Active Vision for Door Localization and Door Opening using Playbot: A Computer Controlled Wheelchair for People with Mobility Impairments

Alexander Andreopoulos and John K. Tsotsos
Department of Computer Science & Engineering and Centre for Vision Research
York University
Toronto, Ontario, M3J 1P3, Canada
{alekos, tsotsos}@cse.yorku.ca

Abstract

Playbot [1, 13] is a long-term, large-scale research project, whose goal is to provide a vision-based computer controlled wheelchair that enables children and adults with mobility impairments to become more independent. Within this context, we show how Playbot can actively search an indoor environment to localize a door, approach the door, use a mounted robotic arm to open the door, and go through the door, using exclusively vision-based sensors and without using a map of the environment. We demonstrate the effectiveness of active vision for localizing objects that are too large to fall within a single camera's field of view and show that well-calibrated vision-based sensors are sufficient to safely pass through a door frame that is narrow enough to tolerate a wheelchair localization error of at most a few centimetres. We provide experimental results demonstrating near perfect performance in an indoor environment.

1. Introduction

Current assistive technology for the physically disabled rely on the user's visual system as part of a closed-loop control system. For example, in one class of robotic aids, specialized sensors are developed for a finger or eyebrow. To grasp an object, the user visually guides a robot manipulator through a series of micro-activations to the target. Each micro-activation moves a particular joint of a robot arm by a small distance. This can be tedious, especially for children with mobility impairments, as the user tires easily and the amount of work done is insufficient. Playbot (Fig. 1) is designed to replace part of this control loop [13]. The user's visual system is still needed to determine the goal of a manipulation and to communicate with the robot. But the robot's visual system, then, takes its place in the closed loop control of the robot in the execution of the task. The user is, thus, spared the frustration, tedium and effort.

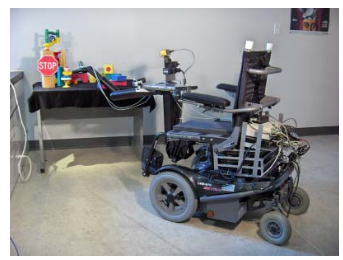


Figure 1. The Playbot wheelchair.

In this paper, we address the problem of how a robotic wheelchair equipped only with vision sensors and a robotic manipulator, can localize, approach, open and pass through a doorway. This is composed of several sub-tasks: (a) From any position and at any spatial orientation in a room or hallway, with or without a map, locate a door using a set of door appearance and structure models. (b) Plan a reasonable, forward motion, route to the door and approach the door. (c) Orient the chair to face the doorway. (d) Determine if the door opens inwards or outwards. (e) Locate the door handle and determine if it is openable given the wheelchair's manipulator. (f) Complete the door approach and open it. (g) Move forward or backward to allow the door to be fully open. (h) Plan a route through the doorway and follow it.

A number of researchers have dealt with various of the above sub-problems. Brooks *et al.* [5] present a robot equipped with a dexterous arm that is capable of finding a door by matching its color histogram, pushing the door open and going through it. However, the robot does not deal with the problem of reaching and turning the door handle and the authors assume that the robot is going down a hallway – i.e., it is in close proximity to the door. Furthermore, it does not deal with the problem of modeling the actual shape of a door or handle. Rhee *et al.* [10] present a control strategy for door opening with the use of a mobile 3-fingered manipulator. Active sensing algorithms are proposed to overcome uncertainties in a real environment. Rather than using a wrist force/torque sensor for force and position control, the contact force data of a multifingered robot hand is used. Niemeyer and Slotine [9] use the path of least resistance to control an arm and open a door. Nagatani and Yuta [8] model the door opening task using a sequence of planned motion primitives – called action primitives by the authors – where each action primitive is designed with an error adjustment mechanism to help deal with positioning errors. Each of the above papers deals with only a sub-problem of the door localization and opening problem. We present an active vision based approach for localizing, opening and going through one side of a doorway assuming an unobstructed path to the door.

Section 2 of the paper describes the hardware of Playbot and the different coordinate systems used in our implementation. Section 3 presents a multistage and multiscale active vision based framework that searches for a door and accurately approaches it. Section 4 outlines our methodology for opening the door using a simple 6+2 d.o.f. manipulator and going through the door. Section 5 provides experimental results and Section 6 concludes the paper.

