Segmenting color images into surface patches by exploiting sparse depth data

Babette Dellen Guillem Alenyà Sergi Foix Carme Torras
Institut de Robòtica i Informàtica Industrial (CSIC-UPC), Barcelona, Spain
{bdellen,galenya,sfoix,torras}@iri.upc.edu

Abstract

We present a new method for segmenting color images into their composite surfaces by combining color segmentation with model-based fitting utilizing sparse depth data, acquired using time-of-flight (Swissranger, PMD CamCube) and stereo techniques. The main target of our work is the segmentation of plant structures, i.e., leaves, from color-depth images, and the extraction of color and 3D shape information for automating manipulation tasks. Since segmentation is performed in the dense color space, even sparse, incomplete, or noisy depth information can be used. This kind of data often represents a major challenge for methods operating in the 3D data space directly. To achieve our goal, we construct a three-stage segmentation hierarchy by segmenting the color image with different resolutions - assuming that "true" surface boundaries must appear at some point along the segmentation hierarchy. 3D surfaces are then fitted to the color-segment areas using depth data. Those segments which minimize the fitting error are selected and used to construct a new segmentation. Then, an additional region merging and a growing stage are applied to avoid over-segmentation and label previously unclustered points. Experimental results demonstrate that the method is successful in segmenting a variety of domestic objects and plants into quadratic surfaces. At the end of the procedure, the sparse depth data is completed using the extracted surface models, resulting in dense depth maps. For stereo, the resulting disparity maps are compared with ground truth and the average error is computed.

1. Introduction

The identification and segmentation of 3D surfaces from an image is an important step towards solving robotic object-manipulation tasks as it facilitates object recognition, grasp-point selection, and, in consequence, the execution of appropriate grasping movements. We are interested in developing efficient tools for representing and manipulating domestic objects, including deformable ones. In particular, we aim to segment plants into their composite structures, i.e., leaves or part of leaves, and to extract color and 3D-shape descriptors to find points of interest with which the robot can interact. However, the characterization and classification of surfaces is only possible if appropriate 3D information is available. Various techniques for 3D data acquisition exist, but there is also always a critical trade-off between the accuracy of the method and its efficiency in terms of computation time, applicability (active versus passive methods), and cost. Laser range scanning for example delivers accurate and dense depth information, but has the drawback that it is very time consuming and thus not practical in the context of our task. Photonic mixer devices (PMDs) or Swissranger cameras deliver depth images in real time using the time-of-flight principle, but are of low resolution and afflicted with uncertainties. Stereo vision has the advantage that it is applicable in most environments but tends to fail in untextured areas (due to correspondence problems) and hence often only delivers sparse and noisy results. This problem motivated us to develop a method for surface segmentation that is also applicable to sparse depth information. It further should allow us to complete missing information.

The segmentation of 3D information is the task of dividing the image into regions so that all the points of the same surface belong to the same region. The regions shall not overlap and, taken together, generate the entire image. Many algorithms for solving this task have been proposed in the past [2, 11, 1, 6, 7, 8]. We roughly distinguish between two main groups: region-based and edge-based segmentation algorithms. Algorithms of the first group segment the image into initial regions which are then merged or extended [1, 7, 8]. Since segmentation is applied at a region level, these methods often produce distorted boundaries. Algorithms of the second group find jump boundaries in the depth image, providing an initial segmentation, which is then refined by fitting quadratic functions to the initial segments [11, 6]. A drawback of these methods is that discontinuities in the depth data are hard to detect and may result in an initial under-segmentation of the image, which cannot be corrected at later steps.

Depth acquisition techniques having the advantage of be-

ing fast and economic (stereo, Swissranger, PMD), or non-invasive (stereo), only provide sparse and noisy depth data, constituting a major problem for the segmentation algorithms described above, which have been developed for 3D information acquired with laser range scanners or structured light. In the case of sparse data, information from other sensors, e.g., color, is required to tackle the given task. Walhoff et al. (2007) combined depth information of a PMD camera with color images to segment objects from their background [12]. In a related work by Bleiweiss and Werman (2009), depth from time-of-flight was fused with color data to improve object segmentation and tracking [3]. Real-time foreground segmentation was also achieved by Crabb et al. (2008), combining range and color imaging [4]. Our goal, however, is the segmentation of images into surface patches, which was not approached by these works.

