"Bring it to me" - Generation of Behavior-Relevant Scene Elements for Interactive Robot Scenarios

Nils Einecke, Manuel Mühlig, Jens Schmüdderich, Michael Gienger

2011

Preprint:

This is an accepted article published in Proceedings of the IEEE International Conference on Robotics and Automation. The final authenticated version is available online at: https://doi.org/[DOI not available]

2011 IEEE International Conference on Robotics and Automation
Shanghai International Conference Center
May 9-13, 2011, Shanghai, China

"Bring it to me" - Generation of Behavior-Relevant Scene Elements for Interactive Robot Scenarios

Nils Einecke*, Manuel Mühlig†*, Jens Schmüdderich* and Michael Gienger*

* Honda Research Institute Europe
Carl-Legien-Strasse 30
63073 Offenbach, Germany

† CoR-Lab Research Institute for
Cognition and Robotics
Universitätsstr. 25
33615 Bielefeld, Germany

***Abstract*—Humanoid robots are intended to act and interact in dynamically changing environments in the presence of humans. Current robotic systems are usually able to move in dynamically changing environments because of an inbuilt depth and obstacle sensing. However, for acting in their environment the internal representation of such systems is usually constructed by hand and known in advance. In contrast, this paper presents a system that dynamically constructs its internal scene representation using a model-based vision approach. This enables our system to approach and grasp objects in an previously unknown scene. We combine standard stereo with model-based image fitting techniques for a real-time estimation of the position and orientation of objects. The model-based image processing allows for an easy transfer to the internal, dynamic scene representation. For movement generation we use a task-level whole-body control approach that is coupled with a movement optimization scheme. Furthermore, we present a novel method that constrains the robot to keep certain objects in the FOV while moving. We demonstrate the successful interplay between model-based vision, dynamic scene representation, and movement generation by means of some interactive reaching and grasping tasks.**

I. INTRODUCTION

Nowadays, humanoid robots have reached a technical state where they are not only capable of walking but also have the ability to grasp and manipulate objects. Furthermore, algorithms for planning the motion of these robots have matured and yield reliable and natural results. One key element for robots that act in dynamic environments is a robust perception of the immediate environment.

There are several approaches that try to tackle the problem of scene recognition and object manipulation in dynamic environments. The authors of [1] present a method for grasp planning in complex scenes. They show that with the use of motion capture data and known object geometries, a stable and collision-free grasping is possible in cluttered scenes. Similarly in [2] object shapes are predefined and a modeling phase is included in which 3D object meshes are designed, and later matched to the visual perception. By this means the robot is able to evaluate different grasp hypotheses based on the known geometries. In our previous work [3] we also predefined object shapes and matched them to stereo-vision input, which allowed a robot to grasp objects on a table. All these approaches rely on predefined 3D object models. This assumes a "closed world", which is not applicable to everyday environments as the amount of different object shapes is too vast.

Without predefining object shapes, one needs to rely on high-quality sensor information about the environment. In [4] the ability to extract a 3D scene representation by using a continuously tilting laser rangefinder is shown. This representation is good enough to grasp box-like and cylindrical objects that are not known before. The authors of [5] impressively show that in static scenes purely vision-based input from a robotic head with 4 cameras can be enough to allow for a model-free grasping of objects. A grasping plane is matched to the 3D point cloud of a segmented object to achieve a top-grasp with a 6-DOF robotic arm. In [6] a model-free approach is presented that incorporates a learning algorithm to infer 3D grasp points on objects given their image. In combination with an obstacle map generated from stereo vision, a robotic arm is able to unload objects from a dish washer.

In conclusion there are several shortcomings in current state-of-the-art systems. Either, the robot's internal model of the environment is predefined, the scene has to be static and analyzed thoroughly, or additional external sensory data (e.g., from a motion capturing system) is necessary to enrich the robot's internal model. There are two reasons for this: First, the visual perception of the environment is still an open issue. Second, the evaluation of motion planning algorithms is more direct if errors in the model generation of the environment can be neglected.

In this paper, we go into the direction of a real-time, vision-based generation of internal scene representations for unconstrained environments. For this, we combine standard block-matching stereoscopic depth estimation and model-based stereoscopic depth estimation with visual segmentation and line detection. It is important to note that the model-based depth estimation is using very generic shape primitives which are not explaining whole object shapes but rather important parts of objects. This enables the robot to interact in environments that have neither been seen in advance nor been predefined by hand. We use a standard block-matching stereo together with a color segmentation to detect the human tutor and for learning the color of objects the tutor is presenting to the robot. The model-based stereoscopic depth estimation is used to robustly estimate accurate 3D positions and orientations of the handles of graspable objects and planar surfaces for collision avoidance and path planning. The handles of the objects are detected in the 2D images by means of a Hough-