2. Playbot Hardware and Coordinate Systems

Playbot consists of a modified electric wheelchair (the Chair-man Entra, by Permobil Inc., Sweden), a 6+2 d.o.f. robotic manipulator (MANUS, by Exact Dynamics, Netherlands), a tablet PC, a number of monocular and binocular cameras, control electronics, three on-board laptops running Linux, an off-board server (Sun Fire X2100 with a dual core AMD Opteron 1.75, 4GB RAM) also running Linux and providing further computational power and potentially, further offboard computers as needs arise. The only sensor used in this paper is a Bumblebee stereo camera mounted on a high precision pan-tilt unit (Directed Perception Inc. model PTU-D46). A total of 12 coordinate systems are defined, as shown in Fig. 2. A value of θ in a node $\mathbf{v}$ indicates that node $\mathbf{v}$ and all its descendants can be rotated about node $\mathbf{v}$'s axis. For example, by panning the camera, all the children of PAN_CENTER are also rotated and by rotating the wheelchair (WHEELCHAIR_CENTER) all the other coordinates in the graph are also affected. Notice that RECTIFIED_POINTS and WORLD_PARALLEL do not affect each other as the camera's intrinsic parameters are constant. The WHEELCHAIR_CENTER axis can also be translated (τ) since the wheelchair can move forward/backward.

1. WHEELCHAIR_CENTER: Defines a vertical axis about which the wheelchair rotates.
2. PAN_CENTER: Defines an axis about which the camera pans.

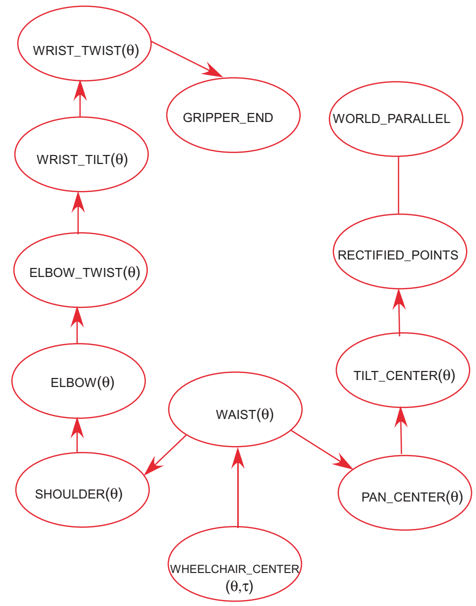


Figure 2. The coordinate systems.

3. TILT_CENTER: Defines an axis about which the camera tilts.
4. RECTIFIED_POINTS: Defines the rectified coordinate system of the Bumblebee's left camera.
5. WORLD_PARALLEL: Defines a coordinate system that is the result of a pure rotation of the RECTIFIED_POINTS coordinate system such that one of the resulting coordinate axes (the z-axis in our implementation) point along the direction of forward motion of the wheelchair and another one of the coordinate axes (the y-axis in our implementation) is normal to the floor plane when $\theta = 0$ for PAN_CENTER, TILT_CENTER and WAIST.
6. WAIST: Defines a coordinate axis about which the arm's waist rotates. Note that, as indicated in Fig. 2, due to the placement of the pan-tilt unit and the Bumblebee camera, a rotation of the waist causes a rotation of the pan-tilt unit and the Bumblebee camera about the waist's axis.
7. SHOULDER: Defines a coordinate axis about which the shoulder rotates.
8. ELBOW: Defines a coordinate axis around which elbow rotations occur.
9. ELBOW_TWIST: Defines a coordinate axis, approximately orthogonal to the ELBOW axis, around which the elbow twists.

10. WRIST_TILT: Defines a coordinate axis, approximately orthogonal to the ELBOW_TWIST axis, around which the arm's wrist tilts.

11. WRIST_TWIST: Defines a coordinate axis, approximately orthogonal to the WRIST_TILT axis, around which the wrist twists.

12. GRIPPER_END: Defines a fixed coordinate system located approximately at the end of the gripper. It is now possible to determine the current coordinates of the gripper's endpoint with respect to any of the above mentioned coordinate systems.

Note that the manipulator has two extra degrees of freedom which we do not use in our current implementation (open-the-grasper and lift-the-entire-manipulator).

3. The Sub-Tasks

3.1. Active Door Frame Localization

The concept of active perception (a.k.a. active vision) was introduced by Bajcsy [4] as "a problem of intelligent control strategies applied to the data acquisition process". Active vision is often described as a subset of attention [12]. A number of problems make the door frame localization problem challenging and point to the need for active vision in this context:

1. Standard stereo correspondences are easier to find in highly textured regions. However, especially in indoor environments, this assumption cannot always be made. We need a depth extraction methodology that is robust and extremely accurate even in cases where typical stereo depth extraction techniques fail. Especially doors, which typically have few salient features, make this task non-trivial.