In this paper, we propose to segment dense color images at different resolutions and to select those segments for which a best fit to the sparse depth data can be obtained using quadratic surface models. In some sense, our method is related to edge-based segmenters, with the important difference that we find "edges" in the color space, not in the depth image. The selected segments are then merged and optionally grown using both their respective surface model and color information. The method is based on the assumption that true surface boundaries must emerge at some point along the segmentation hierarchy. It has the advantage that segmentation can be performed in a dense space even though the data of interest are sparse. Under- and over-segmentation are avoided by considering segmentations at various resolutions. Since segmentation in color space usually provides sharp edges, accurate surface boundaries can be obtained in most cases. Furthermore, the obtained (explicit) surface models can be used to interpolate depth data into regions that are initially undefined.

2. Algorithm

We describe an algorithm for the segmentation of color images into surface patches utilizing sparse depth data. The core idea of the method is based on the notion that surface boundaries are in most cases represented by an edge in the color image (the reverse is often not true). Since we are dealing with sparse depth data, it is further desirable to have as large segments as possible - otherwise model fitting becomes impracticable due to lack of data inside segments. We thus segment the color image with different resolutions (see Fig. 1A). Quadratic surface models are fitted to each segment, and we select those segments from the hierarchy which minimize the total fitting error, while taking into account the hierarchy level, i.e., segments obtained at lower resolutions are given preference to segments at higher ones.

The algorithm thus consists of the following steps (see also Fig. 1B). In step 1, the color image and the corresponding sparse depth data is acquired (see Section 4). Then, in step 2, color segmentation is applied to the color image and segmentations at three different resolutions are obtained, creating a segmentation hierarchy as illustrated in Fig. 1A (see also Section 2.1). In step 3 of the algorithm, surface models (quadrics) are fitted to each segment utilizing the sparse depth data, and, for each segment, the best fitting model is selected (see Section 2.2). Then, in step 4, those segments are selected from the hierarchy that produce the total best fit to the data, as illustrated in detail in Fig. 1C (see also Section 2.3). Segments at lower resolutions are given preference to segments at higher resolutions to avoid over-segmentation. The resulting new segmentation is then further improved by applying an additional region merging step (step 5) through which segments having highly different color values can be merged if they describe the same surface (see Section 2.4). Then, in a final step (6), unclustered points are assigned to the closest surface, using both depth and color information (see Section 2.5).

2.1. Hierarchical color segmentation

The color image is segmented using the method of superparamagnetic clustering of data employed in [5] which allows a segmentation hierachy to be generated by segmenting with different resolutions. For this purpose, we varied the interaction strength by multiplying the mean distance $\overline{\triangle}$ by a factor between 1.4 and 0.6. Segments smaller than a threshold (here 10 pixels) are considered being unlabeled and are excluded from the steps 3-5 of the algorithm. The segmentation algorithm can be replaced by any other method as long as different resolutions result in a segmentation hierarchy similar to the one illustrated in Fig. 1A.

2.2. Model fitting and selection

For each color segment s_i and model type (see Section 3) we perform a minimization of the mean square distance

$$E_{i,\text{model}} = 1/N \sum_j (z_j - z_{j,m})^2 \qquad (1)$$

of measured depth points $z_{j,m}$ from the estimated model depth $z_j = f_{i,\text{model}}(x_j, y_j)$, where $f_{i,\text{model}}$ is the data-model function and N is the number of measured depth points in the area of segment s_i. The optimization is performed with a Nelder-Mead simplex search algorithm provided in MATLAB.

The mean square errors for two different model types, i.e. $E_{i,\text{planar}}$ and $E_{i,\text{curved}}$, are computed. We select the planar model if $E_{i,\text{planar}} < E_{i,\text{curved}} + \tau_1$, and the curved model otherwise.