978-1-61284-380-3/11/$26.00 ©2011 IEEE

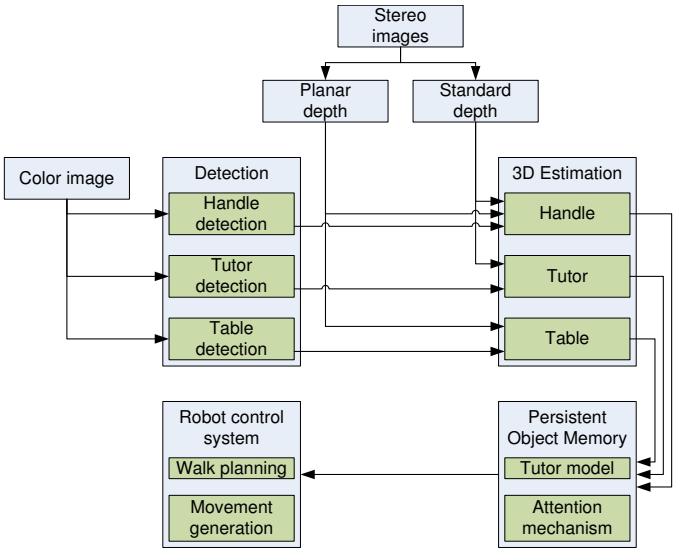


Fig. 1: System overview

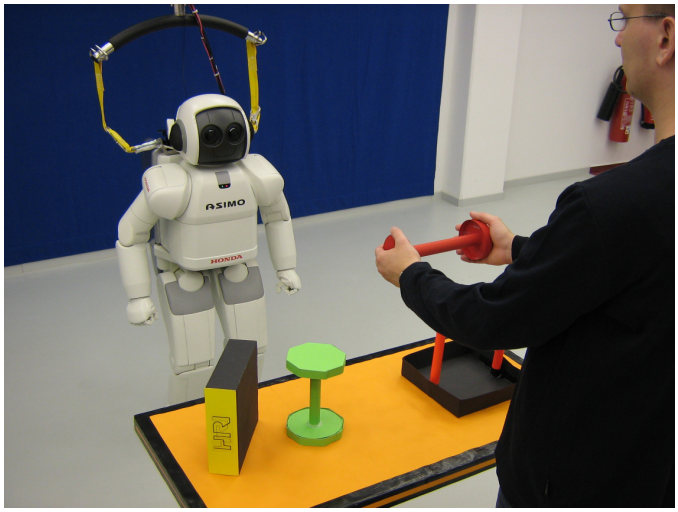


Fig. 2: In the target scenario a tutor presents an object to the robot. Afterwards the tutor asks the robot to bring that object to him. In order to perform this task the robot has to estimate the object, the tutor and possible obstacles.

Line detection stage and enriched with 3D data from the stereo processing. This generic approach does not assume certain object shapes but only requires a graspable object (irrespective of its appearance) to have a straight handle-like part which the robot can grasp. In Section II these steps are presented in more detail.

In order to represent the objects and their 3D position and orientation in a scene, we use a *Persistent Object Memory* (POM). Movement planning is done on this internal world model. We come back to these aspects in Section III. For illustration, Fig. 1 shows a simplified system overview.

With the experiments in Section IV, we demonstrate the system's flexibility by grasping different objects from a table's surface. The objects and the table are unconstrained with respect to their 3D position and orientation. Moreover, we show that our proposed system can cope with different tabletops by using a rectangular and a circular surface.

II. SCENE PERCEPTION

In this section we detail the stereoscopic depth estimation, the detection of objects, and their 3D position and pose estimation. We target for situations as displayed in Fig. 2.

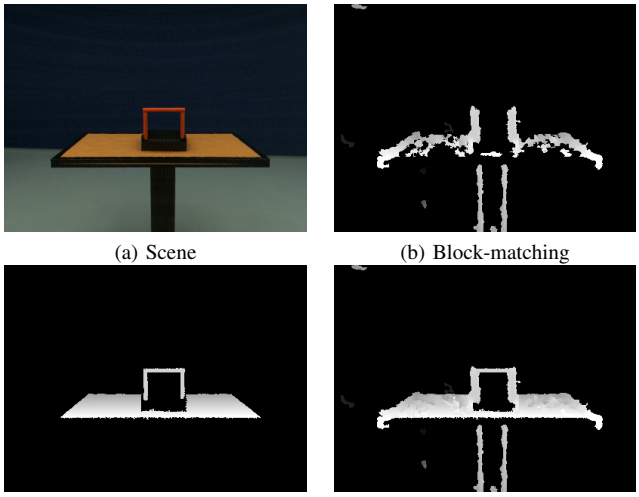
A. Stereoscopic Depth Processing

The robot we use for our experiments is equipped with a stereo camera which allows for a generic depth estimation. In order to be able to react and interact with real-time speed, we use a standard block-matching algorithm for the stereoscopic depth estimation. As we are dealing with real-world environments, it is also important that the stereo algorithm is robust against challenging lighting conditions. It has been shown [7] that for block-matching stereo the normalized cross-correlation (NCC) is one of the most robust matching costs for radiometric changes between the stereo images. In our system, we use a variant of the NCC which is called *summed normalized cross-correlation* (SNCC) [8]. SNCC reduces the so-called fattening effect at depth discontinuities thus producing more accurate results than NCC while having the same robustness with respect to illumination changes.

The standard stereo computation creates a fast feed-forward depth map of the scene which is useful for a holistic scene analysis like obstacle extraction for a rough collision avoidance. Unfortunately, block-matching stereo suffers from the so-called aperture problem, i.e. ambiguous correspondences between the stereo images. Fig. 3a and 3b show two instances of the aperture problem. One problem are the ambiguities caused by structures that are aligned with the horizontal epipolar search lines. Due to this problem, the horizontal part of the reddish basket handle cannot be estimated. Another problem pose weakly textured surfaces like the ocher table top. As can be seen in Fig. 3b the block-matching stereo approach produces a quite sparse depth map for the table.

One way to tackle these problems is to use larger patches for stereo correspondence search. Unfortunately, larger patches would also lead to a very blurry depth map. Hence, we use a model-based stereo approach [9] to complement the depth maps of the standard stereo approach for critical scene elements. The basic idea behind model-based stereo approaches is to integrate parametric surface models directly into the stereoscopic correspondence search. This means that surface models are fit directly to the image data which is in contrast to the usual approach of fitting models into the disparity maps by means of RANSAC [10]. Fitting the model directly to the stereo images leads to a much higher accuracy and robustness because the original stereo input images carry the complete visual information while the disparity maps contain only the extracted depth information.