2. We need to detect a door frame even under partial occlusions and in situtations where only part of the door falls in the camera's field of view. Since we make very few assumptions about the position and orientation of the wheelchair, as well as the camera's intrinsic and extrinsic parameters, it is impossible to build a robust but passive system that accomplishes this task.

3. We need extremely reliable door detection rates with few false positives and false negatives. The algorithm must be robust against small changes in illumination, small calibration errors and image noise. While it is difficult to quantify the exact recognition rates needed to ensure user satisfaction, historically, successful vision systems in industry have achieved over 99% recognition rates [7]. Of course the necessary precision and recall rates can vary drastically depending on the application, but as a rule of thumb, we need to follow a framework that could ultimately provide such excellent recognition rates.

4. From arbitrary initial positions, we need to be able to navigate the wheelchair to within a few centimetres of its ideal target position and with a very precise orientation. In our case a deviation of at most 6-7cm is tolerated for the wheelchair to safely pass through the door. This is difficult with stereo cameras as the accuracy of stereo based depth extraction degrades when the target is too far away (large camera baselines are needed in such cases) or too close to the camera (large disparities or potentially the target not falling in the field of view of both the left and right cameras can cause problems).

We demonstrate how an active vision system can deal with many of these problems in a real world setting.

3.2. Door Model

We use corner based features and edges to detect a door frame by assuming that a door frame can be described by a set of corners connected by lines. Corners are detected using the Good Features to Track algorithm [11] and edges are detected using the Canny edge detector [6]. A door frame is modeled as consisting of two concentric rectangles (Fig.3(a)) parallel to the floor's normal. The outer/inner rectangle is assigned a width of 100.1cm/88.5cm and a height of 219.3cm/213.5cm (Fig.3(a)). Any door consisting of rectangles with salient corners and edges can be similarly modeled. Notice that if the door frame has been successfully localized in an image, and since we have a model of the door's dimensions, it is possible to determine its relative position and orientation with respect to the coordinate system of the monocular camera that acquired the image. We use the left camera of the Bumblebee stereo camera to detect the door frame.

3.3. Features and Search Space

Given a series of N images I_1,...,I_N, acquired by panning and tilting the camera by (p_1, t_1),...,(p_N, t_N) respectively (we discuss in another section how the specific pan and tilt movements are chosen) and keeping all the other coordinate systems constant, we apply a function `extract_corners`(I_i,p_i,t_i) to each image I_i. This function extracts all corners from an image and gives us a line in space on which the corner must lie. The extracted line is described in the RECTIFIED_POINTS coordinate system with a pan and tilt value of zero. In other words, all extracted lines are described in a common reference frame with pan/tilt values of zero. We refer to these lines as the detected *beams*. These beams provide a sort of 'focus-of-attention' as we now know that if the door has M corners $\{c_1, ..., c_M\}$, the localized 3D door model would ideally be coincident to M of those beams. This drastically reduces the search space. If a corner is imaged in more than two images, the 3D door model might be coincident to more than M beams. If the corner detector fails to detect one of the door frame corners, the 3D door model might be coincident to less than M beams. A multiscale, hypothesize-and-test approach is used to localize the door. Each one of the beams is discretized along its arc-length by uniformly sampling the beam every `grainSizeDist*`2^s cm, where $s \geq 0$ denotes the current scale which will be discussed in more detail in the following subsection. We select a subset $C \subseteq \{c_1, ..., c_M\}$ of the door's corners and for each corner $c \in C$ we hypothesize that corner c of the door model might be coincident at each location on the discretized beams. For each such location, the door frame model is rotated about a vertical y-axis passing through c. The rotation angles fall within $[degreeMean - degreeVar$, $degreeMean + degreeVar]$ degrees and account for potential wheelchair rotations with respect to the door. The angle interval is discretized by uniformly sampling every `grainSizeAngle` degrees. This results in a hypothesis set H^s of potential locations for the door at the current scale. The mean (x, y, z) coordinate of a door model hypothesis is referred to as the *door center*. By navigating the wheelchair such that the axis of WHEELCHAIR_CENTER (the wheelchair's axis of rotation) intersects the point in space corresponding to the door center, we can safely pass through the door – assuming of course that the door has been opened first. To keep H^s's cardinality finite, we only retain the hypotheses whose door frame center falls within a $[Xmean - Xvar, Xmean + Xvar] \times [Ymean - Yvar, Ymean + Yvar] \times [Zmean - Zvar, Zmean + Zvar]$ cm^3 region located in front of the wheelchair in the WHEELCHAIR_CENTER coordinate system. Notice that by limiting our search on the beams, we do not have to search the entire $[Xmean - Xvar, Xmean + Xvar] \times [Ymean - Yvar, Ymean + Yvar] \times [Zmean - Zvar, Zmean + Zvar]$ cm^3 volume, making our approach tractable. We also apply a function `extract_lines(`I_i`)` to each image I_i, that returns a binary image L_i of the lines extracted from I_i with the Canny edge detector. Next, we describe a multiscale search that is applied to H^s, to determine the door location.

