Outlet detection and pose estimation for robot continuous operation

Victor Eruhimov and Wim Meeussen

Abstract— **We describe an algorithm for detection of electrical outlets in images obtained by a monocular camera. We provide a method for calculating 3D coordinates of outlet holes with accuracy high enough for a robot to plug in without visual servoing. The paper proposes a novel algorithm for accurate pose estimation of a small planar object. We use the plane normal obtained from stereo data as a hard constraint for a planar PnP problem. A different cost function is proposed together with a closed-form solution for the object pose. This system is used for continuous operation of a PR2 robot enabling it to run for 2 weeks without human supervision.**

I. INTRODUCTION

The purpose of this work is to develop a system that enables a robot to plug itself into a standard electrical outlet. This problem is an important step for mobile robotics towards continuous autonomous operation. Plugging in allows a robot to travel long distances in a building and use the nearest outlet when the battery charge is low.

A charging station is commonly used for continuous operation in household robotics [1], [2]. Such a station occupies space and there always should be a clear path to it. A robot in a large building might have to travel a long path to find a free station to charge. Also, docking stations might fail to provide an electrical connection if a robot is not positioned properly when parking. A standard electrical outlet design is much more robust, being tested by decades of usage. Outlets are widely available in the most part of home and office buildings. This allows a robot to travel less distance when looking for an unobstructed path to an electrical connection.

The problem of plugging into a standard outlet is challenging for both perception and manipulation because it requires accurate, submillimeter-scale positioning of the plug with respect to the outlet. In the final step of the plug-in operation a plug is close to an outlet and occludes a part of it, making visual servoing difficult. There are several robots [3], [4] that use electromagnetic sensor to localize outlet holes. Our goal is to solve this problem using only a general-purpose vision system.

The plugging in problem can be factored into estimating coordinates and orientation of outlet holes with regard to the robot using its cameras and moving the arm with the plug into the corresponding position. This paper addresses the perception part of the problem: how can we reliably detect outlet holes and find their coordinates with submillimeter accuracy? We assume that the robot is already in front of the outlet. [5] describes a farfield outlet detection using stereo and template matching, helping to drive a robot close to an outlet.

We use a local feature detector to find hole candidates in a monocular image that contains an outlet. Some of the false alarms are filtered out by one way descriptor [6] that is trained on a single image of a different outlet. Then we use geometric hashing [7] to select a subset of hole candidates that will be considered an outlet. Once we have a hypothesis about hole locations in the image, we can run a planar PnP solver to reconstruct 3D locations of all points. However since the outlet size is small (a distance between outlet power holes is about 12 mm) while the distance from the outlet to the camera is large (20-50 cm), the relative error of the pose estimation will also be large.

In order to address this problem we introduce a new algorithm for solving a planar PnP problem with a known object normal. We define a different cost function that minimizes the reprojection error in the object plane rather than in the camera plane. It has a single minimum that can be found in closed-form and is more stable against noise in point coordinates. Planar PnP problem needs 4 non-collinear points while our method gives a unique solution with only 2 points.

II. RELATED WORK

There are several approaches for detecting frontal view outlets in images. [8] uses Viola-Jones cascade detector [9], while [10], [11] rely on template matching. [5] describes an algorithm for detecting a block of 2x2 orange outlets using color blob detection. The technique showed good detection accuracy with varying viewangle and lighting conditions. Pose estimation was done by a PnP solver in OpenCV [12] that uses Levenberg-Marquardt optimization. Since the pose was calculated from only 4 points (centers of orange blobs), the results were not stable enough and iterative random search was employed for a successful plugging in.

Recently there was quite a progress in detecting textured objects. A number of feature detectors (SIFT [13], SURF [14], MSER [15], FAST [16], [17]) and descriptors [13], [14], [18] are used to build a generic textured object detection engine using a monocular camera [19], [20], [21]. Descriptors are used to find candidate object features in a test image. Then a geometrically consistent subset of features is found to filter out false alarms. [22], [23] use histograms of intensity gradients to detect untextured objects.

There is quite a lot of work on PnP problem solvers, both closed-form and iterative [24], [25], [26], [27], [28].

This work was not supported by any organization

Victor Eruhimov is with Itseez, Nizhny Novgorod, 603000, Russia and Willow Garage, CA 94025, USA victor.eruhimov@itseez.com

Wim Meeussen is with Willow Garage, CA 94025, USA meeussen@willowgarage.com

For a planar object, it is possible to calculate the pose from the homography between object and image planes (see, for instance, [29]). There are quite a few algorithms specially designed for planar objects [30], some of the general-purpose methods consider planar PnP as a special case [24]. However, since few points are available for fitting (6 holes for a block of 2 outlets), such methods will be sensitive to noise. So, any prior information will be crucial at this step. The authors are not aware of any attempts to use an object normal as a prior in a planar PnP solution.

