The 2009 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 11-15, 2009 St. Louis, USA

Fast Detection of Arbitrary Planar Surfaces from Unreliable 3D Data

Martin Heracles, Bram Bolder and Christian Goerick

Abstract— **Man-made real-world environments are dominated by planar surfaces many of which constitute behavior-relevant entities. Thus, the ability to perceive planar surfaces is vital for any embodied system operating in such environments, be it human or robotic. In this paper, we present an architecture for detection and estimation of planar surfaces in the scene from calibrated stereo images. They are represented in a behavior-oriented way, focusing on geometrical properties that are relevant for enabling basic interaction between a robot and the planar surfaces it perceives. Ego-motion of the robot is compensated for by transforming the representations into a global coordinate system using the kinematics of the robot. Our architecture is able to detect and estimate arbitrary planar surfaces, regardless of their visual appearance, their geometrical properties other than planarity and their being static or arbitrarily moving. The latter is achieved by processing each frame independently of the others. Stable representations are obtained by establishing spatio-temporal coherence between the single-frame representations of subsequent frames. Based on a RANSAC approach to plane fitting, our method is robust to unreliable 3D data such as obtained by local stereo correlation, for example. In our experiments using the Honda humanoid robot ASIMO, we show that our method is able to provide a robot in real-time with representations of planar surfaces in its environment that are sufficiently accurate for basic interaction.**

I. INTRODUCTION

Man-made environments are dominated by planar surfaces. Many of these constitute behavior-relevant entities in the sense that they *afford* certain actions [1]. For example, the seat base of a chair affords sitting. Humanoid robots operate in the same man-made environments as human beings do. Therefore, they are confronted with planar surfaces to the same extent. Due to the similar embodiment humanoid robots share with human beings, planar surfaces afford similar actions to them. Thus, the ability to perceive planar surfaces in the environment is as vital for humanoid robots as it is for human beings.

Planar surfaces in man-made environments vary greatly in terms of their visual appearance, including their color, their texture or lack thereof. They also have very different geometrical properties, for example in terms of their orientation, their shape and size. And even if most planar surfaces in man-made environments are typically static, there are behavior-relevant planar surfaces that can be moved freely — imagine a person holding a tray, for example. Thus, a *generic* perception mechanism for planar surfaces is required, i.e. the system should be able to perceive *all* planar surfaces in its environment.

In our understanding, perception serves to enable an embodied system such as a robot to *interact* with its environment. From this, two implications arise. First, a reasonable degree of accuracy has to be achieved regarding the internal representations of the planar surfaces perceived by the system: If, on the one hand, the representations are too coarse, the system will not be able to interact with them in a meaningful way and even runs the risk of damaging itself by collision. On the other hand, if the accuracy of the representations is higher than what is necessary for safe interaction, computational resources are wasted.

This leads to the second implication: The entire system has to perform in real-time, otherwise it will be unable to properly interact with its environment as well, no matter how accurate its internal representations may be. This imposes limitations on the choice of algorithms that can be used as part of the system. For example, due to the similar embodiment humanoid robots share with human beings they typically infer 3D information about the scene by stereo vision, which is also used by human beings [2]. Although there are well-known algorithms for stereo correlation that achieve highly accurate 3D data [3], humanoid robots often use local correlation methods because of their comparably low computational cost, even if the resulting 3D information is by far more inaccurate and unreliable.

In humanoid robotics, the range of applicable methods is often constrained further by the philosophy to go without non-biological methods such as laser scanners, for example. Real-world environments often involve additional challenges, ranging from inhomogeneous lighting conditions to occlusions. For example, a table top may be partially occluded by objects on the table. Thus, any perception mechanism of an embodied system operating in real-world environments needs to be sufficiently *robust* to these influences. In particular, the internal representations it provides to the system must be sufficiently accurate and stable for interaction despite the inaccurate and unreliable 3D information from which they are computed.

This paper is organized as follows: In Sec. II, we provide a brief overview of existing approaches to planar surface detection and estimation of the corresponding plane parameters. After that, we formally describe and explain our own approach in Sec. III. In Sec. IV, we evaluate our method in a real-world experiment using the Honda humanoid robot ASIMO and discuss the results. Finally, we summarize our main results and contributions in Sec. V.

M. Heracles, B. Bolder and C. Goerick are with the Honda Research Institute Europe, D-63073 Offenbach/Main, Germany {martin.heracles, bram.bolder, christian.goerick}@honda-ri.de

M. Heracles is also with the Research Institute for Cognition and Robotics, Bielefeld University, D-33615 Bielefeld, Germany heracles@cor-lab.uni-bielefeld.de

II. RELATED WORK

The problem of planar surface detection and estimation of the corresponding plane parameters has been studied both in computer vision and robotics. Most of the approaches can be classified as iterative methods, voting-based methods or methods employing a growing procedure. They can be further distinguished by whether they operate on dense or sparse 3D features.

If the detection task is simplified, e.g. by manual selection, the parameters of multiple planes can be estimated by iterative plane fitting [4] or by iteratively refining initial estimates [5]. For the detection and estimation of certain behavior-relevant planar surfaces such as the ground plane, for example, methods using V-disparity [6] or a model of the ground plane disparity [7] have been proposed. If good initial estimates are available, the ground plane can also be estimated by iteratively adapting the underlying homography like in [8] and [9]. Other methods are able to detect and estimate the dominant plane in the scene based on a growing procedure on coplanar sets of sparse 3D points [10] [11]. Textured planar surfaces can be detected and estimated by performing a voting procedure on plane orientations obtained from pairs of spectral peaks [12] or on dense local normals [13] [14]. Finally, methods that circumvent the need for texture by operating on sparse 3D data have been proposed, including a voting procedure on plane candidates defined by point and line features [15] or by triplets of points [16], as well as a growing procedure on the normals of triangles obtained by triangularization of the 3D points [17].