3.4. Multiscale Door Search

As indicated above, a multiscale, hypothesize-and-test approach is used to localize the door. As we will demonstrate in our experiments' section, a multiscale approach combined with a search limited on the detected beams, makes a hypothesize-and-test approach computationally tractable – assuming typical width and height dimensions of a few meters for the indoor environment where the search is conducted. Each image I_i is transformed into a binary image B_i^s, where s denotes scale. Row y, column x of B_i^0 has an assigned value of 1 if and only if there is a detected corner in a `(2*radius+1)`$\times$`(2*radius+1)` square centered at row y, column x of I_i. Typical values for `radius` might be around 2-3 pixels, depending on the task and the hardware used, and are meant to compensate for small calibration errors. For $s > 0$, row y, column x of B_i^s has an assigned value of 1 if and only if there is a corner detected inside a `(2*max(radius,`2^s`)+1)`$\times$`(2*max(radius,`2^s`)+1)` pixel square centered at row y, column x of I_i.

First, we apply a rapid and multiscale rejection of incorrect hypotheses: Let $M' \leq M$. Assuming `s_max` is the total number of scales used, and $s \leftarrow$`s_max-1`, we remove from H^s the set 'H^s of hypotheses for which less than M' of their corners have coordinates with a labelling of 1 in at least one of B_1^s,...,B_N^s – we reproject each of the door model's corners on the original image coordinates to determine this. Then, we let $H^{s-1} \leftarrow$ "H^s – where "H^s is defined below –, we assign $s \leftarrow s - 1$ and repeat the above procedure until we obtain set H^0. "H^s is obtained by sampling twice for each $h \in (H^s -$ '$H^s)$, by shifting h by $\pm\frac{1}{3}$`grainSizeDist*`2^s cm along the arclength of the beam used to obtain the hypothesis h. In other words H^{s-1} is a resampled version of H^s on a finer scale and $|H^{s-1}| = 2 * |H^s -$ '$H^s|$.

Once we have obtained H^0, we assign a probability $p(h)$ to each of the hypotheses in $h \in H^0$, indicating the likelihood that h localizes the door correctly. The most likely hypothesis is chosen as the detected door. Firstly, we assign a probability of zero to each hypothesis in 'H^0. Then, for each hypothesis $h \in (H^0 -$ '$H^0)$, we project the corresponding door model's lines on each of the images L_1,...,L_N. Assume the door model hypothesis h consists of O lines $\{l_1^h, ..., l_O^h\}$. For each line l_i^h's projection on image L_j, we discretize its projection in the image uniformly with a 2-pixel spacing. Let P_{ij}^h denote the set of discretized pixels of line l_i^h falling within image L_j's field of view. For each discretized pixel $p \in P_{ij}^h$, we assign a score $val_{ij}^h(p)$ which equals 0 if and only if a line was detected in L_j somewhere along a 3 pixel-long line perpendicular to l_i^h's projection in the image and centered at p. Otherwise $val_{ij}^h(p)$ is set to 1. The total score V_i^h for line l_i^h across all images is defined as

$$V_i^h = q_i * \left(\frac{\sum_{j=1}^{N} \sum_{p \in P_{ij}^h} val_{ij}^h(p)}{\sum_{j=1}^{N} |P_{ij}^h|}\right) + (1 - q_i) \qquad (1)$$

where $0 \leq q_i \leq 1$ denotes the proportion of line l_i^h that falls in at least one image's field of view. The final probability $p(h)$ for hypothesis h is given by

$$p(h) = exp(-\frac{\sum_{i=1}^{O} w_i V_i^h}{\sum_{i=1}^{O} w_i}) \quad (2)$$

where $w_i \geq 0$ are weights of importance for each of the lines' scores. In our implementation all lines used are assigned an equal weight.