2.3. Color-segment selection procedure

We define the following *selection procedure* considering first only two levels u and $u+1$, where a higher level de-

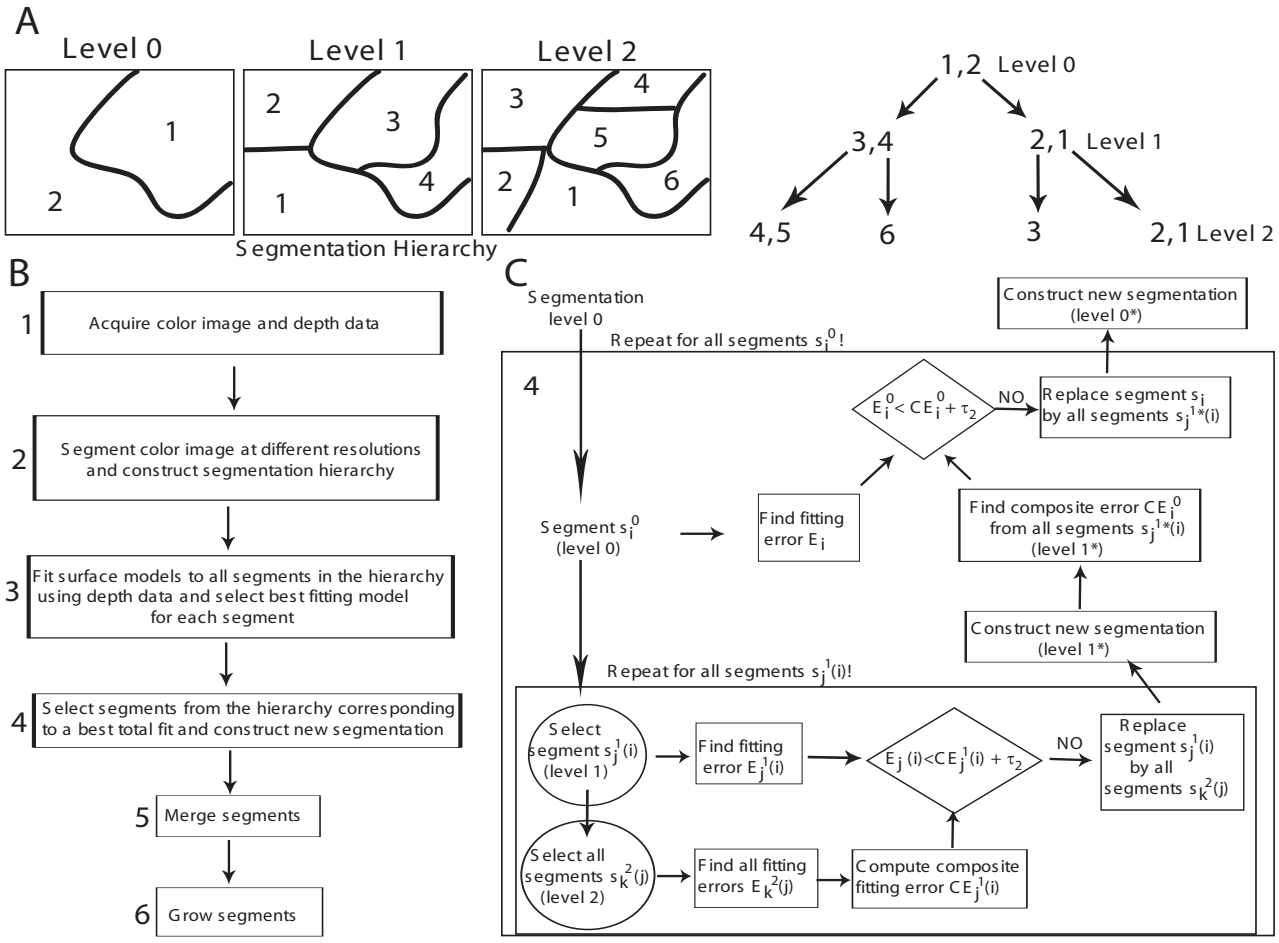

Figure 1. Surface segmentation algorithm. **A** Hierarchical segmentation of an image and resulting segmentation hierarchy. **B** Flow chart of the algorithm. The last step is optional. **C** Detailed schematic of step 4 of the algorithm.