III. METHODS

A. General Considerations

The characteristic property of a planar surface is coplanarity of the 3D points corresponding to its projection in the image plane. Since this property goes beyond the 2D image itself, any system aiming at the detection of planar surfaces and the estimation of their parameters has to take into account 3D information about the scene. This 3D information can be obtained by various means, e.g. by stereo vision, but also by a laser scanner or a combination of several sources in order to increase accuracy and reliability. We suppose the resulting 3D information is organized as *3D map*

$$i_{3D} : \boldsymbol{W} \times \boldsymbol{H} \rightarrow \mathbb{R}^3$$

mapping from image pixels $(u, v) \in \boldsymbol{W} \times \boldsymbol{H}$ to 3D points

$$i_{3D}(u, v) = (x, y, z) \in \mathbb{R}^3$$

in camera coordinates.

For the reasons given in Sec. I, in our case i_{3D} is obtained by local stereo correlation [18]. As a consequence, reliable 3D information is dense within sufficiently textured image regions while homogeneous regions lack reliable 3D information. This is characteristic for any local correlation-based approach to 3D reconstruction: Correlation is ambiguous within homogeneous regions since the pixels considered by the local approach all have similar values. Nevertheless, correlation is *not* ambiguous along the *contours* of homogeneous regions (except for horizontal parts thereof, in the case of stereo vision). Therefore, we detect and estimate textured planar surfaces directly from i_{3D} and textureless planar surfaces based on the 3D data corresponding to their contours. Consequently, our system operates on i_{3D} and the corresponding camera image

$$i_{RGB} : \boldsymbol{W} \times \boldsymbol{H} \rightarrow \{0, \ldots, 255\}^3$$

mapping from image pixels $(u, v) \in \boldsymbol{W} \times \boldsymbol{H}$ to colors

$$i_{RGB}(u, v) = (r, g, b) \in \{0, \ldots, 255\}^3$$

in RGB color space. Note that our method does not require i_{RGB} to be a color image, since it considers similarity of pixels rather than their actual values (see Sec. III-B).

Given i_{3D} and i_{RGB}, our system detects and estimates both textured and textureless planar surfaces in the scene. They are represented in a behavior-oriented way: In order to enable a robot to perform basic interaction with the planar surfaces it perceives while at the same time reduce computational effort, only relevant geometrical properties such as their position, size and orientation in 3D space are represented. Thus, our system computes as output a set

$$\boldsymbol{P} = \{p_1, \ldots, p_n\}$$

of representations

$$p_i = (c_i, n_i, s_i)$$

Each p_i represents a planar surface in the scene, where

$$c_i = (c_x, c_y, c_z) \in \mathbb{R}^3$$

is a 3D point representing its position,

$$n_i = (n_x, n_y, n_z) \in \{n \in [-1, 1]^3 \mid \|n\| = 1\}$$

is a 3D normal representing its orientation, and

$$s_i = (s_x, s_y, 0) \in \mathbb{R}^3_+$$

represents its size with respect to its principal axes.

This kind of representation is consistent with neurobiological findings: In the brain, cortical areas related to behavior and motor control are closely linked with the *dorsal pathway*. As opposed to the *ventral* pathway, which primarily processes detailed object-specific information leading to conscious percepts, the dorsal pathway focuses on behavior-relevant geometrical properties in order to guide behavior [2]. Experimental evidence suggests that this also applies to planar surfaces: For example, the 3D orientation of planar surfaces is represented in the caudal part of the lateral bank of the intraparietal sulcus (area CIP) [19].

B. System Architecture

As a consequence of the above considerations, our system consists of two parallel but converging sub-systems: one is dedicated to textured planar surfaces and operates directly on i_{3D}, the other is dedicated to textureless planar surfaces and

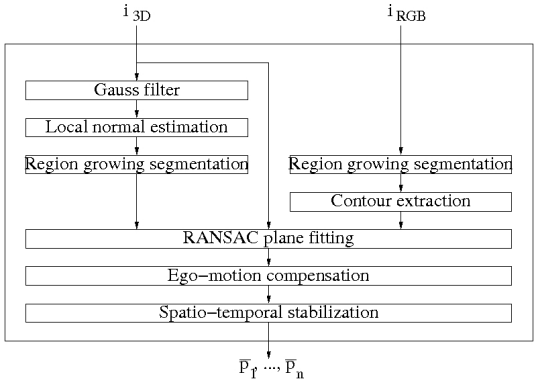

Fig. 1. Basic system architecture. See Sec. III-B for a detailed explanation.

operates on i_{RGB} and i_{3D}. The basic architecture is depicted in Fig. 1 and will be explained below.

The sub-system for textured planar surfaces consists of the following processing steps. First, Gauss filtering is performed on i_{3D}, resulting in the *smooth 3D map*

$$\widetilde{i_{3D}} : \boldsymbol{W} \times \boldsymbol{H} \rightarrow \mathbb{R}^3$$

In our case, actually the disparity map i_{disp} underlying i_{3D} is Gauss filtered, which is faster because i_{disp} only maps to a one-dimensional space. The Gauss filtering serves to prepare the next processing step in two respects: First, it reduces the overall noise level in i_{3D}. Second, it recovers smooth depth gradients. Local correlation like [18] tends to discretize depth gradients in i_{3D}. The Gauss filter is optimal for eliminating the resulting small step-like artifacts due to its ability to filter higher frequencies without affecting lower frequencies.