3.5. Approaching the Door

The further away from the door Playbot is, the greater the error in the door localization, implying that multiple door localizations are needed to minimize the error. Playbot approaches the door in N movements/phases, at each phase using the previous phase's hypothesis to guide the camera search region to the expected door position and effectively using visual servoing for hypothesis verification. The greater the initial distance of Playbot from the localized door, the greater the value of N. We define the *visibility horizon* as the minimum radius circle centered at the door center's projection on the floor, such that from any point on its circumference, the entire door is visible by the camera with pan-tilt movements. In the first $N-3$ phases, Playbot approaches the door along the minimum distance path (a straight line) until it is on the visibility horizon. In our implementation the visibility horizon's radius is slightly overestimated at 2 meters to account for potentially small odometry errors. In each of the first $N-3$ phases, a predefined number of pan and tilt movements are performed, exploring a region in front of the wheelchair to localize the door. Upon localizing the door, the wheelchair rotates, if necessary, so that a subsequent forward motion of the wheelchair would cause the WHEELCHAIR_CENTER axis to intersect the door center and uses a simple odometer it is equipped with, to cover $\frac{1}{N-n-2}$ of the newly determined distance to the visibility horizon – where n denotes the current phase number. The number of phases chosen offers a compromise between being sufficiently large to ensure that, due to errors, the door does not fall outside the search region in front of Playbot between phases and being sufficiently small to minimize the number of door searches.

In phase $N-2$, another door localization takes place. We know the approximate door position, so three camera tilts with a single constant pan value for each tilt are sufficient to explore the entire door and estimate its distance and orientation. Upon localization, the wheelchair rotates so that a forward motion followed by a second rotation, places it in front and perpendicular to the door and approximately on the visibility horizon. The door center now has an approximate x-coordinate of zero in WHEELCHAIR_CENTER coordinates. The final door localization takes place in phase $N-1$. Upon localization of the door, a small rotation is made to guarantee that the door center has an x-coordinate of zero and Playbot moves forward until it is as close as possible to grasping distance of the door handle and the handle is visible by both views of the stereo camera – we assume the door handle's distance to Playbot equals the door's distance.

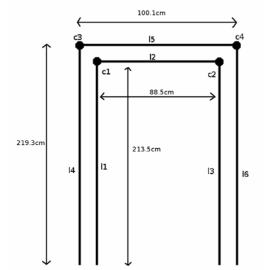
4. Final Phase

4.1. Handle Localization

We now describe the final phase N. To localize the door handle, turn the door handle and open the door, we use the algorithm given by Andreopoulos and Tsotsos in [2]. The handle localization must be extremely reliable so as not to frustrate the user and must be capable of dealing with specularities and modest changes in illumination. Recognition methods based on edges/lines are more robust than appearance based techniques under illumination and contrast changes. To achieve some robustness under changes in contrast and illumination, we use an algorithm introduced by Viola and Jones [14], which is based on a set of Haar-like features. As training data for each door handle, the training algorithm uses a dataset of 1,500 instances of the door handle at slightly different rotations and with different artificially induced illumination and contrast conditions. All the 1,500 instances of the door handle were artificially generated using a single template image of the door handle. One or more candidate handles are typically detected and we, therefore, need a method to remove any potential false positives. A simple template matching approach in HSV space is used. The template of the door handle and each detected region is converted to HSV space and a histogram comparison in the S channel spaces provides us with the candidate door handle. The candidate region most closely matching the door handle's template is selected as the segmented door handle region. To compare the S channels of two image regions, we first normalize each region's S channel histogram to have a total area of 1. The final matching score for the two image regions is the total area of intersection of the two histograms. We ignore the H channel histogram as the door handle images contain white/grey/black, potentially making the H channel histogram ill-conditioned. While the metallic door handle has a low saturation, the area surrounding the handle has a sufficiently different histogram profile to make discrimination based on the S channel reliable.