We want to develop a technique that robustly detects outlets from different viewpoints and in different lighting conditions. Template matching is not stable to variations in lighting and cascade detector fails with a change in viewpoint. Our algorithm follows ideas of [13], combining local feature matching with geometrical validation. The next section describes the algorithm for detecting outlet holes in an image. Then we describe our approach to pose estimation and conclude by talking about the Continuous Operation project.

III. OUTLET DETECTION

We assume that the robot has already positioned itself near an outlet using the approach outlined in [5]. In order to plugin, we need to detect outlet holes and find their 3D positions. Detection is based on a single image taken by a camera that is $20-50cm$ away from the outlet. In order to be robust we detect a block of several outlets with a known 3D structure, such as shown in Figure 1a. The detection algorithm consists of several steps: finding the candidate holes, classifying each of them using their likelihood to the holes of a previously captured outlet, and finally finding a geometrically consistent combination of holes. Each of these steps is covered by the following subsections.

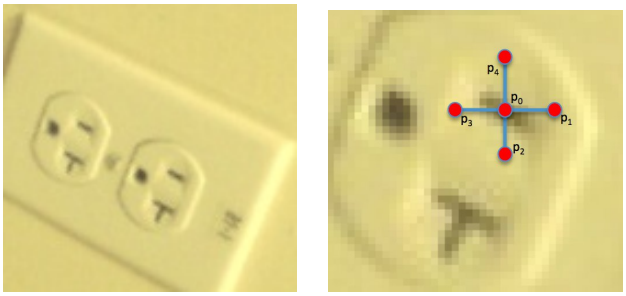
A. Hole detection

The goal of this step is to find pixels whose local regions look like outlet holes. We want the detector to work with 100% detection rate and an acceptable false alarm rate. That is, if there is an outlet in an image, every outlet hole should be detected, at the expense of also returning locations that are not outlet holes. We look for any small dark areas surrounded by larger brighter areas. Due to shadows and glossy surfaces outlet hole appearance can change a lot with a viewangle. We have designed a feature detector specifically for this task. Other detectors such as [14], [15] or multiscale FAST [16] are also acceptable to be used here but our experiments showed them to give worse performance on this specific problem.

The algorithm for detecting outlet holes is based on finding small connected components after thresholding an image on different levels (we use threshold values from 30 to 150 with step 20 for a greyscale image). For each component C we need to check if it is surrounded by a brighter area. We do this by comparing the intensity in the geometrical center $\mathbf{p_0}$ of C with intensities in four pixels around C: $\mathbf{p_{1,2}} \pm s_x\mathbf{x^0}$ and $\mathbf{p_{3,4}} \pm s_y\mathbf{y^0}$, where s_x and s_y are the width and height of the bounding rectangle for the component C, $\mathbf{x^0}, \mathbf{y^0}$ are vectors along the horizontal and vertical axes with the length equal to 1 pixel (see Fig 1b). The contrast is defined as

$$r_C = \frac{\sum\limits_{i=1..4} I(\mathbf{p_i})}{4I(\mathbf{p_0})}, \tag{1}$$

where $I(\mathbf{p})$ is intensity in pixel $\mathbf{p}$. r_C is compared with a constant threshold (equal to 1.5 for the all of the experiments).



(a) (b)

Fig. 1. (a): Outlet (2x1) training image used for Continuous Operation project. (b): Fast estimation of local contrast. The intensity in the central pixel p_0 is compared with the average of intensities in $p_{1..4}$ (1).

B. Classification

Given the detected features, we classify them into three different classes: power holes, ground holes and background. Since the regions around the holes do not have any texture, the best method is intensity-based (as opposed to gradient-based methods such as SIFT and Haar-based SURF descriptor). We use one way descriptor [6] that takes each training patch and creates a set of its affine distortions simulating viewpoint variation. We run affine distortions on 24x24 pixels patches around each hole in the training set, then crop the center area of 12x12 pixels. In both operations we work in 100-dimensional PCA space.

The training set consists of a single image of an outlet with power and ground holes manually labeled, and a single background image. Figure 1a shows the outlet training image used for the Continuous Operation project. For each hole candidate from a test image we crop a 12x12 pixels area around it, and find the nearest neighbor among all affine distorted patches in the training set using Euclidean distance in the PCA space. A test feature is assigned the class of its corresponding nearest neighbor. In order to account for different sizes of train and test outlets, we run the algorithm on scaled test images. For PR2 plugin we use scales from 0.5 up to 2 with a multiplicative step equal to 1.15. Examples of classified hole candidates are available in Figure 2.