For model-based stereo, we use a recently introduced approach [9] that integrates parametric surface models directly into the stereoscopic correspondence search. In order to do so,



(a) Scene (b) Block-matching

(c) Planar stereo (d) Fused

Fig. 3: Example of aperture problems. (a) Image of a horizontal handle and a weakly textured table top. (b) Standard block-matching stereo processing fails to estimate depth for the horizontal part of the basket handle and the table's surface. (c) Result of model-based stereo using a planar model for the basket handle and the table. (d) Fused result of standard stereo and the planar stereo. The depth maps are coded in gray, near pixels are bright and far pixels are dark.

the formulative description of a parametric surface has to be rearranged for the depth z. Here, we use a planar model

$$\boldsymbol{x} = \boldsymbol{R}\left[\boldsymbol{x}' - \boldsymbol{x}_a\right] + \boldsymbol{x}_a \ , \tag{1}$$

that describes 3D world coordinates $\boldsymbol{x} = [x, y, z]^T$ on a plane relative to coordinates on a fronto-parallel plane $\boldsymbol{x}'$. The two planes differ by a rotation $\boldsymbol{R}$ about an anchor point $\boldsymbol{x}_a = [x_a, y_a, z_a]^T$

$$\boldsymbol{R} = \begin{pmatrix} \cos\alpha_y & \sin\alpha_x \sin\alpha_y & \cos\alpha_x \sin\alpha_y \\ 0 & \cos\alpha_x & -\sin\alpha_x \\ -\sin\alpha_y & \sin\alpha_x \cos\alpha_y & \cos\alpha_x \cos\alpha_y \end{pmatrix} . \tag{2}$$

Replacing the 3D world coordinates with their projections on the two-dimensional CCD chips and rearranging the plane equation (1) for the depth z leads to

$$z = f\frac{x_a \sin\alpha_y - y_a \tan\alpha_x + z_a \cos\alpha_y}{u_{Lx} \sin\alpha_y - u_{Ly} \tan\alpha_x + f \cos\alpha_y} \ , \tag{3}$$

where u_{Lx} and u_{Ly} are the pixel coordinates of the left camera image. In a second step, the parametric surface formula has to be integrated into the stereoscopic mapping equation

$$\boldsymbol{u}_R = \boldsymbol{u}_L - b\frac{f}{z}\begin{pmatrix} 1 \\ 0 \end{pmatrix}, \tag{4}$$

which relates pixel coordinates in the left $\boldsymbol{u}_L$ and the right $\boldsymbol{u}_R$ camera image under the stereo camera parameters focal length f and baseline b. Integrating the planar equation (3) into the stereoscopic mapping equation (4) leads to the planar mapping equation

$$u_{Rx} = u_{Lx} - b\frac{u_{Lx} \sin\alpha_y - u_{Ly} \tan\alpha_x + f \cos\alpha_y}{x_a \sin\alpha_y - y_a \tan\alpha_x + z_a \cos\alpha_y} \tag{5}$$

$$u_{Ry} = u_{Ly} \ . \tag{6}$$

This equation describes the view changes of a planar surface between rectified stereo camera images. In the same fashion the mapping equation for other parametric surface models can be derived. With this description the model-based correspondence search breaks down to an optimization problem, i.e. finding the model parameters that best describe the actually perceived view changes of the surface. As proposed in [9], we use the Hooke-Jeeves optimization [11] because it has some advantages over gradient descent. First, the Hooke-Jeeves optimization searches the parameter space by means of sampling which is especially useful for non-linear surface models like spheres and cylinders that would lead to very complex gradient formulas. Second, the Hooke-Jeeves optimization is quite unconstrained with respect to the matching cost. This allows us to use NCC for an estimation that is robust against illumination. Third, the Hooke-Jeeves optimization is numerically very stable since only simple arithmetic and trigonometric functions are used for the image transformations.

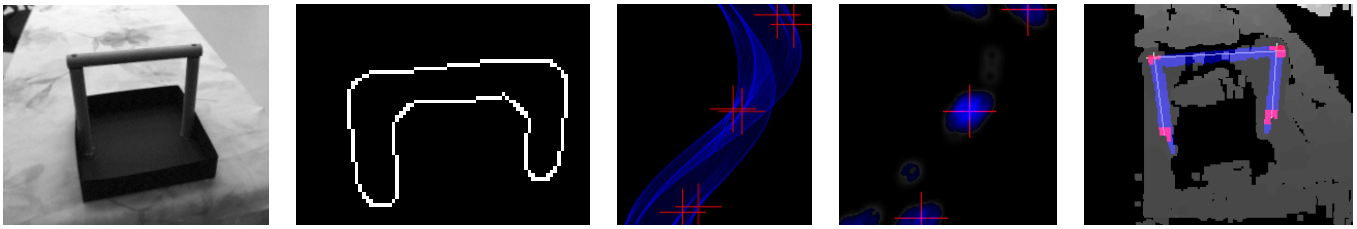
Fig. 3c shows the exemplary application of this method for estimating the handle of the basket and the surface of the weakly textured table of Fig. 3a. Unlike the block-matching approach a dense estimation is achieved. One can either use the results of the model-based depth estimation to fill the holes in the disparity of the standard stereo approach as shown in Fig. 3d or one can directly use the estimated model-parameters. In order to apply the planar stereo, a rough 2D image mask of the surface to estimate is necessary. These masks are provided in a top-down manner by a color-segmentation based on learned object knowledge. The object learning will be explained in the next section.

B. Object Detection

In the presented system the robot detects relevant objects based on a color-segmentation of the visual input. The color to be tracked is defined through interaction with a human tutor, whose hands and head are determined by means of skin color detection. The method described in this section allows to easily define relevant objects by taking an object into both hands. This allows for a natural interaction with the robot in unknown environments populated with unknown objects.

In the first step the hands of the tutor are detected by skin color segmentation and stored in a 2D binary map. Then the convex hull around both hands is calculated. The hand pixels are subtracted from the convex hull, resulting in a 2D binary map of the area between the tutor's hand. We then select those pixels from the original input image that fulfill the following conditions:

- They lie in the area between the hands.
- They have approximately the same depth as the hands.
- They are within a reasonable lighting range in order to counteract overexposure and camera noise (dark parts).