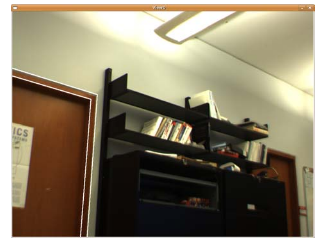
4.2. Door Opening

We extract reliable depth values from any regions of the door handle where conspicuous features exist. In our case this region corresponds to the keyslot location, as it is seen in Fig. 3(d). We use the relative coordinate system implied



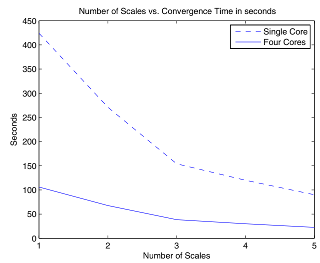
(a)

(b)

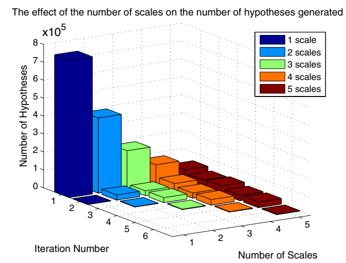


(c)

(d)



(e)



(f)

Figure 3. (a): Diagram of the door model's dimensions with the 4 corners used ($c_1, ..., c_4$) and the 6 lines used ($l_1, ..., l_6$). (b),(c): Examples of the localized door frame, with white lines. Notice that in (c) the door does not fall completely in the camera's field of view. (d): The door handle's medial axis and the detected keyslot used for depth extraction. (e): Effect of increased scale and parallelization on door frame localization speed. (f): Number of hypotheses generated for each scale's iterations.

by the segmented door handle region and the known geometry of the door handle to accomplish this. A Canny edge detector is used in conjunction with a probabilistic Hough transform in order to detect the upper and lower parts of the door handle falling in the segmented door handle region. The parts are detected by searching in the segmented door handle region for the two most horizontally parallel lines that have a certain minimum amount of separation. As the lines detected by the Hough transform do not always detect the entire length of the upper and lower region of the door handle we extend the detected lines to cover the entire segmented region. From those two lines we extract their medial line which also intersects the lock region, as shown in Fig. 3(d). We use this medial line to detect the keyslot location by searching for the darkest region close to the right-half of the line. Due to the way the wheelchair approached the door we know that the stereo camera is almost parallel to the door. We obtain an accurate depth estimate for the keyslot using a standard stereo based depth extraction algorithm. A small correction of the wheelchair's orientation is performed at this stage by rotating around the WHEELCHAIR_CENTER axis. Playbot rotates so that the door handle's keyslot x-coordinate is equal to that needed to safely pass through the door, based on our prior knowledge of the door handle's position on the door. At this point Playbot approaches the door so that the door handle's z-axis distance with respect to the WORLD_PARALLEL coordinate system is about $55cm$. At this distance the 6+2 d.o.f. arm can reach and open the door. A single contact point on the door handle is defined and the arm is used to push that contact point along a desired circular trajectory, turning the door handle. A small forward push on the door followed by a counterclockwise rotation of the extended arm around the WAIST axis, pushes the door wide open. A simple forward motion of the wheelchair is now sufficient for the wheelchair to pass through the door. This single contact point simplifies the task and makes the solution more reliable as no grasping is needed and we only need to solve the inverse kinematics problem in order to push the contact point along the desired circular trajectory without having to worry about force/torque control of the arm. However, force/torque control with compliance would likely simplify the problem and lessen the need for very accurate vision modules. As our robot arm is not equipped with any force sensors, all arm control is vision based.