C. Geometric filtering

The final step in the detection algorithm is to select a geometrically consistent set of hole candidates. If $\mathbf{h_i}$ are pixel coordinates of test features and $\mathbf{t_j}$ – of labeled holes

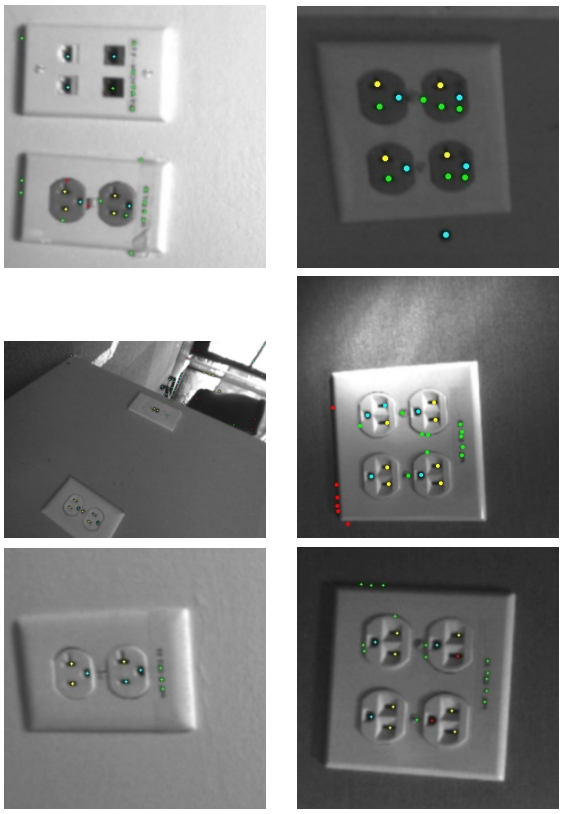


Fig. 2. One way descriptor results. Blue circles correspond to holes classified as ground, yellow – as power, green – non-hole features from training outlet image, red – background. The left column contains images of 2x1 outlets, the right – 2x2.

in the training image, we can define a score for each affine transform $\mathbf{A}$ from training to test image:

$$S = \sum_j \delta\left(\exists i || \mathbf{t}_j - \mathbf{h}_i| < MaxDist \wedge c(\mathbf{t}_j) = c(\mathbf{h}_i)\right). \tag{2}$$

Here $\delta(x) = 1$ if x is true, 0 otherwise, $c()$ is feature class. We look for an affine transform that has a score higher than a threshold S_{max}. For 2x1 outlets used in continuous operation we use $S_{max} = 4$, that is, we need to find a geometrically consistent solution with 4 holes out of 6, correctly classified by one way descriptor.

We restrict ourselves to a class of affine transformations that map 3 points from the training set exactly into 3 points in the test set. Geometric hashing [7] is used to make the search fast. We construct a hash table by considering all ordered combinations of 3 labeled points $\mathbf{t}_j$ in a training set. Each combination corresponds to a 2-dimensional affine basis with the origin $\mathbf{o}$ in the first point and two base vectors $\mathbf{a}$, $\mathbf{b}$ directed from the origin to the other two points. An arbitrary point in the training set can be represented as $\mathbf{t}_j = p_j\mathbf{a} + q_j\mathbf{b}$. We discretize the space of coefficients p_j, q_j, building a hash table and assigning an integer list to each of the table bins. For each point $\mathbf{t}_j$ generating a pair $\{p_j, q_j\}$ that falls into a specific hash bin, we add the current basis index to the list of this bin. We repeat this for each basis and for each point in the training set.

During the online stage, we construct bases from triples of test features and use the hash table to find possible matches from the training set. We iterate through all test features. For each of them, we randomly select two more features in the proximity of the current one. If the selected points are almost collinear, we discard the triplet and repeat the selection procedure again. For a selected triple, we build a basis and represent each point as a linear combination of its vectors. Thus we can map each test point to a hash bin and count the number of times we meet each of the training bases in it. These counts are accumulated for all test points, each count being the measure of how likely is the representation of test points in the test basis and training points in the training basis. A pair of training and test bases generates a unique affine transform $\mathbf{A}$. We calculate the score (2) for each of the training bases that get the count higher than a threshold. A solution with the maximum score is returned, if the score is higher than S_{max}. Otherwise we conclude there is no outlet in the test image.