Based on $\widetilde{i_{3D}}$, a *local normal map*

$$i_{\vec{n}} : \boldsymbol{W} \times \boldsymbol{H} \rightarrow \{n \in [-1,1]^3 \mid \|n\| = 1\}$$

is then computed, mapping from image pixels $(u,v) \in \boldsymbol{W} \times \boldsymbol{H}$ to local surface normals

$$i_{\vec{n}}(u,v) = (n_x, n_y, n_z) \in \{n \in [-1,1]^3 \mid \|n\| = 1\}$$

at the corresponding 3D points $\widetilde{i_{3D}}(u,v)$. Each $i_{\vec{n}}(u,v)$ is obtained by performing a principal component analysis (PCA) on the set

$$\left\{\widetilde{i_{3D}}(u',v') \mid (u',v') \in \{(u,v)\} \cup \mathrm{Neigh}_8(u,v)\right\}$$

of 3D points, where $\mathrm{Neigh}_8(u,v)$ denotes the 8-neighborhood of (u,v). Note the importance of the antecedent Gauss filtering: Since the $i_{\vec{n}}(u,v)$ are computed locally, they are sensitive to noise and the aforementioned artifacts, thus they would be unreliable if computed from i_{3D} instead of $\widetilde{i_{3D}}$.

Once $i_{\vec{n}}$ has been obtained, contiguous regions characterized by similar local normals are identified by performing a region growing segmentation on $i_{\vec{n}}$. The resulting *segmentation*

$$i^{\bullet}_{\vec{n}} : \boldsymbol{W} \times \boldsymbol{H} \rightarrow \mathbb{N}$$

assigns a region label $l \in \mathbb{N}$ to each image pixel $(u,v) \in \boldsymbol{W} \times \boldsymbol{H}$. Pixels that are assigned the same region label form a region, i.e. the *region* $\boldsymbol{R}_{i^{\bullet}_{\vec{n}}}(l)$ corresponding to region label $l \in \mathbb{N}$ is defined as

$$\boldsymbol{R}_{i^{\bullet}_{\vec{n}}}(l) = \{(u,v) \in \boldsymbol{W} \times \boldsymbol{H} \mid i^{\bullet}_{\vec{n}}(u,v) = l\}$$

By construction, the $\boldsymbol{R}_{i^{\bullet}_{\vec{n}}}(l)$ are contiguous.

The sub-system for textureless planar surfaces exploits that their lack of texture implies smooth transitions or even homogeneity with respect to color. Thus, it identifies contiguous regions characterized by similar color by performing a region growing segmentation on i_{RGB}. Effects of over- and undersegmentation are dealt with at subsequent processing steps. Analogous to $i^{\bullet}_{\vec{n}}$, the resulting *segmentation*

$$i^{\bullet}_{RGB} : \boldsymbol{W} \times \boldsymbol{H} \rightarrow \mathbb{N}$$

assigns a region label $l \in \mathbb{N}$ to each image pixel $(u,v) \in \boldsymbol{W} \times \boldsymbol{H}$. The *region* $\boldsymbol{R}_{i^{\bullet}_{RGB}}(l)$ corresponding to region label $l \in \mathbb{N}$ is defined as

$$\boldsymbol{R}_{i^{\bullet}_{RGB}}(l) = \{(u,v) \in \boldsymbol{W} \times \boldsymbol{H} \mid i^{\bullet}_{RGB}(u,v) = l\}$$

Like the $\boldsymbol{R}_{i^{\bullet}_{\vec{n}}}(l)$, the $\boldsymbol{R}_{i^{\bullet}_{RGB}}(l)$ are also contiguous by construction.

The region growing approach used to obtain $i^{\bullet}_{RGB}$ and $i^{\bullet}_{\vec{n}}$ is basically the same: The $\boldsymbol{R}_{i^{\bullet}_{RGB}}(l)$ are characterized by similarity with respect to color while the $\boldsymbol{R}_{i^{\bullet}_{\vec{n}}}(l)$ are characterized by similarity with respect to the local normals. The only difference is that for the $\boldsymbol{R}_{i^{\bullet}_{RGB}}(l)$ *local* similarity of neighboring pixels is considered while for the $\boldsymbol{R}_{i^{\bullet}_{\vec{n}}}(l)$ *global* similarity with respect to the average normal of a growing region is considered. Thereby, the region growing is able to follow color gradients in i_{RGB}, which is desirable because they are often artifacts caused by inhomogeneous lighting conditions, while it does not follow gradients with respect to the local normals. This is desirable as well, since the latter do not correspond to planar but curved surfaces. In addition, since region growing evaluates similarity, not absolute values, nothing has to be known about the actual color or the actual orientation of a planar surface.