5. Experiments

Our implementation consists of approximately 10,000 lines of C++ code and uses the OpenCV and GSL libraries. All our code is executed on an Intel Quad Core CPU with 3GB RAM. The multiscale hypothesis generation/evaluation stage was parallelized through the use of the Intel Threading Building Blocks library, achieving an almost four-fold speed improvement compared to using a single core for hypothesis generation/evaluation. We used four phases ($N = 4$) in our experiments. We performed twenty-five test runs, investigating the algorithm's performance when Playbot was placed at a variety of initial positions and orientations. In all our test cases, Playbot's WHEELCHAIR_CENTER axis was placed at a phase I initial position of approximately $250 - 350$cm perpendicular to the door center (Fig. 4), ± 200cm parallel to the door center and with an orientation that would ensure that the door fell within the field of view of at least one of a series of phase I pan-tilt movements (within a ± 30 degree rotation around the WHEELCHAIR_CENTER axis). We set $M' = M = 4$, $O = 6$ and search for the four corners and six lines indicated in Fig. 3(a). During phase I, the camera performs five (pan,tilt) movements of $(-20, 20), (-10, 20), (0, 20), (10, 20), (20, 20)$ degrees, covering approximately the region in front of the wheelchair. In Fig.3(b),(c) we show the localized door from two phase I images. Notice that as we keep the tilt values constant, the entire door frame is not imaged by the Bumblebee camera. During each of phase II and phase III, three pan-tilt movements were performed ($(p, 26)$, $(p, 10)$, $(p, -6)$) where p is a pan movement that would ensure the door falls in the camera's field of view. The necessary value of p is calculated using the previous phase's door localization information. We also investigated how the multiscale search affects the door localization speed, by varying the number of scales on a search problem (Fig. 3(e),(f)). During phase I we set $degreeMean = 0°$, $Xmean = 0$, $Ymean = c$, $Zmean = 300cm$, $degreeVar = 45°$, $Xvar = 200cm$, $Yvar = 3cm$ and $Zvar = 50cm$ where c is a prior estimate of the door center's y-coordinate, determined by a calibration phase. Phase I door localization is done in approximately 20 seconds, depending on the number of detected features. During phases II and III we set $degreeVar = 20°$, $Xvar = 50cm$, $Yvar = 5cm$, $Zvar = 50cm$ to model the lower uncertainty in the door position. $Xmean$, $Ymean$, $Zmean$ and $degreeMean$ are set using the previous phase's door localization estimate, taking into consideration of course the movement made by Playbot since the last phase. Due to this smaller search space, the door localization takes about 5 seconds. In a typical phase I/II/III run, Playbot detects about 8,000/4,000/4,000$\pm$1,000 corners respectively.

As indicated above, we performed twenty-five test runs for a total of seventy-five door localizations across all three phases. We observed a 100% success rate in localizing the door frame. The door handle localization algorithm

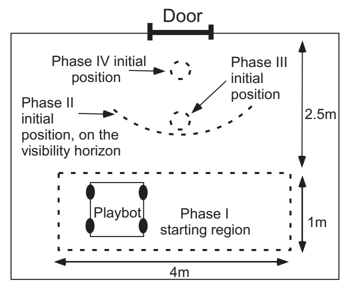


Figure 4. Positioning of Playbot.

failed to localize the keyslot once out of the twenty-five test runs. When this failure occured, we re-ran the door handle localization part of our algorithm without moving the wheelchair. The keyslot was localized successfully and the test run was also completed successfully. We also observed twice that the door handle was not pushed sufficiently downwards to open the door. A manual intervention slightly pushed the door handle downwards to open the door in these two cases. We do not consider this a serious problem, as it could easily be solved if we had at our disposal an arm with force-feedback. When no more force could be applied to turn the door handle further, we would know that we could open the door. In all twenty-five cases Playbot safely passed through the open door, demonstrating the validity of our approach. Calibration errors can make the door localization more challenging. For example, we have observed that the greater the error in the WORLD_PARALLEL coordinates, the more distorted the door model's projection on the image is, making localization less reliable. The problem becomes noticeable even with a few degrees of added error.

Notice that the door model's projection on an image is identical to the projection of a rescaled version of the same door model. One might argue that at least one more camera is needed to extract accurate depth. This could involve applying the same algorithm to the features extracted with the other cameras and verifying that they all lead to the same hypothesis. However, this would obviously slow down the door localization algorithm. Our visual servoing implemented through the three phases, minimizes the likelihood of this occurence, and demonstrates another benefit of the active approach to vision: Given a door localization $\mathbf{p}$ at phase I, we know approximately the door location $\mathbf{p}'$, $\mathbf{p}''$ relative to the camera once the wheelchair has moved to its new position in phases II, III respectively. If we are dealing with a scaled version of the correct door model – a false positive – the likelihood of erroneously re-localizing the false positive door model during phases II, III is diminished, since the door search region becomes centered around the door's expected position ($\mathbf{p}'$ or $\mathbf{p}''$), the door search region size decreases and at the same time the scaled door model's projection on an image will almost certainly cause the door model localization to differ from its expected position and fall outside its search region. Such false positive localizations did not appear during our testing and did not, in any way, affect our results. In our implementation, the small value used for $Yvar$ poses extremely stringent constraints on the set of scaled door models that could result in such false positives and makes their likelihood of occurence negligible.