IV. EXPERIMENTAL RESULTS

Our training dataset consists of a single image of an outlet, a single background image and 3D coordinates of each outlet hole with one of the holes in the origin. Therefore, we use different training sets for 2x1 and 2x2 outlets and we know in advance what structure we are looking for. Obviously this condition can be relaxed by subsequent application of the detection with different training sets if the number of false alarms for each of the structures is sufficiently low. Figure 3 contains examples of images from the test dataset. Detection time varies from 0.15s to 1s for 2x1 outlets and from 0.4s up to 4s for 2x2 outlets. The most of the time is spent on classifying holes on different scales.

We test outlet detection on a dataset that consists of 3 different 2x1 and 4 2x2 outlets. Each image consists of exactly one outlet. We know coordinates of each hole and we consider the detection successful if the detected position is off from the ground truth by less than 5 pixels. The detection function either returns exactly one solution or no solution, so the quality is estimated with the correct detections count and false alarms count. We give absolute numbers of both since the number of test samples in each dataset is relatively low.

Dataset	# images	# detections	# false alarms
fw_2x1white	111	88	0
orange2x1	103	94	0
whitehall2x1	103	100	0
cracked2x2	100	91	0
fw_2x2orange	34	33	0
specular2x2	56	21	0
white2x2	104	97	0

TABLE I
OUTLET DETECTION RESULTS.

Table I contains the detection results for all datasets. The algorithm is tuned so that it misses some of the objects but has almost no false alarms. It is important from the

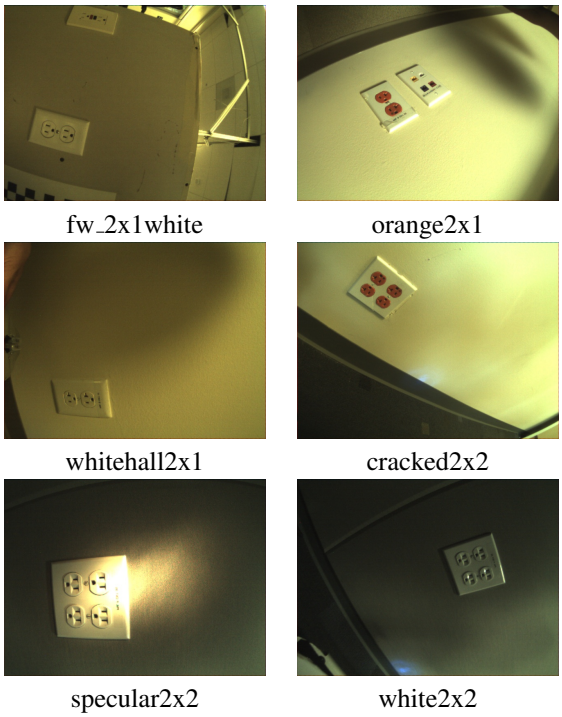


Fig. 3. Examples of images from test datasets.

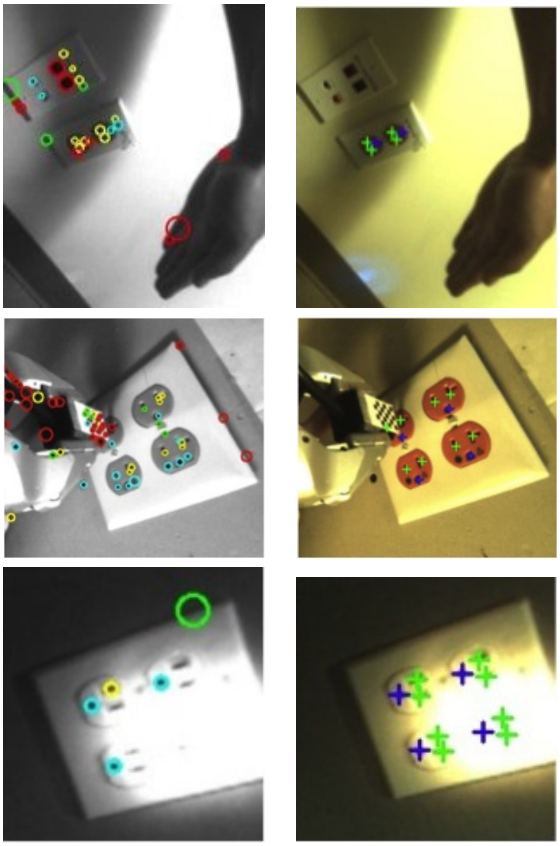


Fig. 4. Examples of challenging detections. The left column contains output of one way descriptor. The right column contains detection results.