notes a higher resolution. Let s_i^u be a segment at level u of the hierarchy having a fitting error E_i^u (see Section 2.2). At the level $u+1$ of the hierarchy, segment s_i^u is composed of k segments $s_j^{u+1}(i)$ with respective fitting errors E_j^{u+1}. The composite error of the k segments $s_j^{u+1}(i)$ at level u is then defined as

$$CE_i^u = \sum_j a_j^{u+1} E_j^{u+1} / \sum_j a_j^{u+1} \quad , \tag{2}$$

where a_j^{u+1} is the number of valid depth points in the area of segment s_j^{u+1}. We select s_i^u at level u if

$$E_i^u < CE_i^u + \tau_2 \quad , \tag{3}$$

and the k segments $s_j^{u+1}(i)$ from level $u+1$ otherwise. Parameter τ_2 is introduced in order to avoid an over-segmentation of the image by preferring segments obtained at lower resolutions. The procedure is applied to each segment at level u. In this way a new segmentation is constructed which replaces the initial segmentation at u by a segmentation u^*.

Let us now consider a segmentation hierarchy consisting of p levels. We apply the selection procedure to the initial segmentations $u = p-1$ and $u = p$. The selection procedure is applied to the initial segmentation $u = p-2$ and $u^* = p-1$, and so on, until the end of the hierarchy is reached. In this paper, we choose a three-level hierarchy. More levels can be included if desired. The procedure for a three-level hierarchy is illustrated in Fig. 1C.

2.4. Region merging

Let us consider two segments s_i and s_j having fitting errors E_i and E_j. Both segments are merged if $E_{i \cap j} <$

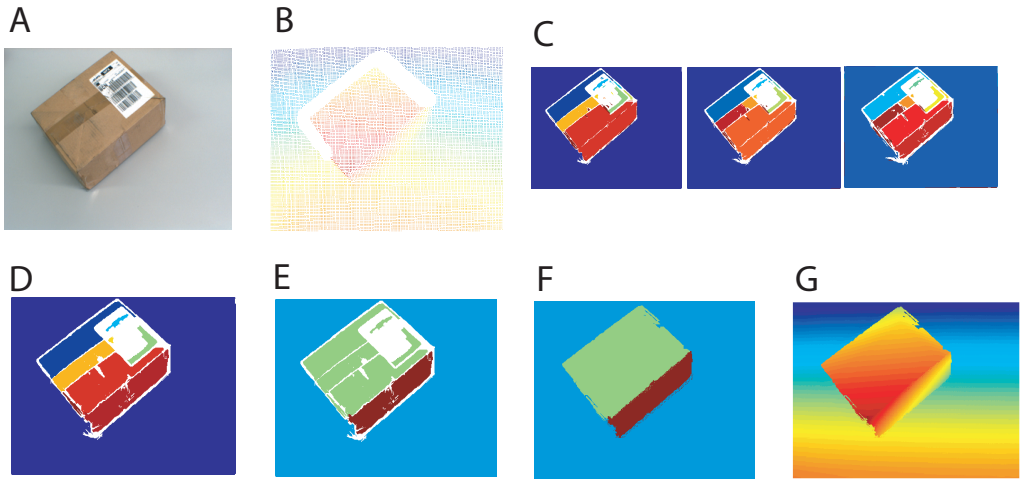

Figure 2. Illustrative example: Segmentation results for a carton box using Swissranger time-of-flight depth. **A** Color image. **B** Sparse depth data, transformed to the color space. **C** Color segmentations at different resolutions (levels 0-2), respectively. **D** Selected segments before merging (after step 4 of the method). **E**. Selected segments after merging (after step 5). **F** Segments after applying region growing (step 6). **G** Fitted depth using surface models. Unclustered points are shown in white.

$(a_i E_i + a_j E_j)/(a_i + a_j) + \tau_2$, where $E_{i \cap j}$ is the fitting error of the merged segments and a_i and a_j is number of valid depth points in the area of the segments s_i and s_j, respectively. This procedure allows segments that have different colors to be merged. The procedure is applied to all segments that are neighbors of each other (i.e. their closest pixels have to be less than 8 pixels apart). When accepting a merge, segments are updated and the new segmentation is used when evaluating the remaining segment pairs.