The importance of the multiscale approach to the door localization should be emphasized. In Fig. 3(e),(f) we provide some results demonstrating the excellent speed improvements offered by our multiscale approach. As we increase the number of scales used, we observe a significant improvement in the door localization speed. A 5-scale door localization provided an over 500% speed improvement compared to a single scale localization, making it possible to run the algorithm within a reasonable time frame on the hardware we have at our disposal. We tested the algorithm on a total of 5 scales. In Fig. 3(f) we demonstrate that the multiscale approach to door localization also offers a non-trivial reduction in memory requirements, as the number of generated hypotheses decreases drastically with increasing scale. To counter the increasing relative scale of the door's projection on the camera as we approach the door, we set `grainSizeDist` to 1cm/0.5cm/0.25cm and set `grainSizeAngle` to $1°$/$0.5°$/$0.5°$ during phases I/II/III respectively. Too high of a value for `grainSizeDist`, `grainSizeAngle` causes failures in the door localization, while too low of a value causes increased processing time. A video demonstration of this work is found at: `www.cse.yorku.ca/~playbot/door.wmv`. Playbot is tethered with cables in the video, as all the computation is done on an off-board computer that uses the cables for communicating with the wheelchair.

6. Conclusion

While sonars and lasers are superior to vision-based sensors for depth estimation, we have demonstrated that in an indoor environment with controlled illumination conditions, vision sensors, combined with an active and purposive approach to vision, are in many ways superior, as they can provide sufficiently accurate depth estimates while simultaneously achieving accurate object localization. It is our belief that as the need for more advanced recognition in robotic systems emerges, the active approach to vision will become indispensable. Calibration and odometry errors make a one shot door localization and opening approach infeasible. We have demonstrated that an active step-by-step approach allows us to better deal with such errors and solve a non-trivial task. Future work involves testing with multiple door models and lighting conditions, opening and closing the door from both sides using different handles and employing multi-camera systems [3].

Acknowledgements

JKT holds the Canada Research Chair in Computational Vision and gratefully acknowledges its financial support. AA also gratefully acknowledges the financial support of the NSERC PGS-D scholarship program.

References

[1] The Playbot Project `www.cse.yorku.ca/~playbot/`.

[2] A. Andreopoulos and J. K. Tsotsos. A framework for door localization and door opening using a robotic wheelchair for people living with mobility impairments. In *Robotics: Science and Systems - Robot Manipulation: Sensing and Adapting to the Real World*, 2007.

[3] A. Andreopoulos and J. K. Tsotsos. Information fusion for multi-camera and multi-body structure and motion. In *8th Asian Conference on Computer Vision (ACCV)*, 2007.

[4] R. Bajcsy. Active perception vs. passive perception. In *IEEE Workshop on Computer Vision Representation and Control*, Bellaire, Michigan, 1985.

[5] R. Brooks, L. Aryananda, A. Edsinger, P. Fitzpatrick, C. Kemp, U. O'Reilly, E. Torres-jara, P. Varshavskaya, and J. Weber. Sensing and manipulating built-for-human environments. *International Journal of Humanoid Robotics*, 1(1):1–28, 2004.

[6] J. Canny. A computational approach to edge detection. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 8:679–714, 1986.

[7] M. Ejiri. Machine vision in early days: Japan's pioneering contributions. In *8th Asian Conference on Computer Vision (ACCV)*, 2007.

[8] K. Nagatani and S. Yuta. Designing strategy and implementation of mobile manipulator control system for opening door. In *IEEE International Conference on Robotics and Automation*, April 1996.

[9] G. Niemeyer and J. Slotine. A simple strategy for opening an unknown door. In *IEEE International Conference on Robotics and Automation*, 1997.

[10] C. Rhee, W. Chung, M. Kim, Y. Shim, and H. Lee. Door opening control using the multi-fingered robotic hand for the indoor service robot. In *IEEE International Conference on Robotics and Automation*, April 2004.

[11] J. Shi and C. Tomasi. Good features to track. In *Computer Vision and Pattern Recognition*, 1994.

[12] J. Tsotsos. On the relative complexity of active vs. passive visual search. *International Journal of Computer Vision*, 7(2):127–141, 1992.

[13] J. Tsotsos, G. Verghese, S. Dickinson, M. Jenkin, A. Jepson, E. Milios, F. Nuflo, S. Stevenson, M. Black, D. Metaxas, S. Culhane, Y. Ye, and R. Mann. PLAYBOT: A visually-guided robot to assist physically disabled children in play. *Image and Vision Computing: Special Issue on Vision for the Disabled*, pages 275–292, April 1998.

[14] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *CVPR*, 2001.