application point of view: we rather change camera position to try detection on another viewpoint than provide a robot arm with wrong coordinates. The most of the failures are on side views of outlets and are due to misclassification of outlet holes. For PR2 plugging in, we try to position the camera exactly in front of the outlet so that the detection is more stable. Note that a single template is used for detecting outlets of different color. This is because all steps of the algorithm are invariant to linear scaling of pixel intensities. *specular2x2* dataset contains the most challenging examples of outlet images: strong shadows, oversaturated or dark images. This explains relatively low detection rate. Figure 4 shows some of the successful detections in these challenging conditions. The test dataset is available from [31], the code for running detection on this data is in ROS package outlet_test [32].

V. POSE ESTIMATION

Estimation of 3D positions of outlet holes from a monocular image is a hard problem. The outlet is a planar object so the problem can be formulated as a planar PnP, where the cost function is defined as a reprojection error to the camera plane. However, if the object linear size is much smaller than the distance to the camera, then a large variation in object pose will cause a small variation in the reprojection error. Moreover, for an object that is symmetric such as a 2x1 outlet, there are going to be 2 solutions corresponding to different poses.

In order to address these issues, we introduce a hard constraint on the outlet orientation coming from 3D sensors such as stereo camera or lidar that estimate wall normal. Also, we redefine a pose estimation cost function: instead of minimizing the reprojection error in the camera plane, we minimize the reprojection error in the object plane. This function is more sensitive to changes in object pose although we still get a significant error from hole coordinates in the image plane. Also, it appears that we can minimize the reprojection error inside the object plane in closed form.

Let us consider a set of m points $\{\mathbf{p_i^0} \in R^3\}$ in an arbitrary reference frame, all lying in the same plane. An unknown orthogonal transformation described by a rotation matrix $\mathbf{R}$ and a translation vector $\mathbf{T}$ is applied to bring the points into the camera reference frame. The points after transformation will be denoted as $\{\mathbf{p_i} \in R^3\}$. We observe the projections $\mathbf{k_i} \in R^3$ of $\mathbf{p_i}$ into the camera plane. Our goal is to recover $\mathbf{R}$ and $\mathbf{T}$ given $\{\mathbf{p_i^0}\}$, $\{\mathbf{k_i}\}$ and the normal $\mathbf{n}$ of the outlet plane in the camera reference frame.

Since all $\mathbf{p_i}$ lie in the same plane, they can be represented as

$$\mathbf{p_i^0} = \alpha_i \mathbf{a^0} + \beta_i \mathbf{b^0}, \tag{3}$$

where $\mathbf{a^0}, \mathbf{b^0} \in R^3$, $\mathbf{a^0}$ and $\mathbf{b^0}$ define an orthonormal basis in object plane. Without loss of generality we can assume that object mass center is in the origin so that

$$\sum_i \alpha_i = \sum_i \beta_i = 0. \tag{4}$$

The coordinates of the same point set after an unknown transformation into the camera reference frame can be ex-

pressed as

$$\mathbf{p_i} = \alpha_i \mathbf{a} + \beta_i \mathbf{b} + \mathbf{T}, \tag{5}$$

where $|\mathbf{a}| = |\mathbf{b}| = 1$, $\mathbf{a} \perp \mathbf{b}$.

We observe projections of $\mathbf{p_i}$ into the camera plane $\{\mathbf{k_i} \in R^3\}$. All $\mathbf{k_i}$ lie in the camera plane, so that without observational errors $\mathbf{p_i}$ should lie on the rays going from origin through $\mathbf{k_i}$, so that there exists t_i such that $\mathbf{p_i} = \mathbf{k_i} t_i$. In reality there is a considerably large error in measuring $\mathbf{k_i}$ coming from image processing so that the last equation cannot be fulfilled. A traditional approach to this problem is to minimize the distance between projections of $\mathbf{p_i}$ into camera plane and the corresponding $\mathbf{k_i}$, with respect to $\mathbf{R}$ and $\mathbf{T}$. We suggest a different cost function that measures the distance between $\mathbf{p_i}$ and projections of $\mathbf{k_i}$ into the object plane. We define object plane by the equation $\mathbf{n} \cdot \mathbf{p} = d$ ($\cdot$ denoting scalar product), so that a projection of $\mathbf{k_i}$ into the object plane can be expressed as

$$\mathbf{r_i} = \frac{\mathbf{k}_i d}{\mathbf{k}_i \cdot \mathbf{n}}, \tag{6}$$

where $\mathbf{n}$ is the object normal (provided by stereo or lidar measurements) and d is the distance from the origin to the object plane. Note that since the object is planar, its mass center is also in the same plane and

$$d = \mathbf{n} \cdot \mathbf{T}. \tag{7}$$

We define the cost function as

$$f(\mathbf{a}, \mathbf{b}, \mathbf{T}) = \sum_i \left\| \mathbf{p_i} - \mathbf{r_i} \right\|^2. \tag{8}$$