2.5. Region growing

2.5.1 Time-of-flight

Let p_i be a previously unclustered point with coordinates (x_i, y_i, z_i) and color $\mathbf{c}_i$. We find all segment neighbors of this pixel within a radius of 5 pixels. We compute the distance of p_i to the surface of segment s_j as $\text{dist}_i^j = |z_i - f_j(x_i, y_i)|$, where f_j is the explicit surface-model function of segment s_j, and assign p_i to the closest segment in the neighborhood. For points for which no depth value was originally measured we use the local mean depth value computed over a small area around the point.

2.5.2 Stereo

Let p_i be a previously unclustered point with image coordinates (x_i, y_i). We find all neighboring segments of this pixel within a radius of 10 pixels. We compute the distance of p_i to the surface of segment s_j as $\text{dist}_i^j = |I_{\text{L}}(x_i) - I_{\text{R}}(x_i - f_j(x_i, y_i))|$, where f_j is the explicit surface-model function of segment s_j, and I_{L} and I_{R} are the left and right color images, respectively. According to this distance measure, we assign p_i to the closest segment in the neighborhood.

3. Surface types

We choose two types of surfaces as surface models: planes and a quadratic function, which allows (among others) the modeling of spherical and cylindrical shapes. If desired, higher-order terms can be included. Here we use quadratic functions that allow computing depth z explicitly for the x-y coordinates in the form of $z = f(x, y)$.

3.1. Planes

Planar surfaces are described by three parameters a, b, and c, where the depth z can be expressed as a function of x and y through

$$z = ax + by + c \quad . \tag{4}$$

3.2. Curved surfaces

Curved surfaces are described by five parameters a, b, c, d, and e, where the depth z can be expressed as a function of x and y through

$$z = ax^2 + by^2 + cx + dy + e \quad . \tag{5}$$

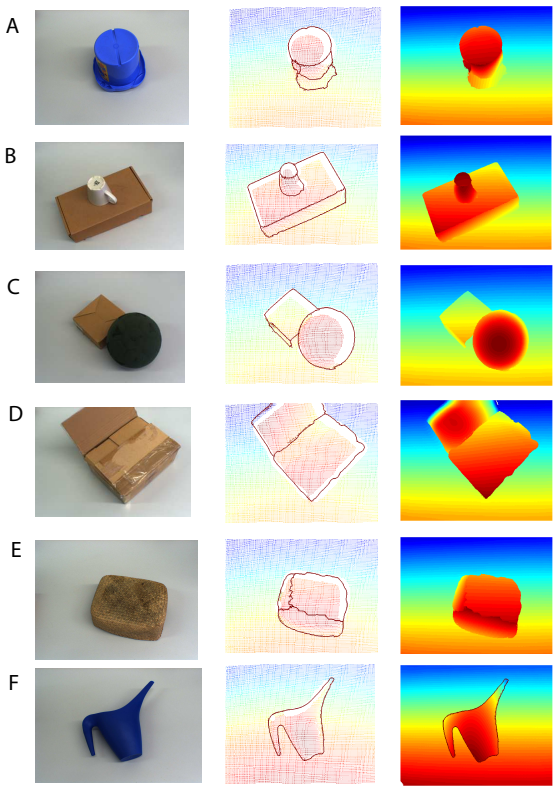

Figure 3. Segmentation results for several domestic objects. Left panels: Original color image. Middle panels: Initial Swissranger sparse depth plotted together with final segment boundaries. Right panels: Fitted depth using segment surface models.

This function has sufficient complexity to allow modelling of basic curved shapes, including spherical, cylindrical, and conical shapes.