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

Fig. 4: The process of estimating 3D lines using Hough transformation. The region of the object in the input-image (a) is size-normalized, color-segmented, and edges are extracted (b). The edge image is transformed to the Hough-Space (c), where a DOG-filter is applied (d) to extract lines in the center of the object. The end-points of each line are combined with depth-information (e) to obtain a 3D line representation.

- They contain color information (saturation threshold).

The color of the object is extracted by averaging the hue of all selected pixels. In order to increase the robustness we apply a simple recursive low-pass filter and update the value only if the image has enough selected pixels. The resulting color is used to detect objects in a similar way as in [12]. Furthermore, the knowledge about an object's color can be used to generate masks for the model-based depth estimation explained in the previous section.

C. 3D Handle Estimation

After an object has been identified the 3D orientation and position of its handle need to be estimated in order to grasp the corresponding object. Assuming straight handles, we use a Hough transformation based approach.

In the first step, a rectangular region surrounding the object is estimated. This region is smoothed, size-normalized, and color-segmented, using the previously acquired target color. The subsequent application of a Canny-Edge-Detector results in an edge image as displayed in Fig. 4b for an exemplary object shown in Fig. 4a. The application of a Hough transformation is visualized in Fig. 4c: In the displayed Hough-accumulator, the x-axis codes the radius r, and the y-axis represents the angle α of each line, with respect to the origin. The intensity for each point in the Hough space indicates the support for the respective line given by the edge points in the image. In Fig. 4c six maxima can be identified (indicated by red crossed), representing the edges of the handle. This leads to two problems: First, extracting one maximum with the typical subsequent plateau-based suppression usually also suppresses the neighboring maximum, with the effect, that only one edge for each handle-bar could be detected. Second, for a precise estimation of the handle, a line lying on the center of the handle and not at the border is required.

To overcome this limitation, we filter the Hough space using a *Difference of Gaussians filter* (DOG-filter), which delivers maximal responses for areas with high Hough-activations in the surrounding, and low activations in the center. The therefore required periodic extension of the Hough space and the correct choice for the filter size are detailed in [13]. Thresholding the obtained representation leads to the image displayed in Fig. 4d. Here the red crosses indicate the areas of maximum activation.

The lines encoded by the obtained maxima are transformed back to the image-plane, where start- and end-points are determined based on the color-segmentation. Using the depth map, 3D coordinates are collected from a region around each of the start- and end-points, which is shown in Figure 4e. Median-filtering these coordinates leads to lines in 3D space.

III. MOVEMENT GENERATION

After explaining the perception of objects, this section will describe the object's representation and the robot movement generation.

A. Internal scene model

In order to allow the robot to interact with its environment, we represent all perceived scene elements within a *Persistent Object Memory* (POM). It serves as a working memory, which subsumes all sensory information and stabilizes it consistently with a mixture of low-pass, median and model-based filters. Each perceived entity is associated with a confidence value that is related to the quality of perception. If objects are occluded or have not been perceived for a certain time, the value decays. The POM also maintains a model of a human tutor whose pose is defined by the detected head and hands position (see Section II-B) and an inverse kinematics control scheme. An evaluation of the tutor's pose (e.g. if a hand is raised) allows a simple interaction with the robot. Additionally, an attention mechanism increases the saliency of objects that are touched or moved by the tutor.

The POM representation is based on a kinematic tree, which comprises the robot's links as well as all perceived entities including their geometrical properties. This allows to easily define controllers for the robot that operate directly on observed objects.

B. Whole Body Motion

To generate the motion, we employ a motion control system that is based on the redundant kinematic control scheme proposed in [14]. The task space trajectories are projected

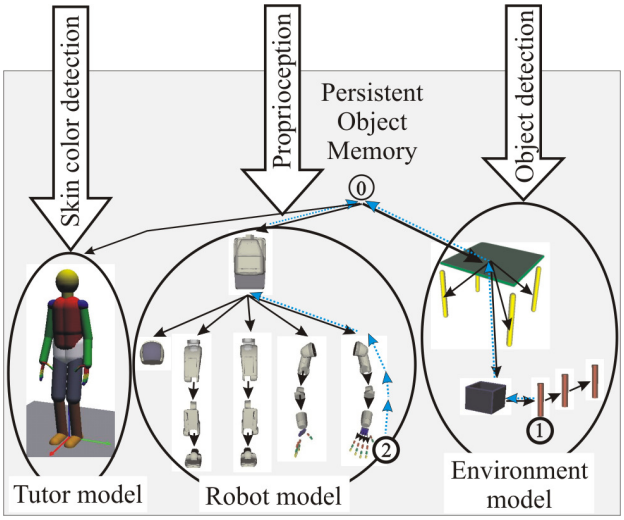


Fig. 5: Schematic view of the kinematic object memory.

into the configuration (joint) space $\boldsymbol{q}$ of the system using a weighted generalized pseudo-inverse $\boldsymbol{J}^{\#}$ of the task Jacobian:

$$\dot{\boldsymbol{q}} = \boldsymbol{J}^{\#}\, \dot{\boldsymbol{x}}_{task} - \alpha\, \boldsymbol{N}\boldsymbol{W}^{-1} \left(\frac{\partial H}{\partial \boldsymbol{q}}\right)^{T}. \tag{7}$$

This allows to track a primary control objective $\boldsymbol{x}$ precisely in the task space, while the redundant null space (projection $\boldsymbol{N}$) can be exploited to satisfy a secondary objective H, in our case a joint limit avoidance criterion. It is scaled with scalar α. We represent the trajectories $\boldsymbol{x}$ in relative coordinates, for instance the movement of the hand (body 2) with respect to an object (body 1) as depicted in Fig. 5. Details on the whole body control algorithm are given in [15]. The whole body controller is coupled with a walking and balancing controller [16], which stabilizes the motion.