The pose estimation problem is reduced to finding the minimum of f with respect to the following constraints:

$$\begin{aligned} \mathbf{a} \cdot \mathbf{b} = \mathbf{a} \cdot \mathbf{n} = \mathbf{b} \cdot \mathbf{n} = 0 \\ |\mathbf{a}| = |\mathbf{b}| = 1 \end{aligned} \tag{9}$$

It is easy to show that f has a single minimum that can be found in closed form. Let us find all argument values that make all partial derivaties of f on its arguments equal to zero. We will start with the translation vector:

$$\begin{aligned} \tfrac{\partial f}{\partial T} = 2 \sum_i (\mathbf{p}_i - \mathbf{r}_i) + \\ \mathbf{n} \sum_i (\mathbf{p}_i - \mathbf{r}_i) \cdot \tfrac{\mathbf{k}_i}{\mathbf{k}_i \cdot \mathbf{n}} = 0, \end{aligned} \tag{10}$$

Proposition 1. The only solution for equation (10) satisfies the following:

$$\mathbf{T} = \frac{1}{m} \sum_i \mathbf{r}_i. \tag{11}$$

Proof. First, let us note that if we take a scalar product of both sides of (11) with $\mathbf{n}$ and use (6), we get an identity. Then, if we take a cross product of both sides of (10) with $\mathbf{n}$ and use (5), (4), we obtain

$$\mathbf{n} \times \mathbf{T} = \mathbf{n} \times \frac{1}{m} \sum_i \mathbf{r}_i, \tag{12}$$

This proves the proposition.

Equation (11) defines $\mathbf{T}$ up to a scale:

$$\mathbf{T} = \frac{1}{m} \sum_i \frac{\mathbf{k}_i (\mathbf{n} \cdot \mathbf{T})}{\mathbf{k}_i \cdot \mathbf{n}}. \tag{13}$$

We note that using (11) and (9) we can prove

$$\sum_i (\mathbf{p}_i - \mathbf{r}_i) \cdot \mathbf{n} = 0, \tag{14}$$

Then an equation for scale can be obtained by taking a scalar product of both sides of (10) with $\mathbf{n}$:

$$d = \frac{\sum_i (\alpha_i \mathbf{a} + \beta_i \mathbf{b}) \cdot \frac{\mathbf{k}_i}{\mathbf{k}_i \cdot \mathbf{n}}}{\sum_i \mathbf{q}_i \cdot \frac{\mathbf{k}_i}{\mathbf{k}_i \cdot \mathbf{n}}}, \tag{15}$$

where $\mathbf{q}_i = \frac{\mathbf{k}_i}{\mathbf{k}_i \cdot \mathbf{n}} - \frac{1}{m} \sum_j \frac{\mathbf{k}_j}{\mathbf{k}_j \cdot \mathbf{n}}$.

We see that if $\mathbf{a}$ and $\mathbf{b}$ are defined, the last equation gives a unique solution for d. Now, with the constraints (9), all possible solutions are reduced to a rotation around an axis going through object mass center $\mathbf{T}$ parallel to n. Let us choose an arbitrary basis in the object plane $\mathbf{a}^p$,$\mathbf{b}^p$. For example, $\mathbf{a}^p = \mathbf{n} \times \mathbf{x}^0$, $\mathbf{b}^p = \mathbf{n} \times \mathbf{a}^p$, where $\mathbf{x}^0$ is a unit vector parallel to x axis in the image plane. Then $\mathbf{p}_i$ can be expressed with the new basis:

$$\begin{aligned} \mathbf{p}_i = (\alpha_i \cos(\phi) + \beta_i \sin(\phi))\, \mathbf{a}^p + \\ (-\alpha_i \sin(\phi) + \beta_i \cos(\phi))\, \mathbf{b}^p + \mathbf{T}, \end{aligned} \tag{16}$$

where ϕ is the angle between $\mathbf{a}^p$ and $\mathbf{a}$. Then we can find the optimal value of ϕ. It is easy to see, using (9), that

$$\begin{aligned} \tfrac{\partial f}{\partial \phi} = 2d \sum_i (\mathbf{q}_i \mathbf{b}^p \alpha_i - \mathbf{q}_i \mathbf{a}^p \beta_i) \cos(\phi) + \\ (\mathbf{q}_i \mathbf{b}^p \beta_i + \mathbf{q}_i \mathbf{a}^p \alpha_i) \sin(\phi) = 0. \end{aligned} \tag{17}$$

This equation gives us two solutions for ϕ that differ by π. If d is positive for one of these solutions, then it is negative for another, since both $\mathbf{a}$ and $\mathbf{b}$ change direction to the opposite (see (15)). This gives us a closed-form solution for the minimum point of f with regard to $\mathbf{T}$ and ϕ.