4. Data Acquisition

4.1. Time-of-flight depth

We acquire color images together with depth images and combine them by bringing both sensor-pixel matrices in correspondence. For our experiments a Swissranger SR3100 camera and a PMD CamCube camera have been used. The camera is provided with its own illumination system, composed of a set of modulated infra-red LEDs. The SR3100 camera has a low pixel resolution of 176×144, while the PMD camera has a resolution of 204×204, which is however still far below the resolutions of standard RGB cameras. Both cameras have a high frame rate average of 25 fps. This high frame rate makes these cameras suitable for real time applications. Depth and RGB color, provided by a PointGrey Flea camera, can be easily combined for a specific depth range thanks to the precise intrinsic and extrinsic camera calibration. To perform a good data registration reducing partial occlusions, both cameras have to be close to each other (few centimeters) and share the same field of view. By means of using multiple images from both cameras with the same calibration pattern, intrinsic and extrinsic parameters for each camera can be computed. Once this information is achieved, both sensor pixel matrices can be put in correspondence using the calibration pattern as a reference frame and hence registering depth and color information.

Due to the different viewpoints of the two cameras, occlusions can occur in particular for close objects. These occlusions are detected and removed using a buffer approach: The 3D point cloud is transferred to the RGB camera reference space using extrinsic parameters and, if several points are detected along the same line of view, the closest point to the camera is selected. Since the ToF camera has a smaller resolution than the RGB camera, the 3D points are collected from a region around the line of view.

4.2. Stereo disparity

We acquire sparse stereo disparity using a recent algorithm proposed in [5], which is applicable to weakly textured images and thus combines well with color-based segmentation. In this method, stereo segments are found for stereo images and segment silhouettes are computed in both frames. Unique correspondences of silhouette-edge points are searched, and the respective disparities are calculated. Occluded edges are identified and removed from the edge disparity map. Additionally, texture inside segments is exploited by applying a window-based stereo algorithm which operates strictly inside stereo segments. Confidence values are computed and only those disparities that have a high confidence are used, resulting in sparse disparity maps. In our method, we use these sparse disparity maps directly as input without employing the interpolation process described in [5]. We further use a phase-based method to establish initial correspondences [9].

5. Results

The algorithm is tested using both time-of-flight depth data (see Section 4) and stereo disparity. In the case of time-of-flight depth, color images and depth images are acquired together and combined by bringing both sensor-pixel matrices in correspondence using the respective camera-calibration information (see also Section 4). The simulations were performed using MATLAB with an Intel Duo Core Processor T2250 of 1.73GHz. Total run time of the algorithm using non-optimized code for segmenting the time-of-flight data in the carton box example (Fig. 2) is 88 s (including computation times for getting color segmentations). Parameters for segmenting time-of-flight data are

$\tau_1 = 2$ cm^2 and $\tau_2 = 0.25$ cm^2. For stereo disparity, we have $\tau_1 = 0.3$ pixels^2 and $\tau_2 = 0.1$ pixels^2.

5.1. Time-of-Flight

We first illustrate the algorithm on the example of images of a carton box taken with the RGB and the Swissranger depth camera. In Fig. 2A-B, the color image and the corresponding Swissranger depth are presented, respectively. Points for which no depth is computed (due to occlusions and the smaller size of the depth images compared to the color images) are plotted in white. The results of color segmentation at different resolutions (levels 0-2) are displayed in Fig. 2C. Unclustered points are plotted in white color. The results after step 4 of the algorithm are displayed in Fig. 2D. We observe that those color segments corresponding best to the surface structure of the objects got selected. Remaining over-segmentations are removed by applying step 5 of the algorithm. The result is shown in Fig. 2E. Region growing assigns previously unclustered points to the closest surface (see Fig. 2F). The surface models that have been determined can now be used to fit depth to the segment regions, as shown in Fig. 2G. We obtained a total fitting error of 1.1 cm. Fitting errors of the following examples are in a similar range.

Next, we apply the algorithm to color images with Swissranger depth of several domestic objects, i.e., boxes, a cup, a basket, a ball, and a water can, containing planar, spherical, cylindrical shapes, and other curved shapes (Fig. 3A-F). In the left panels, the original color images are shown. In the middle panel, final segment boundaries (after region growing) are presented together with the initial depth data. In the right panels, the final fitted depth is shown. We observe that the algorithm successfully segments most of the prominent surfaces, including curved ones. Only in areas of highly complex structure or insufficient depth information, failures are observed, e.g., the handle of the cup (Fig. 3B) (which was lost during the procedure) and the edge of the cylinder in Fig. 3A. In the example shown in Fig. 3D, we have some problems with false points in the depth map which suggest the upper segment of the box to be curved even though we would expect it to be planar.