According to the considerations in Sec. III-A, for the $\boldsymbol{R}_{i^{\bullet}_{RGB}}(l)$ reliable 3D information is only available along their contours. Therefore, the next step consists in identifying the region contours of the $\boldsymbol{R}_{i^{\bullet}_{RGB}}(l)$, which is straightforward given $i^{\bullet}_{RGB}$: A pixel $(u,v) \in \boldsymbol{W} \times \boldsymbol{H}$ is a contour pixel iff it has a neighbor pixel $(u',v') \in \boldsymbol{W} \times \boldsymbol{H}$ that has a different region label, i.e. iff

$$\exists (u',v') \in \mathrm{Neigh}_4(u,v) : i^{\bullet}_{RGB}(u',v') \neq i^{\bullet}_{RGB}(u,v)$$

where $\mathrm{Neigh}_4(u,v)$ denotes the 4-neighborhood of (u,v). Note that the 4-neighborhood is sufficient here, while the 8-neighborhood is used for the computation of $i_{\vec{n}}$ because the local normals thus estimated are more reliable. The resulting *contour image*

$$i^{\circ}_{RGB} : \boldsymbol{W} \times \boldsymbol{H} \rightarrow \mathbb{N}$$

is a subset of $i^{\bullet}_{RGB}$ in the following sense: The *region contour* $\boldsymbol{R}_{i^{\circ}_{RGB}}(l)$ corresponding to region label $l \in \mathbb{N}$ is given by

$$\boldsymbol{R}_{i^{\circ}_{RGB}}(l) = \{(u,v) \in \boldsymbol{W} \times \boldsymbol{H} \mid i^{\circ}_{RGB}(u,v) = l\} \subseteq \boldsymbol{R}_{i^{\bullet}_{RGB}}(l)$$

Thus, we end up with a set

$$\boldsymbol{\mathcal{R}} = \{\boldsymbol{R}_{i^{\bullet}_{\vec{n}}}(l) \mid l \in \mathbb{N}\} \cup \{\boldsymbol{R}_{i^{\circ}_{RGB}}(l) \mid l \in \mathbb{N}\}$$

of candidate regions that are likely to correspond to textured and textureless planar surfaces in the scene, respectively. For each $\boldsymbol{R} \in \boldsymbol{\mathcal{R}}$, a RANSAC approach to plane fitting [20] verifies whether the set

$$i_{3D}(\boldsymbol{R}) = \{i_{3D}(u,v) \mid (u,v) \in \boldsymbol{R}\}$$

of 3D points corresponding to $\boldsymbol{R}$ is indeed coplanar, disregarding outliers. If successful, a plane is fitted to the coplanar subset of $i_{3D}(\boldsymbol{R})$. This is achieved as follows: First, $m \in \mathbb{N}$ plane *hypotheses*

$$p^{(1)}_{\boldsymbol{R}}, \ldots, p^{(m)}_{\boldsymbol{R}}$$

are generated for $\boldsymbol{R}$, where each

$$p^{(j)}_{\boldsymbol{R}} = (q^{(j)}_{\boldsymbol{R}}, \vec{n}^{(j)}_{\boldsymbol{R}})$$

is defined by a point $q^{(j)}_{\boldsymbol{R}} \in \mathbb{R}^3$ and a normal $\vec{n}^{(j)}_{\boldsymbol{R}} \in \{n \in [-1,1]^3 \mid \|n\| = 1\}$ in 3D space. Each hypothesis $p^{(j)}_{\boldsymbol{R}}$ is computed from three 3D points that are randomly chosen according to a uniform distribution over $i_{3D}(\boldsymbol{R})$.

After that, the *support set*

$$\overline{i_{3D}(\boldsymbol{R})}^{(j)} = \{q \in i_{3D}(\boldsymbol{R}) \mid \mathrm{dist}(p^{(j)}_{\boldsymbol{R}}, q) \leq \epsilon\}$$

is determined for each $p^{(j)}_{\boldsymbol{R}}$, consisting of all 3D points $q \in i_{3D}(\boldsymbol{R})$ that are sufficiently close to $p^{(j)}_{\boldsymbol{R}}$, where $\epsilon \in \mathbb{R}_+$ and $\mathrm{dist}(p^{(j)}_{\boldsymbol{R}}, q) \in \mathbb{R}_+$ denotes the orthogonal distance between q and $p^{(j)}_{\boldsymbol{R}}$.

Provided that a given $p^{(j)}_{\boldsymbol{R}}$ has sufficient support, i.e. that

$$\frac{\left|\overline{i_{3D}(\boldsymbol{R})}^{(j)}\right|}{|i_{3D}(\boldsymbol{R})|} \geq \tau$$

where $\tau \in [0,1]$, its *cost*

$$c(p^{(j)}_{\boldsymbol{R}}) = \frac{1}{\left|\overline{i_{3D}(\boldsymbol{R})}^{(j)}\right|} \sum_{q \in \overline{i_{3D}(\boldsymbol{R})}^{(j)}} \mathrm{dist}(p^{(j)}_{\boldsymbol{R}}, q)$$

is computed as the average distance between $p^{(j)}_{\boldsymbol{R}}$ and its support points $q \in \overline{i_{3D}(\boldsymbol{R})}^{(j)}$.

Finally, the $p^{(j^\star)}_{\boldsymbol{R}}$ with minimum cost is chosen, i.e.

$$p^{(j^\star)}_{\boldsymbol{R}} = \underset{p^{(j)}_{\boldsymbol{R}} \in \{p^{(1)}_{\boldsymbol{R}}, \ldots, p^{(m)}_{\boldsymbol{R}}\}}{\arg\min} \{c(p^{(j)}_{\boldsymbol{R}})\}$$

— or none, if no $p^{(j)}_{\boldsymbol{R}}$ has sufficient support. Since $p^{(j^\star)}_{\boldsymbol{R}}$ is only an initial hypothesis based on three 3D points, the plane parameters are re-estimated by performing a principal component analysis (PCA) on its support set $\overline{i_{3D}(\boldsymbol{R})}^{(j^\star)}$. This results in the (least-squares) best-fitting plane

$$p_{\boldsymbol{R}} = (q_{\boldsymbol{R}}, \vec{n}_{\boldsymbol{R}})$$

By computing the size $s_{\boldsymbol{R}} \in \mathbb{R}^3_+$ of $\overline{i_{3D}(\boldsymbol{R})}^{(j^\star)}$ with respect to its principal axes, we obtain

$$p'_{\boldsymbol{R}} = (q_{\boldsymbol{R}}, \vec{n}_{\boldsymbol{R}}, s_{\boldsymbol{R}})$$

in camera coordinates, which is consistent with the planar surface representation proposed in Sec. III-A.