C. Walk path generation

For reaching and grasping objects, it is very important to select a good target stance position of the robot with respect to the object. In our framework, this is achieved by optimizing the robot's stance position under the constraint of reaching and grasping the object (see [15] for details).

The walking path is then computed between current and target stance position using optimal control techniques. The key idea is to represent the walk path as a sequence of attractor points. Employing the movement optimization described in [17], we determine a path that when being followed is collision-free and smooth. A separate step pattern generator ensures that the foot steps are aligned with the desired path.

D. Visual-Frustrum Constrained Motion

In dynamic scenes with many objects at arbitrary locations, it is hard to ensure that the task-relevant objects are tracked consistently. Common saliency mechanisms usually track one or several objects without consideration of the limitation of the field of view. In mobile robotics, additional degrees of freedom are available like increasing the distance to the objects

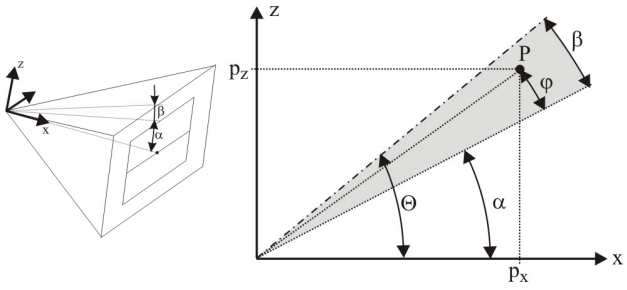


Fig. 6: Camera view frustrum description.

by moving backwards. In this section, we propose a method to track a set of objects in a scene by considering the constraints of the visual field of view of the camera, and mapping it to all available degrees of freedom of the robot. It results not only in changing the camera's view direction, but for instance also in increasing the distance to the scene when the objects are too scattered to fit in the current field of view. For this, let's consider the problem to align a frustrum so as to comprise a set of points to the frustrum's enclosing planes. The frustrum shall have all six rigid body degrees of freedom. Its angular half range is given by angle Θ. As depicted in Fig. 6, the gray area should be the range to be avoided. It is determined by angle β. Now we compute a penalty $x_f = k\varphi^2$ which is a measure of the penetration of point $\boldsymbol{p}$ into the forbidden area. The overall penalty is $\sum_{i=1}^{points} c_{f,i}$, the sum of penalties of the points enclosing the set of objects to be visible.

In order to project this penalty on the overall robot's movement, we derive the gradient $\frac{\partial x_f}{\partial \boldsymbol{q}}$ with respect to the robot's configuration (joint) space $\boldsymbol{q}$. For the two-dimensional case it is

$$\frac{\partial x_f}{\partial \boldsymbol{q}} = \frac{\partial x_f}{\partial \boldsymbol{\varphi}}\, \frac{\partial \boldsymbol{\varphi}}{\partial \boldsymbol{p}}\, \frac{\partial \boldsymbol{p}}{\partial \boldsymbol{q}} \tag{8}$$

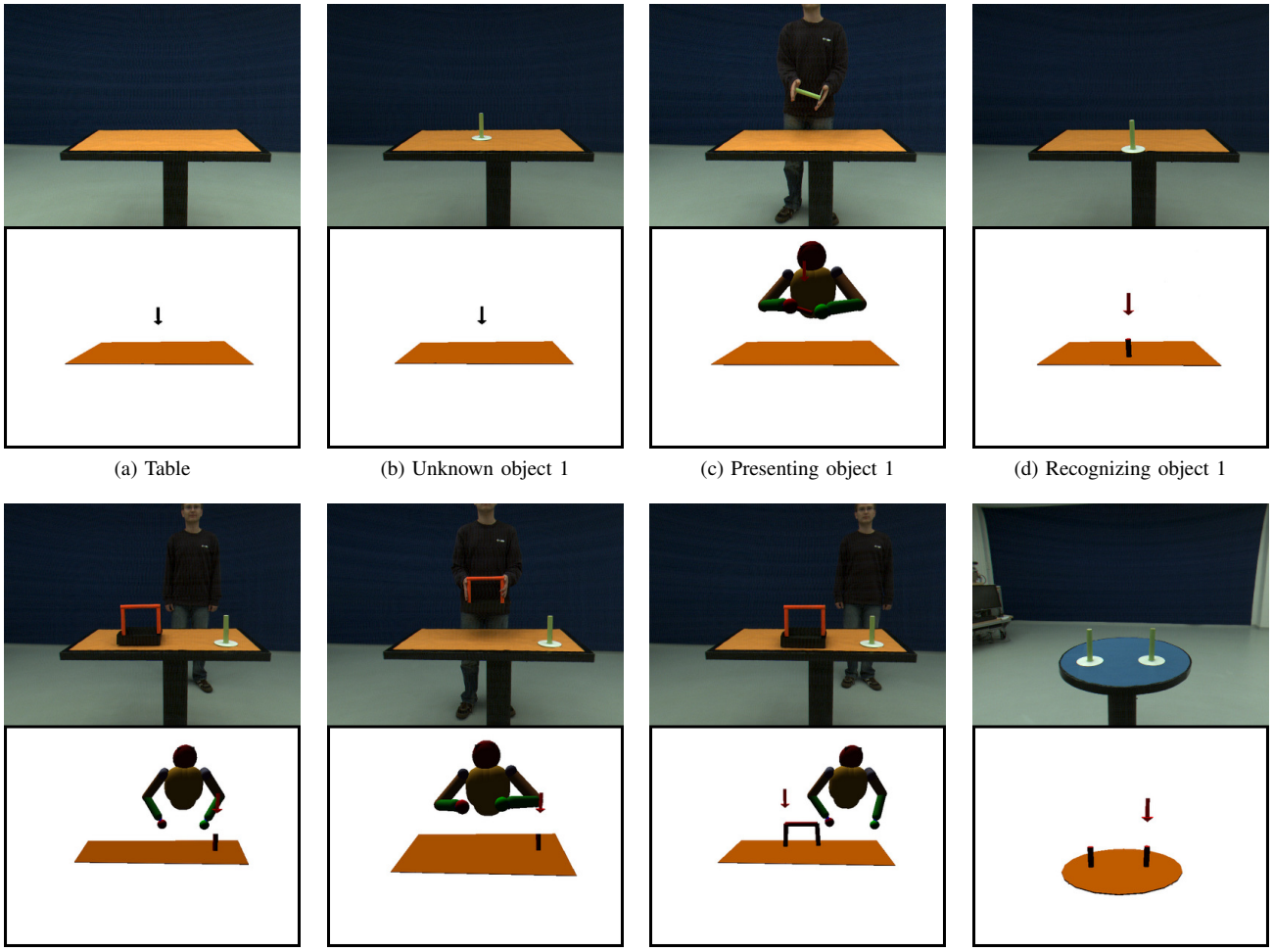
where $\frac{\partial x_f}{\partial \boldsymbol{\varphi}} = 2k\varphi$, $\frac{\partial \varphi}{\partial \boldsymbol{p}} = \left(-\frac{p_z}{p_x^2+p_z^2}\ 0\ \frac{p_x}{p_x^2+p_z^2}\right)^T$ and $\frac{\partial \boldsymbol{p}}{\partial \boldsymbol{q}}$ is the linear Jacobian of point $\boldsymbol{p}$. The 3D case can easily be derived using the chain-rule. Augmenting vector $\boldsymbol{x}$ and Jacobian $\boldsymbol{J}$ of eq. (7) with the penalty x_f and the gradient of eq. (8), realizes the desired whole-body movement behavior.