In the next step, we apply the method to color-depth images of plants. Here depth is recorded with a PMD camera. Plant images are challenging because they contain many depth layers and occlusions, caused by overlapping leaves, and weak contrast boundaries separating adjacent leaves. The results are presented in Fig. 4A-E. In the left panels, the original color images are shown. In the middle panels, final segment boundaries (after region growing) are presented together with the initial depth data. In the right panels, the final fitted depths are shown together with the final segment boundaries. Even though plants exhibit complicated shapes and have many occlusions, most of the main surfaces have been found, often corresponding to leaves or at least part of leaves, and curved shapes could be modelled correctly in most cases (for example the large leaf at the bottom in Fig. 4A). Basic segment properties such as mean color, size, and mean fitting error are computed, and, based on these criteria, candidate segments (e.g. for robot manipulation) are selected representing leaf structures. An exemplary candidate segment has been marked red for each plant (Fig. 4, left panels). Also the center point of the segments has been marked red.

Scene	E_{fin}	E_{merge}	ρ_{merge}	E_{pb}
Tsukuba	0.67	0.56	81	0.72
Venus	0.34	0.31	83	0.73
Teddy	1.05	0.75	79	3.61
Cones	1.51	1.1	73	2.3
Baby1	1.61	1.18	65	3.78
Lampshade2	2.95	2.67	87	6.79
Plastic	3.31	3.14	87	6

Table 1. Average error in pixels for Middlebury images of final disparity maps E_{fin}, disparities after merging E_{merge} (with densities ρ_{merge}), and of the phase-based approach E_{pb} [9].

5.2. Stereo disparity

Next we apply the algorithm to stereo images from the Middlebury stereo database (URL: vision.middlebury.edu/stereo/) [10]. Three examples are shown in Fig. 5A-C, left panels (Lampshade2, Venus, and Teddy). A sparse disparity map of the scene is computed using a recent algorithm proposed in [5]. We use disparity as input for z. The initial sparse disparity maps are shown together with computed final segment boundaries in Fig. 5A-C, middle panels. Points for which no disparity could be found are shown in white color. The extracted surface models of the segments can then be used to create dense disparity maps, as presented in Fig. 5A-C, right panels. The basic structure of the scene could be captured and surfaces identified. Results of our method for Tsukuba, Venus, Teddy, and Cones have been evaluated using the Middlebury evaluation, summarized in Fig. 6 [10]. We further provide the average disparity error for several Middlebury images in Table 1. These results demonstrate that the method can be used to find dense disparity maps even in cases where little information is given at the beginning, e.g., in case of weakly textured images.

6. Discussion

We proposed a novel algorithm for the segmentation of color images into surface patches using sparse depth data, acquired using either time-of-flight or stereo techniques. The color image is segmented with different resolutions, 3D