VI. CONTINUOUS OPERATION

For robots to be useful in real-world environments, it is necessary for them to be capable of performing tasks over long periods of time. The ability for a robot to recharge itself is critical to achieve autonomy that lasts longer than a single battery charge. Based on the PR2 robot, we build a robust platform to execute long term autonomous tasks in an office environment, such as mail delivery, cleanup or security monitoring. The base platform has two basic capabilities: autonomous navigation and recharging. We stress test this platform by commanding the PR2 to continuously navigate to randomly chosen locations in the environment. When the battery level runs low, the PR2 navigates to one of three designated recharging locations and plugs itself into a standard 2x1 outlet. While this experiment ran for multiple months, the longest continuous run lasted for 13 days and covered a distance of 138.9 km. In this run the robot had to recharge itself over 100 times. Recharging is triggered at a battery level of 35%, giving the robot 15-20 minutes to

navigate to an outlet and plug itself in, before its batteries run out. When plugging in fails, the robot navigates to a different recharging location to try plugging in again. In practice this means the plugging in task can only fail 1-2 times before the robot dies. During the 13 day continuous run, only 5% of the recharging attempts failed, and 60% of those failures were caused by obstacles in the recharging location. This means the insertion of the plug into the outlet succeeded 98% of the time. Taking into account retries, the plugging in task never caused the robot to run out of battery power during the 13 days run. The code for outlet detection and pose estimation used for Continuous Operation project is available in ROS package outlet_pose_estimation [33]

VII. CONCLUSION

We have shown that a video camera can be used to detect outlets in close range with few false alarms. The detection algorithm works for outlets of different structure and color and is robust against shadows, oversaturated and dark images. We have also presented an algorithm for finding the outlet holes pose with high accuracy by using the wall normal obtained from stereo data. An experiment with PR2 showed that the system is robust enough to be used for robot continuous operation. .

VIII. ACKNOWLEDGEMENTS

The authors are grateful to Gary Bradski, Kurt Konolige and Patrick Mihelich for many useful discussions.

REFERENCES

[1] iRobot, "Roomba cleaning robot with self-charging home base station." [Online]. Available: http://www.irobot.com

[2] Sony, "Aibo entertainment robot with self-charging function." [Online]. Available: http://support.sony-europe.com/aibo/index.asp

[3] A. Bagepalli, M. Zamora, and D. Sanchez, "Charger the wall charging robot [online]," 2005. [Online]. Available: http://www.engr.panam.edu/ee/labs/senior/website/index.htm

[4] B. Mayton, L. LeGrand, and J. Smith, "Robot, feed thyself: Plugging in to unmodified electrical outlets by sensing emitted ac electric fields," in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, May 2010, pp. 715 –722.

[5] W. Meeussen, M. Wise, S. Glaser, S. Chitta, C. McGann, P. Mihelich, E. Marder-Eppstein, M. C. Muja, V. Eruhimov, T. Foote, J. Hsu, R. B. Rusu, B. Marthi, G. R. Bradski, K. Konolige, B. P. Gerkey, and E. Berger, "Autonomous door opening and plugging in with a personal robot," in *ICRA*, 2010, pp. 729–736.

[6] S. Hinterstoisser, O. Kutter, N. Navab, P. Fua, and V. Lepetit, "Real-time learning of accurate patch rectification," *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, vol. 0, pp. 2945–2952, 2009. [Online]. Available: http://dx.doi.org/10.1109/CVPRW.2009.5206794

[7] H. J. Wolfson and I. Rigoutsos, "Geometric hashing: An overview," *Computing in Science and Engineering*, vol. 4, pp. 10–21, 1997.

[8] E. R. Torres-jara and R. Brooks, "A self-feeding robot," Jan 2002. [Online]. Available: http://people.csail.mit.edu/etorresj/MST/main.pdf

[9] P.Viola and M.Jones, "Robust real-time object detection," in *Int'l. J. Computer Vision*, 2004.