We use RANSAC plane fitting for two reasons: First, it is computationally efficient due to its randomization. Second, it is robust to large amounts of outliers. This is very important because there are numerous sources of outliers in i_{3D}: ranging from sensor noise to mismatches during stereo correlation, objects partially occluding planar surfaces, and undersegmentation in the textureless case that causes the $\boldsymbol{R}_{i^{\circ}_{RGB}}(l)$ to contain non-coplanar parts of the scene as well.

So far we have described how $\boldsymbol{P} = \{p'_{\boldsymbol{R}} \mid \boldsymbol{R} \in \boldsymbol{\mathcal{R}}\}$ is computed from i_{3D} and i_{RGB}. The set $\boldsymbol{P}$ is computed independently for each frame. This way, our system is able to detect and estimate static planar surfaces as well as arbitrarily moving ones. Once the single-frame representations $p'_{\boldsymbol{R}} \in \boldsymbol{P}$ have been obtained, they are transformed into global coordinates using the kinematics of the robot to compensate for ego-motion. Then, spatio-temporal coherence is established in order to obtain stable representations: This is achieved by considering a short *history*

$$\boldsymbol{\mathcal{H}} = \{\boldsymbol{P}^{(f^\star)}, \ldots, \boldsymbol{P}^{(f^\star - h)}\}$$

containing the

$$\boldsymbol{P}^{(f)} = \{p^{(f)}_1, \ldots, p^{(f)}_{n_f}\}$$

of subsequent frames f, where $n_f \in \mathbb{N}_0$ denotes the number of planar surfaces that have been detected and estimated in frame f, $f^\star \in \mathbb{N}_0$ denotes the current frame and $h \in \mathbb{N}_0$ is the history length. By performing nearest neighbor clustering across the $\boldsymbol{P}^{(f)} \in \boldsymbol{\mathcal{H}}$, a set

$$\boldsymbol{\mathcal{P}} = \{\boldsymbol{P}_1, \ldots, \boldsymbol{P}_n\}$$

is obtained. The representations in each $\boldsymbol{P} \in \boldsymbol{\mathcal{P}}$ correspond to the same planar surface in the scene. Stable representations

$$\overline{\boldsymbol{P}} = \{\overline{p}_1, \ldots, \overline{p}_n\}$$

are then obtained by computing the (componentwise) average of the representations in each of the $\boldsymbol{P}_i$, i.e.

$$\overline{p}_i = \frac{1}{|\boldsymbol{P}_i|} \sum_{p \in \boldsymbol{P}_i} p$$

These representations are fully consistent with the so-called *proto-object* representations used in [21].

IV. RESULTS

In order to evaluate our method, we have performed several experiments in an indoor environment using the Honda humanoid robot ASIMO [22]. ASIMO was equipped with a calibrated stereo camera that acquires RGB color images at a resolution of 400×300 pixels. For 3D reconstruction, the commercial software described in [18] was used, which is a local stereo-correlation method. The resulting 3D maps had a resolution of 400×300 pixels as well and were available at a framerate of 23 fps. Both the 3D maps and the camera images were downsampled by a factor of 2 before being fed into our architecture.

A. Changing Visual Appearance

To begin with, we give an example of a textured planar surface and a textureless one. The purpose is to illustrate the fundamental difference in how our method reconstructs them in 3D space, and to familiarize the reader with the visualization of the resulting 3D representations.

As an example of a textured planar surface, we covered the seat base of an office stool with a textured tablecloth and presented it to the robot (see Fig. 2, upper left). The green ellipse is the result of back-projecting the corresponding 3D proto-object representation onto the image plane. Note that the proto-object representation itself is fully three-dimensional, as described in Sec. III-A: The two straight lines are in fact orthogonal, in 3D space, and represent the principal axes, their intersection corresponds to the 3D center point, and the ellipse visualizes the size. Obviously, the textured seat base is successfully detected and reconstructed.

Fig. 2, upper right, shows the disparity map underlying the 3D points from which the planar surface is reconstructed. One can see that, due to the texture of the tablecloth, dense disparities are available for reconstruction.

We then removed the tablecloth from the office stool in order to expose its textureless visual appearance (see Fig. 2, bottom left). This was done at run-time, without pausing the system and without changing any parameters. Obviously, despite the changed visual appearance the seat base is still successfully detected and reconstructed, as can be seen by the green ellipse, which resembles the one obtained in the textured case.

Fig. 2, bottom right, reveals that only sparse disparities are available in the textureless case and that these concentrate along the boundaries. As described in Sec. III-A, this is a fundamental difference between textured and textureless planar surfaces, which is the reason for the two parallel sub-systems we employ in our approach.