IV. EXPERIMENTS

With these experiments, we will demonstrate two major points: First, we will show the suitability of the presented scene perception and representation for our targeted interaction scenarios. Second, we will demonstrate that the integration of perception, representation, and movement generation is well applicable for manipulative and collision avoidance tasks.

A. Scene Perception

As explained in Section II, our system uses a set of diverse vision algorithms to perceive its environment. The gathered information is used to dynamically construct an internal representation that the robot can use for planning grasping and movement trajectories. Fig. 7 shows a sequence of the robot's camera views together with the corresponding



(a) Table (b) Unknown object 1 (c) Presenting object 1 (d) Recognizing object 1

(e) Unknown object 2 (f) Presenting object 2 (g) Recognizing object 2 (h) Round Table

Fig. 7: Internal scene representation. (a) A table is recognized. (b) Unknown object on the table is not recognized. (c) Human tutor is presenting the unknown object, shifting the robot's attention to it. (d) Robot detects the now known object. (e) Another unknown object. (f) Human tutor presents the new object. (g) Robot has shifted its attention to the new object. (h) Recognition of a round table with two recognized objects.

internal representation. First, a table is placed into the scene. The robot notices the table by means of its color, estimates the table's position and orientation and adds the table to its internal representation. As long as the table is visible the robot will keep track of the table's positional parameters. Next, an object is placed on top of the table. As the object is unknown to the robot it will ignore this object. In a next step, the tutor takes the object into both hands making the robot recognize the object as being important and shift its attention to it. The robot extracts the object's color in order to be able to detect it without tutor interaction. If a new object with a different color is placed on the table the robot will ignore the object because it is unknown to him. Again the tutor can present the new object to the robot which will make the robot shift its attention from the old object to the new one. After the robot has extracted the new object's color it will recognize this new object without the tutor's interaction. The last image shows a different scene setup with a round table and two objects that are the identical. This demonstrates that our system is not tailored to a specific type of table and that it does not expect only one instance of the object it is currently focused on.

For an assessment of the accuracy of the visual perception, we measured the mean and variance of the visual estimation over time. We recorded 350 image frames of the scene in Fig. 7g and 200 image frames of the scene in Fig. 7h. Table I shows the results for the position, orientation and the size of the objects in these scenes. Please note that the handle of the basket in Fig. 7g is split into three straight parts due to the Hough line extraction. The position of the estimated objects are the (x, y, z) coordinates. For these we have no ground truth values. However, the standard deviation of the estimation is in the most cases below one centimeter, i.e. the estimation is very stable. The z-vector in Table I denotes the plane normal for the table and the symmetry axis of the straight handle parts. Here the standard deviation is between one and four degrees. Again we have no ground truth information but as one would

object	μ position (m)	σ pos. (10^{-3}m)	μ z-vector (m)	σ z-vector (deg)	μ size (m)	real size (m)	σ size (10^{-3}m)
rect. table	(1.47, 0.14, 0.72)	(3.3, 0.86, 0.85)	(-0.36, -0.01, -0.93)	1.02	(0.46, 1.02)	(0.5 1.0)	(9.6, 2.9)
basket left	(1.54, 0.26, 0.82)	(8.6, 2.0, 1.8)	(0.18, 0.00, -0.98)	4.68	0.1234	0.15	2.8
basket up	(1.55, 0.16, 0.89)	(7.0, 1.1, 1.6)	(0.99, 0.00, 0.02)	0.63	0.18	0.20	2.7
basket right	(1.53, 0.05, 0.82)	(6.3, 1.4, 1.4)	(0.13, -0.00, -0.99)	2.62	0.11	0.15	2.4
round table	(1.43, 0.04, 0.71)	(16.7, 13.1, 3.1)	(-0.01, -0.22, -0.97)	4.29	0.2956	0.3	12.0
left object	(1.38, -0.07, 0.76)	(25.9, 2.9, 7.3)	(0.05, -0.00, -0.99)	1.95	0.15	0.14	3.5
right object	(1.39, 0.16, 0.77)	(33.4, 3.6, 9.0)	(0.05, -0.00, -0.99)	1.96	0.14	0.14	3.2

TABLE I: Accuracy of the visual perception over 350 frames of the scene in Fig. 7g and for 200 frame of the scene in Fig. 7h. Position, sizes and vectors are in meters and the vector deviation in degree. The size of the rectangular table is the (x,y) size, the size of the handle elements is their length and the size of the round table is the radius. The basket handle is split up into its three straight parts.

expect, the vector for the table and the vertical handles is pointing upward and the vector of the horizontal handle part is pointing to the side.

