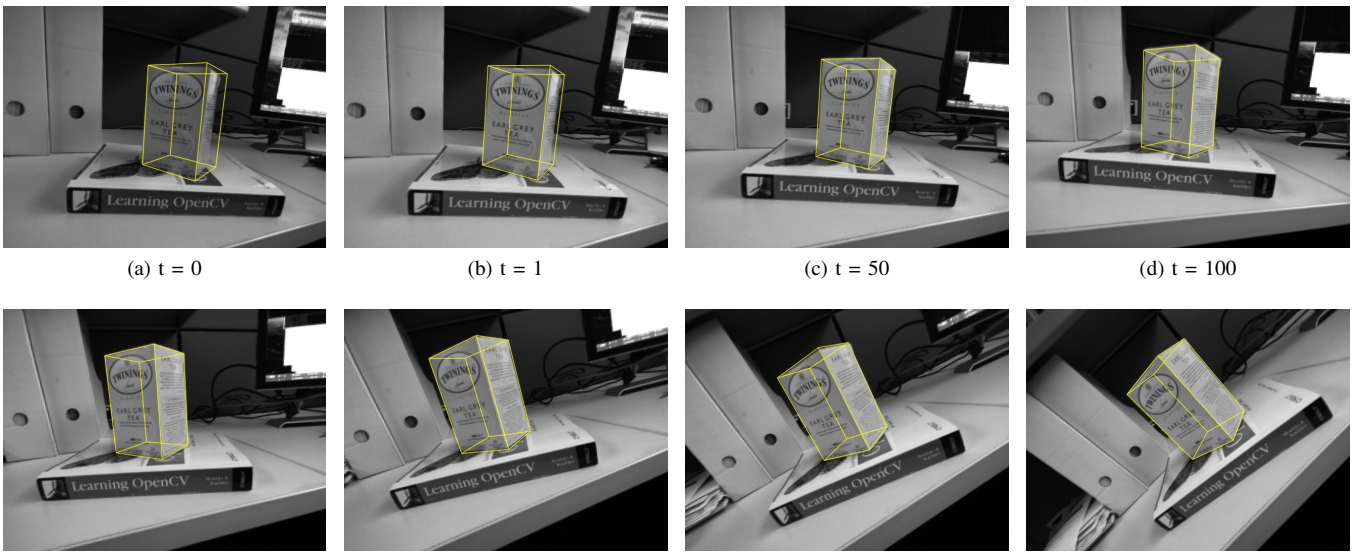
C. Object Recognition and Tracking

One of the key components of a mobile manipulation system is recognition and tracking of objects. The keypoint and descriptor based methods [13], [1] which have been rapidly improved over the last decade have opened up the possibility of using them in visual servo control. However, these state-of-the-art schemes are still computationally expensive, making it difficult to directly apply them to visual servo control. Therefore, we utilize a combined approach that uses SURF features [1] for the global pose estimation and then relies on edge features for the local pose estimation.



(a) Estimated pose (global) (b) Extracted edges (c) Projected 3D model (d) Estimated pose (local)

Fig. 3: Global and local pose estimation results. (a) The initial pose is estimated by the SURF keypoints matching in the global pose estimation. (b) After estimating the initial pose, our algorithm use edges for the local tracking. (c) Since the algorithm already knows the initial pose, it projects the 3D CAD model and generates sample points along the model. (b) and (c) The green points are the generated sample points which are used in calculating errors (the red lines) between 3D sample points and the closest 2D edges. (d) Note that the result of the local pose estimation is more accurate than that of the global pose estimation.



(a) t = 0 (b) t = 1 (c) t = 50 (d) t = 100

(e) t = 150 (f) t = 200 (g) t = 250 (h) t = 300

Fig. 4: Tracking results over 300 frames. The yellow parallelepiped shows the estimated pose of the tea box object. (a) At t = 0, the pose is estimated by the global pose estimation. (b) After one execution of the local pose estimation, the pose starts to converge to the real pose. Our algorithm can converge after several frames. Since the local pose estimation relies on the 1D search along the normal direction of each sample point, it is much faster than the global pose estimation.