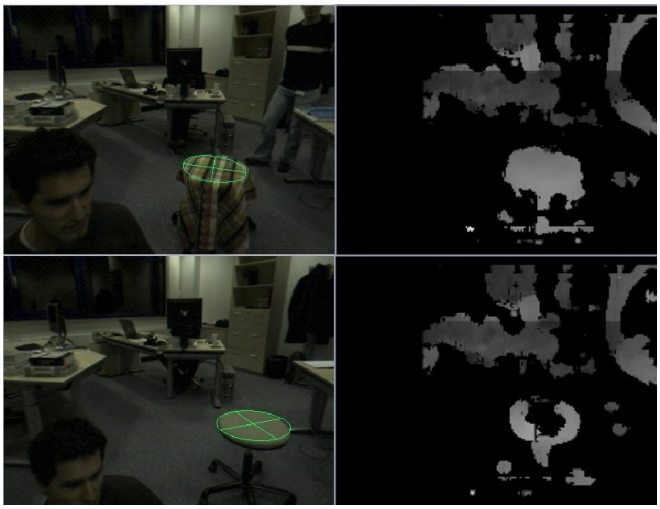

Fig. 2. Example of a textured and a textureless planar surface.

Fig. 3. Example of an object that consists of more than one planar surface.

B. Multiple Planar Surfaces

In Sec. IV-A, we have only considered a single planar surface at a time. In this experiment, we test our method on an object that consists of more than one planar surface. The planar surfaces involved differ from each other in terms of their orientation in 3D space and, in one part of the experiment, also in terms of their visual appearance.

We presented an office chair to ASIMO that consists of two different planar surfaces: seat base and backrest. While the chair itself is textureless, we first covered its backrest with a textured tablecloth, thus presenting a textured planar surface and a textureless one at the same time. In addition, we turned the chair during run-time (see Fig. 3, upper row).

Obviously, both the seat base and the backrest of the chair are successfully detected and reconstructed, independent of the viewing angle. This demonstrates that, in this case, the sub-system for detection of textured planar surfaces (backrest) and the sub-system for detection of textureless planar surfaces (seat base) are active at the same time.

Note that the drawing color of the ellipse corresponding to the backrest varies with the orientation of the backrest, ranging from red over magenta to blue. The reason is that the 3D normal (x, y, z) of each planar surface representation defines the drawing color $(r = x, g = y, b = z)$ in RGB color space in which the representation is visualized. Horizontal planes, for example, are drawn in green, fronto-parallel planes are drawn in blue, and vertical planes viewed from the side are drawn in red. Moreover, the color of a fronto-parallel planar surface rotating to a side-view varies smoothly from blue over magenta to red, as with the backrest of the chair, and the color of a horizontal plane rotating into a fronto-parallel position would range from green over cyan to blue.

In the second part of this experiment, we removed the textured tablecloth from the backrest of the chair, revealing its textureless appearance. Together with the textureless seat base, we were thus presenting an object consisting of two textureless planar surfaces at the same time (see Fig. 3, bottom row). We turned the chair the same way as in the first part of this experiment.

As can be seen from the images, both backrest and seat base of the chair are again successfully detected and reconstructed. The ellipses resemble their respective counterparts in the upper row. For a detailed quantitative analysis of the accuracy of the planar surface representations computed by our method, see Sec. IV-D.

Exemplarily considering the sub-system for textureless planar surfaces, the second part of this experiment shows that our approach is not limited to one planar surface per sub-system. This is further emphasized by the following experiment.

C. More Complex Scenes

In this experiment, we consider more complex scenes, not only an individual object in isolation. Examples of such scenes are depicted in Fig. 4, which consist of a table with several different objects on top. The planar surfaces provided by these objects vary considerably in terms of their orientation and their visual appearance.

As one can see, our system enables ASIMO to correctly perceive the most important planar surfaces corresponding to the objects on the table. As for the table top itself, occluded parts are not represented because these have insufficient 3D points corresponding to the table and not to the occluding objects. Nevertheless, the non-occluded part of the table top that is closest to ASIMO, which is for this reason the behaviorally more relevant part, is reliably represented.

D. Quantitative Analysis

In order to evaluate the 3D proto-object representations computed by our method under controlled conditions, focusing on their accuracy and their framerate, we presented a single planar surface to ASIMO and systematically varied both its distance and its orientation. For simplicity, the planar surface was always presented directly ahead and at eye level of ASIMO. Its distance was varied within a range from 3 m (farthest) to 1 m (closest), in steps of 0.25 m, which is a realistic operating range for ASIMO. At each distance, the orientation of the planar surface was varied in terms of its elevation, ranging from 90° (steepest) to 0° (shallowest) in steps of 22.5°. This experiment has been conducted twice, one time with the planar surface being textureless and one time being textured. As planar surface we chose the seat base of the stool used in Sec. IV-A, having a radius of 0.20 m.

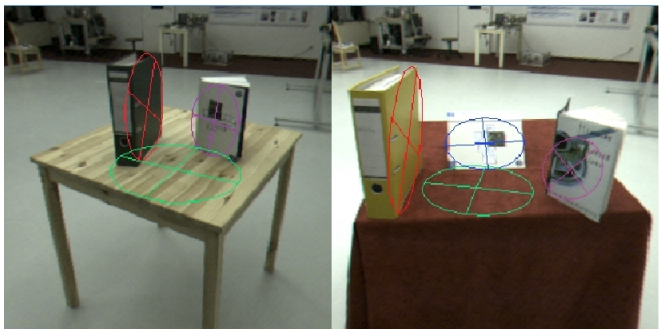

Fig. 4. More complex scenes, consisting of various objects with planar surfaces.