The experiments show that the estimation is accurate enough for grasping and collision avoidance. In contrast to the position and orientation, we have ground truth data for the size of the objects. In Table I the size for the rectangular table denotes its x and y elongation, for the round table it denotes the radius and for the straight handle parts their length. One can see that the size is estimated quite accurately up to 1-2 centimeters. It has to be noted here that the estimation error is a combined error of the depth estimation, the segmentation and the 2D Hough line extraction.

B. Delivering Experiment

In the last section, we have shown how the robot dynamically constructs an internal representation of its environment. As the representation is extracted from the current visual input it is generic and allows the robot to interact and move in unknown scenes. We demonstrate this by means of a delivering task that involves the unconstrained 3D position and orientation estimation of an previously unknown object, the grasping of the object, and the delivering of the object to the human tutor. Fig. 8 shows an internal view sequence of the robot while performing such a task.

In the first image the robot looks at the scene using the gaze behavior described in Section III-D. The next two frames show the robot approaching the table for grasping the object. The walk path is determined by the algorithm described in Section III-C and depicted by a thin dotted line on the floor. Frames four and five show a grasp movement of the robot using the whole-body motion control described in Section III-B. In frames six to eight the robot walks towards the tutor thereby avoiding to collide with the table. Again the planned path is depicted by the dotted line. In the last two frames the robot hands over the object to the human tutor.

It is important to note that neither the position and orientation of the object nor the position, orientation and physical extent of the table are known to the robot in before. The robot estimates all this data from its visual input. Here the estimation of the table is very important because its physical presence is important for planning the grasping (not trying to reach through the table) and for planning the movement towards the human tutor (not colliding with the table). As the sequence in Fig. 8 shows the robot successfully estimates its surrounding and successfully plans and executes the grasping of the object and the movement to the human tutor for delivering the demanded object.

V. CONCLUSION

In this work, we presented a system that couples a visual scene perception, a Persistent Object Memory and a task-level whole-body control for interacting in previously unknown scenes with unknown objects. For a stable and robust visual 3D perception, we have coupled a standard block-matching stereo approach with a model-based depth estimation that works on stereo images. Furthermore, we use a Hough line approach to detect object handles independent of object identity, thus allowing for a generic object manipulation. The robot uses this visual information to dynamically construct a representation of the current scene. In contrast to other approaches no a priori object or scene knowledge is used. By means of the integration with a dynamic movement generation the robot is able to use its online 3D scene estimation to plan and successfully execute grasping and delivering of objects, thereby avoiding detected obstacles.

Although the presented system integrates a variety of algorithms from different domains it still has some limitations. First, the current systems extracts only the color of objects in order to detect them. This could be extended by an object classifier like [18] in the future. Second, the segmentation is based solely on color. Here further research has to be done to allow for a more generic object segmentation. For example a segmentation of textured objects could be done by using also disparity data. Third, the robot will forget about the last object when it shifts its attention to a new one. By integrating a memory architecture like [19] the robot could store and remember also multiple objects. Last but not least, the obstacle avoidance is only based on the detected objects. It is important to complement this with a generic obstacle avoidance using the nevertheless computed disparity maps to avoid also objects that are not actively recognized.

REFERENCES

[1] D. Berenson, R. Diankov, K. Nishiwaki, S. Kagami, and J. Kuffner, "Grasp planning in complex scenes," in *IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, December 2007.