1) Global Pose Estimation: In the global pose estimation, our algorithm calculates an initial pose of the object. Like [20], we obtain keyframes offline. Using the SURF keypoint matching, the keyframes are compared with the current image in order to estimate the pose. The estimated pose is then refined by using RANSAC [7]. Fig. 3a shows the estimated pose from the global pose estimation. The initial pose estimate is accurate enough to initialize the local pose estimation procedure.

2) Local Pose Estimation: Once an initial pose of the object is estimated, edge features are used in our algorithm that is based on Drummond and Cipolla's work [5] for faster pose estimation. First of all, we extract edge features by using the Canny edge detector [3] as shown in Fig. 3b. Since we already know the initial pose of the object from the global pose estimation, we project the 3D CAD model onto the current image (Fig. 3c). Sample points are generated along sharp edges, which are determined from the 3D CAD model offline.[1] The sample points are depicted as green points in Fig. 3b and Fig. 3c. Point visibility is determined through an OpenGL occlusion query. After searching the closest edges on the sample points along the normal direction, the final 6 DOF pose parameters are estimated by weighted least squares [5] (Fig. 3d). Fig. 4 shows the tracking results of our algorithm. Note that although the initial pose is not perfect,

[1]We use the face normal vectors from the 3D CAD model to determine sharp edges. For example, if the face normal vectors of two adjacent faces are close to perpendicular, the edge shared by the two faces is regarded as a sharp edge. Similarly, if two face normal vectors are close to parallel, the edge is regarded as a dull edge.

the local pose estimation enables the tracker to converge to a consistent pose.

For the model format, we use polygon mesh models which can be modeled by using a standard 3D modeling tool such as Blender. Since we automatically determine sharp edges in a polygon mesh model by using the aforementioned method, we can handle any shapes of objects even having smooth surfaces.

While the global pose estimation searches all of the possible keypoint matches over all keyframes and the current image, the local pose estimation only relies on the 1D search along the normal direction of each sample point. Therefore, the local pose estimation is much faster than the global pose estimation.[2] However, since the edges the local pose estimation uses are not descriptive enough, it may lead to spurious results when the algorithm is stuck in local minima. To recover from this, we monitor tracking results of the local pose estimation, and if the tracker loses the object, it reruns to the global pose estimation.

D. Semantic Reasoning

To begin investigating how one might build a system that is aware of semantic relationships such as those described in section I, we devised a mobile manipulation scenario in which a user can request an object, and optionally specify some additional semantic property of the object in order to disambiguate it from other objects of the same type. In our scenario, the objects used are two boxes of tea which are placed side by side on a table. Because these objects are both boxes of tea, if a user requests that the robot find a tea box, it will need to make use of additional semantic or contextual information to determine which one the user intended.

As a first step towards making robots that can reason about this type of semantic relationship between objects, we designed a system capable of reasoning about one of the simplest semantic relationships: horizontal spatial arrangement (left vs. right). If the system is presented with multiple objects of the same class (for example, 'cups' or 'boxes'), it can be instructed to indicate the leftmost or rightmost object. This is accomplished by searching for all objects of a given class. If multiple objects of a given class are detected, either the leftmost or rightmost object is selected, based on the user's request.

One might also want a robot to be aware of the time of day, and behave differently with respect to this contextual information. For example, if a user requests some tea, the robot could be made aware that it is inappropriate to bring the user caffeinated tea after a certain hour. Our system stores some object properties along with its object models, to which we added whether or not the object in question contains caffeine. If a user doesn't specify otherwise, the system will bring the user caffeinated tea if the current time is before 4pm, but decaffeinated tea after 4pm.

[2]In our system, the global pose estimation takes about 200 ms per frame (5 Hz). The local pose estimation requires less than 50 ms per frame (20 Hz).