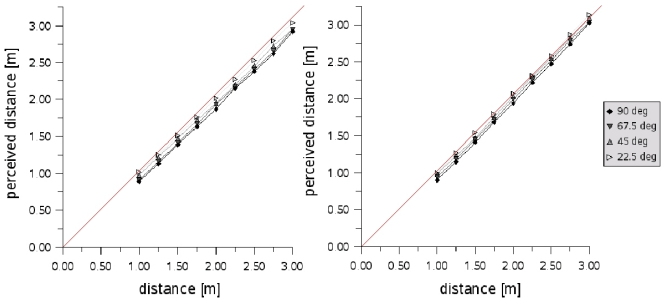

Fig. 5. Perceived distance of a textureless (left) and textured (right) planar surface, depending on its actual distance and its orientation.

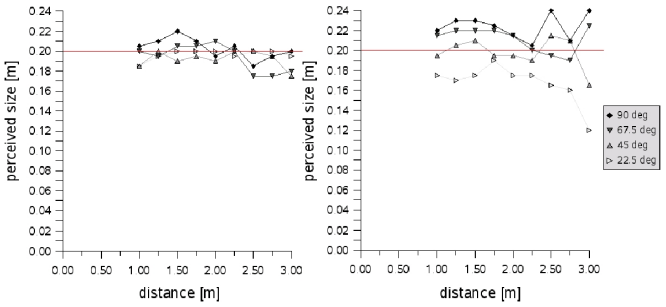

Fig. 6. Perceived size of a textureless (left) and textured (right) planar surface, depending on its distance and its orientation.

Throughout the experiment, we measured the distance, size and elevation of the planar surface as perceived by our system as well as the overall framerate. The results can be seen in Fig. 5 – Fig. 8. In each figure, the left graph shows the results for the textureless planar surface and the right graph shows the results for the textured planar surface. The red lines indicate the ground truth.

The graphs in Fig. 5 both show a strong agreement between the perceived distance and the actual distance of the planar surface. The average error is about 6 cm in the textureless case and 5 cm in the textured case. The error is not significantly affected by the distance itself: It is not affected at all in the textureless case, and not in a systematic way in the textured case. In contrast, there is a correlation between the perceived distance and the orientation of the planar surface: In both cases, the perceived distance decreases as the elevation of the planar surface increases. We attribute this to the experimental setup: For practical reasons, the center of rotation was not identical with the center point of the planar surface but rather below that point (referring to an elevation of 0°). In combination with the thickness of the planar surface, increasing its elevation thus indeed slightly decreased its distance.

The graphs in Fig. 6 both show a strong agreement between the perceived size and the actual size of the planar surface. The maximum error occuring is approximately 2 cm in the textureless case and 4 cm in the textured case. Apparently, the error is not significantly affected by the distance of the planar surface. Accuracy is better in the tex-

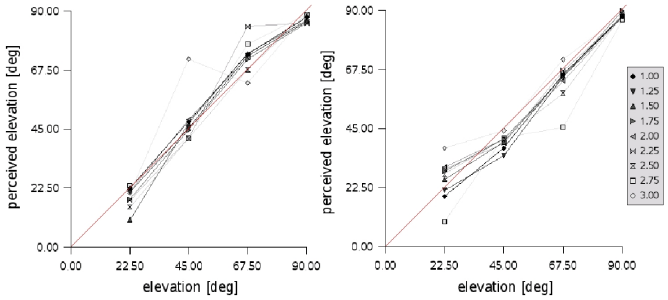

Fig. 7. Perceived elevation of a textureless (left) and textured (right) planar surface, depending on its actual orientation and its distance.

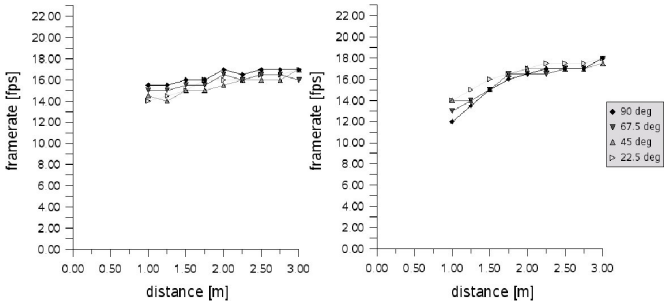

Fig. 8. Overall framerate of our architecture while detecting a textureless (left) or textured (right) planar surface, depending on their distance and orientation.

tureless case, which is because the detection and estimation of textureless planar surfaces involves determining their exact contours in the camera images (see Sec. III-B). In contrast, textured planar surfaces are detected and estimated directly from the 3D maps which, as a consequence of the stereo correlation method used in this experiment, preserve contours only coarsely.

The graphs in Fig. 7 both show a strong agreement between the perceived elevation and the actual elevation of the planar surface. A closer look reveals that the accuracy depends on the distance of the planar surface: Considering the full range of up to 3 m, the average error is about $4.3°$ in the textureless case and $4.9°$ in the textured case. Considering a range of up to 2 m, the average error reduces to approximately $3.7°$ in the textureless case and $3.8°$ in the textured case. At a distance of 1 m, the average error reduces further to approximately $2.9°$ in the textureless case and $3.8°$ in the textured case. Thus, accuracy significantly increases as the planar surface draws closer, which can be achieved by letting the robot approach the planar surfaces it perceives, for example.

The graphs in Fig. 8 show that the framerate of our system enables a robot to interact with the planar surfaces it perceives in real-time: In the textureless case, the framerate ranges from approximately 14 fps to 17 fps, while in the textured case it ranges from approximately 12 fps to 18 fps. Obviously, the framerate depends on the distance of the planar surface: At distances of about 3 m about 17–18 fps are achieved, while at distances of about 1 m the framerate reduces to approximately 12–14 fps. The reason for this is that the projection of the planar surface in the images gets larger as the planar surface gets closer, which implies that increasingly more 3D points are involved in the detection and estimation of the planar surface. The textured case is affected stronger because textured planar surfaces are reconstructed from *dense* 3D points, while textureless planar surfaces are reconstructed from sparse 3D points (see Sec. III-B and Fig. 2).