[10] T. Yamada, K. Nagatani, and Y. Tanaka, "Autonomous insertion of a plug into real electrical outlet by a mobile manipulator," in *Proc. of the Intl. Conf. on Field and Service Robotics (FSR)*, P. Corke and S. Sukkarieh, Eds., vol. 25. Springer, 2006, pp. 389–400.

[11] L. Bustamante and J. Gu, "Localization of electrical outlet for a mobile robot using visual servoing," in *Electrical and Computer Engineering, 2007. CCECE 2007. Canadian Conference on*, april 2007, pp. 1211 –1214.

[12] G. Bradski and A. Kaehler, *Learning OpenCV: Computer Vision with the OpenCV Library*. Cambridge, MA: O'Reilly, 2008.

[13] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[14] H. Bay, T. Tuytelaars, and L. V. Gool, "SURF: Speeded Up Robust Features," in *9th European Conference on Computer Vision*, Graz Austria, May 2006.

[15] J. Matas, O. Chum, U. Martin, and T. Pajdla, "Robust wide baseline stereo from maximally stable extremal regions," in *Proceedings of British Machine Vision Conference*, vol. 1, London, 2002, pp. 384–393.

[16] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," in *European Conference on Computer Vision*, vol. 1, May 2006, pp. 430–443. [Online]. Available: http://mi.eng.cam.ac.uk/ er258/work/rosten_2006_machine.pdf

[17] ——, "Fusing points and lines for high performance tracking." in *IEEE International Conference on Computer Vision*, vol. 2, October 2005, pp. 1508–1511. [Online]. Available: http://mi.eng.cam.ac.uk/ er258/work/rosten_2005_tracking.pdf

[18] M. Ozuysal, M. Calonder, V. Lepetit, and P. Fua, "Fast keypoint recognition using random ferns," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, pp. 448–461, 2010.

[19] I. Gordon and D. G. Lowe, *What and Where: 3D Object Recognition with Accurate Pose*. Springer, 2006, vol. 4170, pp. 67–82.

[20] A. C. Romea, D. Berenson, S. Srinivasa, and D. Ferguson, "Object recognition and full pose registration from a single image for robotic manipulation," in *IEEE International Conference on Robotics and Automation (ICRA '09)*, May 2009.

[21] M. M. Torres, A. C. Romea, and S. Srinivasa, "Moped: A scalable and low latency object recognition and pose estimation system," in *Proceedings of ICRA 2010*, May 2010.

[22] *Histograms of Oriented Gradients for Human Detection*, vol. 1, 2005. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1467360

[23] S. Hinterstoisser, V. Lepetit, S. Ilic, P. Fua, and N. Navab, "Dominant orientation templates for real-time detection of texture-less objects," *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, vol. 0, pp. 2257–2264, 2010.

[24] V. Lepetit, F. Moreno-Noguer, and P. Fua, "EPnP: An Accurate $O(n)$ Solution to the PnP Problem," *International Journal of Computer Vision*, vol. 81, no. 2, pp. 155–166, 2009.

[25] R. Haralick, D. Lee, K. Ottenburg, and M. Nolle, "Analysis and solutions of the three point perspective pose estimation problem," in *Computer Vision and Pattern Recognition, 1991. Proceedings CVPR '91., IEEE Computer Society Conference on*, June 1991, pp. 592 –598.

[26] M. Dhome, M. Richetin, J.-T. Lapreste, and G. Rives, "Determination of the attitude of 3d objects from a single perspective view," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 11, no. 12, pp. 1265 –1278, Dec. 1989.

[27] R. Horaud, B. Conio, O. Leboulleux, and B. Lacolle, "An analytic solution for the perspective 4-point problem," in *Computer Vision and Pattern Recognition, 1989. Proceedings CVPR '89., IEEE Computer Society Conference on*, June 1989, pp. 500 –507.

[28] L. Quan and Z. Lan, "Linear n-point camera pose determination," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 21, no. 8, pp. 774 –780, Aug. 1999.

[29] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge University Press, ISBN: 0521540518, 2004.

[30] G. Schweighofer and A. Pinz, "Robust pose estimation from a planar target," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, pp. 2024–2030, 2006.

[31] "Outlet test dataset." [Online]. Available: http://itseez.com/data/outlet_detection/outlet_dataset.tar.gz

[32] [Online]. Available: https://code.ros.org/svn/wg-ros-pkg/branches/trunk_cturtle/sandbox/outlet_test

[33] [Online]. Available: http://www.ros.org/wiki/outlet_pose_estimation