Another important semantic property of objects is color. People often use color as a way to distinguish objects from one another, so it is helpful for robots to be able to understand what a user means if they request 'the orange mug'. Our system was modified to be able to distinguish between objects based on their dominant color. For this simple example, we chose a point in hue-saturation color space that is representative for colors such as 'orange' and 'yellow'. Objects can be classified as having one of these color attributes by making a hue-saturation histogram of the object model image, and comparing these against the manually defined reference colors.

IV. Experiments

Although extensive experiments are beyond the scope of this paper, some preliminary tests were performed in order to evaluate the system. We set up a simple task for our mobile manipulation platform to test the semantic reasoning portion of our system. In the experiment, a user indicates an object of interest, and specifies the object's location. The user can optionally specify additional semantic information such as the properties of 'left' or 'right', which can disambiguate two objects of the same type. The user interface to our system is a web page, which includes drop down lists of known objects, locations, and additional optional semantic information (left, right, orange, yellow, etc). The robot started in the middle of our lab, then drove to a nearby table that it was told contained the objects of interest. Upon arrival, the vision system would detect and track the two boxes on the table, and the robot would servo to and "point at" either the left or right box. We ran six trials, three for the left object and three for the right object, which were all successful. Because our grasp controller is one of the less reliable portions of our system, we chose to point at an object rather than actually grasping it, so as to focus on testing the robot's perceptual ability rather than its grasping ability. The robot can be seen in figure 5 successfully pointing at the left tea box.

Fig. 5: The robot successfully points at the left tea box.

V. Discussion

The system demonstrated is shown to work reliably for distinguishing semantic relationships with several differ-

ent objects in relatively simple environments. While our approach succeeds in guiding the perception of semantic properties with a small number of objects, it would not scale well to a large number of objects and object types, as the number of semantic relationships needed to accurately describe a very large group of objects would be considerable.

While the threshold-based determination for time of day in the selection of objects is simple, it serves to demonstrate another possible use of semantic properties to disambiguate between user requests. One could imagine a system where simple properties such as these are learned and determined through experiences with user preference.

Although we have investigated several ways that semantic and contextual information can be applied to mobile manipulation systems, there are many more opportunities for the use of this type of information. Additional semantic information of the types demonstrated here could clearly be added to the system, for example allowing the robot to know about additional spatial semantic relationships such as 'top' vs. 'bottom', or 'front' vs. 'back'. Other examples of qualitative semantic relationships could include shape (square, round, triangular, flat), size (large, small, smallest).

Our system currently allows for labels to be attached to both places and objects (or sets of objects). One could imagine adding functionality to the system to allow it to know what types of objects it should expect to find in different places. For example, one might find mugs either in the kitchen, or on people's desks.

While the focus of this paper has been on using semantic information to inform the perceptual system, we think that this type of information could also be used to inform robot control. For example, if we know that we are grasping a square object, this implies that our controller should ensure that the fingers grasp the sides of the object, rather than the corners. However, if the object is known to be cylindrical, we don't have this type of constraint.

VI. Conclusion

In this paper we presented a system that demonstrated semantic perception in mobile manipulation tasks. Our mobile manipulator was given the task of identifying specific objects based on semantic relationships. When the system navigated to the specified object location, model-based techniques were used for object recognition and tracking. Once objects were being tracked at the specified location, the robot was successfully able to distinguish the semantic relationship between the objects (the 'left' object versus the 'right' object) and correctly identify the desired object. In a separate experiment, the system was able to successfully disambiguate between objects based on outside knowledge, the time of day, demonstrating basic contextual reasoning.

VII. Future Work

Our group is exploring the use of new hardware and software components for mobile manipulation tasks. We will be switching to the DLR/KUKA LBR arm to take advantage of its additional workspace as well as its payload capacity. The LBR also can be powered with batteries, which is impossible with the KUKA industrial controller. We are assembling a 3D laser scanner to use for the object recognition and grasp planning routines.