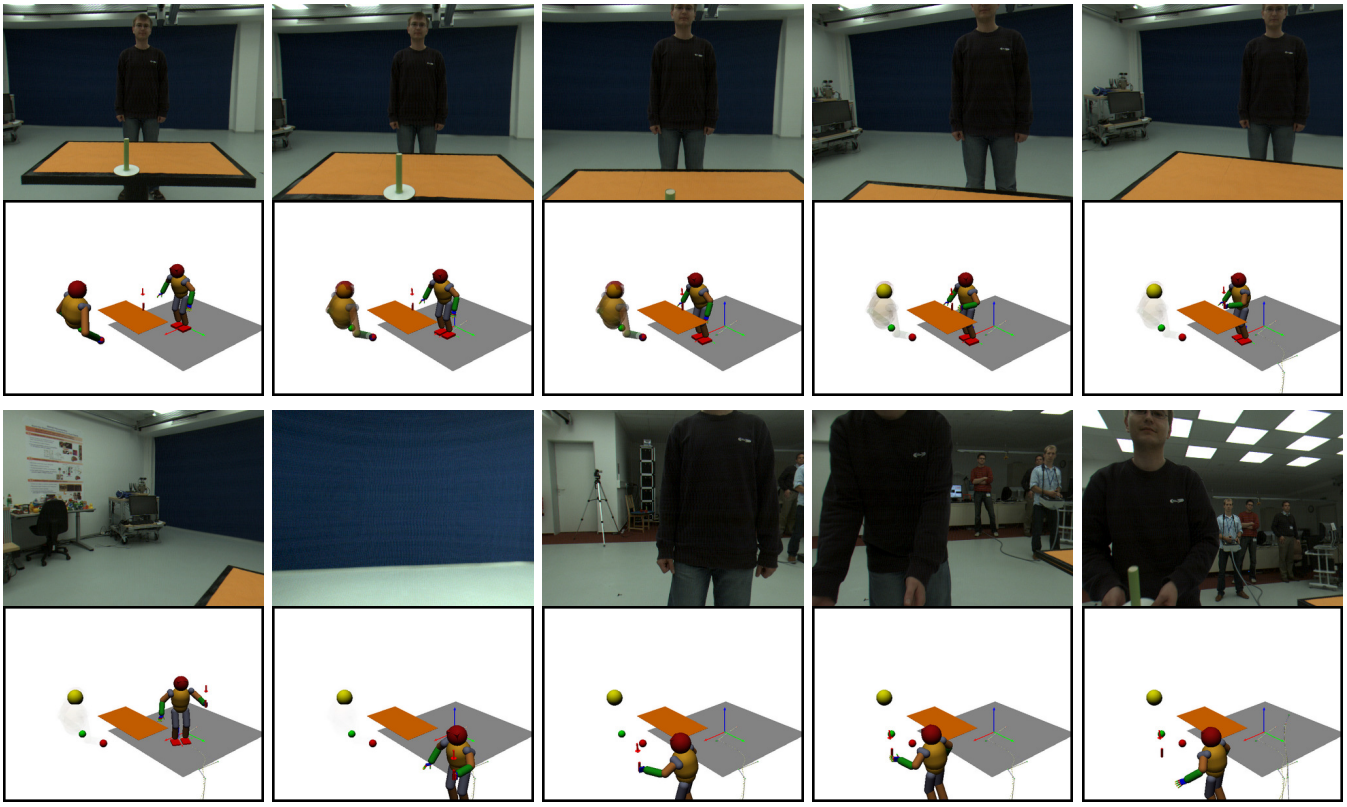


Fig. 8: Image sequence of the robot's camera view and its internal representation for an object delivering scenario. First the human tutor indicates an object the robot shall deliver. Then the robot approaches the table and grasps the object using the depth estimation for judging the 3D position and orientation of the object. Afterwards the robot walks to the tutor avoiding the table obstacle and hands over the object.

[2] K. Huebner, K. Welke, M. Przybylski, N. Vahrenkamp, T. Asfour, D. Kragic, and R. Dillmann, "Grasping known objects with humanoid robots: A box-based approach," in *Advanced Robotics, 2009. ICAR 2009. International Conference on*, 2009, pp. 1–6.

[3] M. Mühlig, M. Gienger, and J. J. Steil, "Human-robot interaction for learning and adaptation of object movements," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2010.

[4] R. B. Rusu, I. A. Sucan, B. P. Gerkey, S. Chitta, M. Beetz, and L. E. Kavraki, "Real-time perception-guided motion planning for a personal robot," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, St. Louis, 11/10/2009 2009, pp. 4245–4252.

[5] B. Rasolzadeh, M. Björkman, K. Huebner, and D. Kragic, "An active vision system for detecting, fixating and manipulating objects in the real world," *Int. J. Rob. Res.*, vol. 29, no. 2-3, pp. 133–154, 2010.

[6] A. Saxena, L. Wong, M. Quigley, and A. Y. Ng., "A vision-based system for grasping novel objects in cluttered environments," in *International Symposium of Robotics Research (ISRR)*, 2007.

[7] H. Hirschmüller and D. Scharstein, "Evaluation of stereo matching costs on images with radiometric differences," *PAMI*, vol. 31, no. 9, pp. 1582–1599, September 2009.

[8] N. Einecke and J. Eggert, "A two-stage correlation method for stereoscopic depth estimation," in *International Conference on Digital Image Computing: Techniques and Applications (DICTA)*, 2010, pp. 227–234.

[9] N. Einecke, S. Rebhan, V. Willert, and J. Eggert, "Direct surface fitting," in *International Conference on Computer Vision Theory and Applications (VISAPP)*, 2010, pp. 125–133.

[10] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[11] R. Hooke and T. A. Jeeves, ""Direct Search" Solution of Numerical and Statistical Problems," *Journal of the Association for Computing Machinery*, vol. 8, no. 2, pp. 212–229, 1961.

[12] B. Bolder, M. Dunn, M. Gienger, H. Janssen, H. Sugiura, and C. Goerick, "Visually guided whole body interaction," in *IEEE International Conference on Robotics and Automation (ICRA 2007)*, 2007.

[13] J. Schmuedderich, *Multimodal Learning of Grounded Concepts in Embodied Systems*, ser. Berichte aus der Robotik. Shaker Verlag GmbH, Germany, April 2010. [Online]. Available: http://www.amazon.co.uk/Multimodal-Learning-Grounded-Concepts-Embodied/dp/3832290737

[14] A. Liégeois, "Automatic supervisory control of the configuration and behavior of multibody mechanisms," in *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-7 no. 12, December 1977.

[15] M. Gienger, M. Toussaint, and C. Goerick, "Whole body motion planning - building blocks for intelligent systems," in *Motion Planning for Humanoid Robots*, 1st ed., K. Harada, E. Yoshida, and K. Yokoi, Eds. Springer, 2010.

[16] M. Hirose, Y. Haikawa, T. Takenaka, and K. Hirai, "Development of humanoid robot Asimo," in *IEEE/RSJ International Conference on Intelligent Robots and Systems — Workshop 2*, 2001.

[17] M. Toussaint, M. Gienger, and C. Goerick, "Optimization of sequential attractor-based movement for compact movement representation," in *Proceedings of the IEEE-RAS/RSJ International Conference on Humanoid Robots*, Pittsburgh, USA, December 2007.

[18] H. Wersing, S. Kirstein, M. Goetting, H. Brandl, M. Dunn, I. Mikhailova, C. Goerick, J. J. Steil, H. Ritter, and E. Koerner, "Online learning of objects in a biologically motivated visual architecture," *International Journal of Neural Systems*, vol. 17, no. 4, pp. 219–230, 2007.

[19] S. Rebhan, F. Röhrbein, J. Eggert, and E. Körner, "Attention modulation using short- and long-term knowledge," in *Proceedings of the 6th International Conference on Computer Vision Systems*, ser. LNCS, vol. 5008, 2008, pp. 151–160.