V. CONCLUSION

We have presented an architecture for detection and estimation of planar surfaces in the scene from calibrated stereo images. Robustness to the unreliable 3D data obtained by local correlation is achieved by Gauss filtering the 3D data and by employing a RANSAC approach to plane fitting. Due to the different sub-systems for textured and textureless planar surfaces, operating either directly on the 3D map or focusing on similarity between pixels in the camera images rather than on their absolute values, planar surfaces of arbitrary visual appearance can be handled. Since coplanarity of 3D points is the only geometric property evaluated by our architecture, assumptions about other geometrical properties such as orientation, shape or presence of straight lines are avoided. By computing 3D representations for each individual frame, arbitrarily moving planar surfaces as well as static ones can be handled. Stable percepts are obtained by transforming these single-frame representations into global coordinates, thus compensating for the ego-motion of the robot, and by establishing spatio-temporal coherence across successive frames. The experiments show that our architecture is able to provide these percepts in real-time while at the same time achieving a degree of accuracy that is sufficient to enable an autonomous robot to interact in a basic manner with the planar surfaces in its environment (see Fig. 9 for an example).

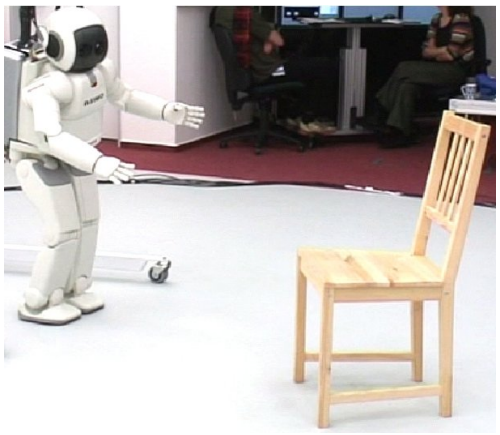

Fig. 9. ASIMO detecting and approaching the planar seat base of a chair.

REFERENCES

[1] J. J. Gibson, *The Ecological Approach to Visual Perception*. Lawrence Erlbaum Associates, 1986.
[2] E. R. Kandel, J. H. Schwartz, and T. M. Jessel, *Principles of Neural Science*. McGraw-Hill, New York, 2000.
[3] M. Z. Brown, D. Burschka, and G. D. Hager, "Advances in computational stereo," *PAMI 25, 8*, 2003.

[4] M. Habbecke and L. Kobbelt, "Iterative multi-view plane fitting," in *VMV*, 2006.
[5] H. Baltzakis and P. Trahanias, "Iterative computation of 3D plane parameters," in *BMVC*, 2000.
[6] J. Zhao, J. Katupitiya, and J. Ward, "Global correlation based ground plane estimation using v-disparity image," in *ICRA*, 2007.
[7] S. Se and M. Brady, "Ground plane estimation, error analysis and applications," *Robotics and Autonomous Systems 39, Elsevier*, 2002.
[8] M. Okutomi, K. Nakano, J. Maruyama, and T. Hara, "Robust estimation of planar regions for visual navigation using sequential stereo images," in *ICRA*, 2002.
[9] M. Asatani, S. Sugimoto, and M. Okutomi, "A footstep-plan-based floor sensing method using stereo images for biped robot control," in *IROS*, 2007.
[10] M. Zucchelli, J. Santos-Victor, and H. I. Christensen, "Multiple plane segmentation using optical flow," in *BMVC*, 2002.
[11] N. Pears and B. Liang, "Ground plane segmentation for mobile robot visual navigation," in *IROS*, 2001.
[12] E. Ribeiro and E. R. Hancock, "Detecting multiple texture planes using local spectral distortion," in *BMVC*, 2000.
[13] K. Okada, S. Kagami, M. Inaba, and H. Inoue, "Plane segment finder: Algorithm, implementation and applications," in *ICRA*, 2001.
[14] K. Okada, T. Ogura, A. Haneda, and M. Inaba, "Autonomous 3d walking system for a humanoid robot based on visual step recognition and 3D foot step planner," in *ICRA*, 2005.
[15] M. I. A. Lourakis, A. A. Argyros, and S. C. Orphanoudakis, "Detecting planes in an uncalibrated image pair," in *BMVC*, 2002.
[16] G. Silveira, E. Malis, and P. Rives, "Real-time robust detection of planar regions in a pair of images," in *IROS*, 2006.
[17] J. Piazzi and D. Prattichizzo, "Plane detection with stereo images," in *ICRA*, 2006.
[18] K. Konolige, "Small vision system: Hardware and implementation," in *8th International Symposium on Robotics Research, Japan*, 1997.
[19] M. Taira, K.-I. Tsutsui, M. Jiang, K. Yara, and H. Sakata, "Parietal neurons represent surface orientation from the gradient of binocular disparity," *Journal of Neurophysiology 83*, 2000.
[20] H. Cantzler, "Random sample consensus (Ransac)." [Online]. Available: http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/CANTZ–LER2/ransac.pdf
[21] J. Schmüdderich, H. Brandl, B. Bolder, M. Heracles, H. Janßen, I. Mikhailova, and C. Goerick, "Organizing multimodal perception for autonomous learning and interactive systems," in *Humanoids*, 2008, submitted.
[22] "Honda worldwide — ASIMO." [Online]. Available: http://world.honda.com/ASIMO/