VIII. Acknowledgments

The authors gratefully acknowledge the contribution of KUKA Robotics for loaning us the KR5-Sixx robot arm.

References

[1] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool. Speeded-up robust features (SURF). *Computer Vision and Image Understanding*, 110(3):346–359, 2008.

[2] W. Bluethmann, R. Ambrose, M. Diftler, S. Askew, E. Huber, M. Goza, F. Rehnmark, C. Lovchik, and D. Magruder. Robonaut: A robot designed to work with humans in space. *Autonomous Robots*, 14(2):179–197, 2003.

[3] J. Canny. A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence*, pages 679–698, 1986.

[4] Alvaro Collet, Dmitry Berenson, Siddhartha S. Srinivasa, and Dave Ferguson. Object recognition and full pose registration from a single image for robotic manipulation. In *Robotics and Automation, 2009. ICRA '09. IEEE International Conference on*, pages 48–55, 2009.

[5] T. Drummond and R. Cipolla. Real-time visual tracking of complex structures. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(7):932–946, 2002.

[6] A. Edsinger and C.C. Kemp. Manipulation in human environments. In *Humanoid Robots, 2006 6th IEEE-RAS International Conference on*, pages 102–109, 2006.

[7] Martin A. Fischler and Robert C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, 1981.

[8] Iryna Gordon and David Lowe. What and where: 3D object recognition with accurate pose. In *Toward Category-Level Object Recognition*, pages 82, 67. Springer, 2006.

[9] D. Hahnel, W. Burgard, D. Fox, and S. Thrun. An efficient fastslam algorithm for generating maps of large-scale cyclic environments from raw laser range measurements. In *Intelligent Robots and Systems, 2003. (IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*, volume 1, pages 206–211 vol.1, Oct. 2003.

[10] C. Harris. *Tracking with Rigid Objects*. MIT Press, 1992.

[11] J. Jackson. Microsoft robotics studio: A technical introduction. *IEEE Robotics & Automation Magazine*, 14(4):82–87, 2007.

[12] E. Klingbeil, A. Saxena, and A. Ng. Learning to open new doors. *RSS workshop on robot manipulation*, 2008.

[13] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[14] Giorgio Panin and Alois Knoll. Fully automatic real-time 3d object tracking using active contour and appearance models. *Journal of Multimedia*, 1:62–70, 2006.

[15] M. Quigley, S. Batra, S. Gould, E. Klingbeil, Q. Le, A. Wellman, and A.Y. Ng. High-Accuracy 3D Sensing for Mobile Manipulation: Improving Object Detection and Door Opening. 2009.

[16] M. Quigley, E. Berger, and A.Y. Ng. Stair: Hardware and software architecture. In *AAAI Robotics Workshop*, 2007.

[17] E. Rosten and T. Drummond. Fusing points and lines for high performance tracking. In *Tenth IEEE International Conference on Computer Vision, 2005. ICCV 2005*, volume 2, 2005.

[18] G. Simon and M. O Berger. A two-stage robust statistical method for temporal registration from features of various type. In *International Conference on Computer Vision*, page 261266.

[19] JG Trafton, NL Cassimatis, MD Bugajska, DP Brock, FE Mintz, and AC Schultz. Enabling effective human-robot interaction using perspective-taking in robots. *IEEE Transactions on Systems, Man and Cybernetics, Part A*, 35(4):460–470, 2005.

[20] L. Vacchetti, V. Lepetit, and P. Fua. Fusing online and offline information for stable 3D tracking in real-time. In *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings*, volume 2, 2003.

[21] L. Vacchetti, V. Lepetit, and P. Fua. Combining edge and texture information for real-time accurate 3d camera tracking. In *Third IEEE and ACM International Symposium on Mixed and Augmented Reality, 2004. ISMAR 2004*, pages 48–56, 2004.