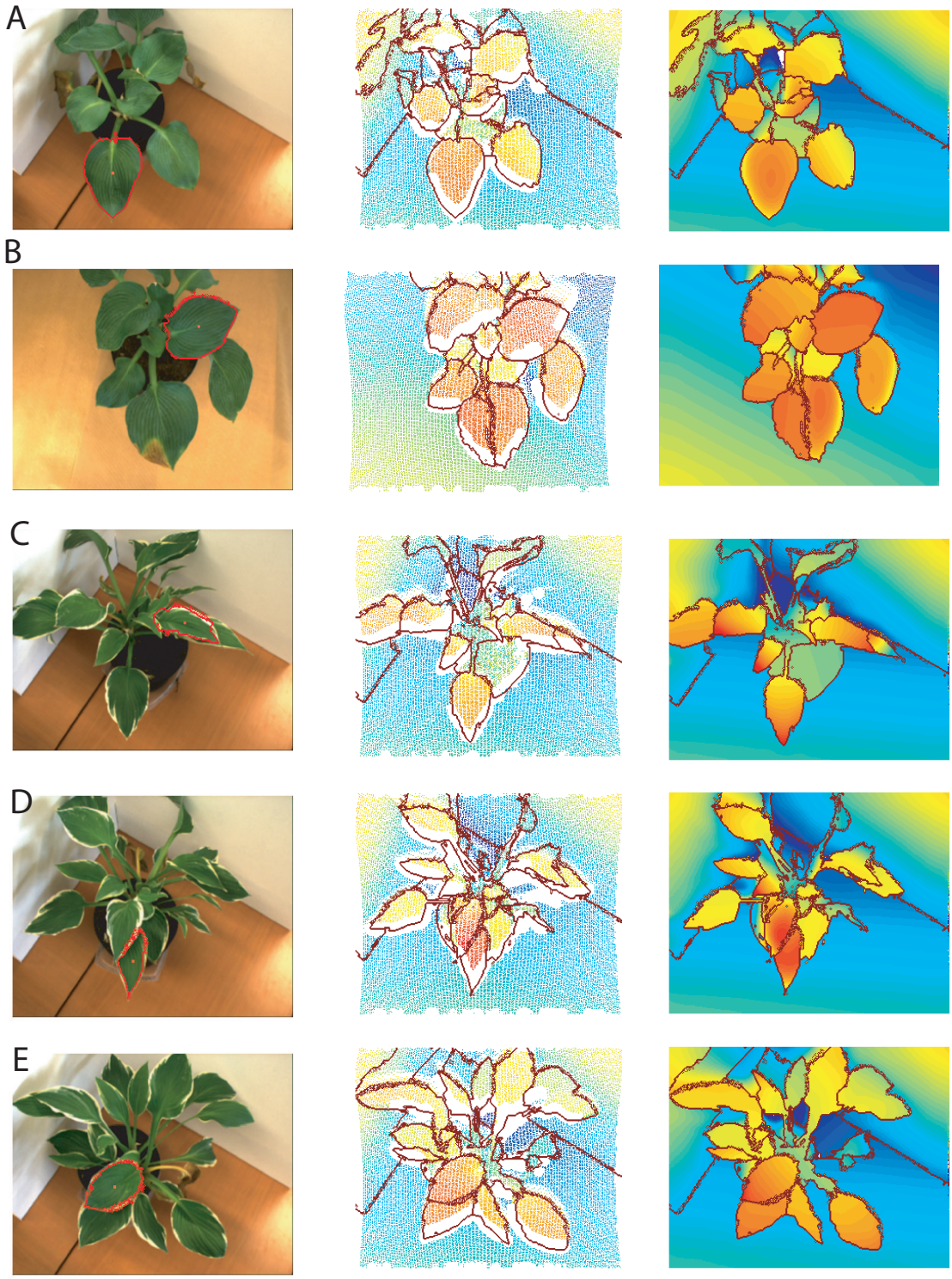

Figure 4. Segmentation results for plants. Left panels: Original color image together with an exemplary candidate segment boundary (marked in red) (see text). Middle panels: Initial PMD sparse depth plotted together with final segment boundaries. Right panels: Fitted depth using segment surface models plotted together with the final segment boundaries.

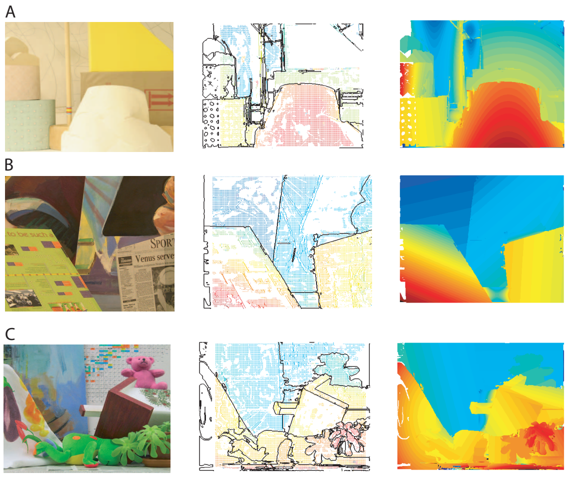

Figure 5. Segmentation results using stereo disparity for **A** Lampshade2, **B** Venus, and **C** Teddy. Left panels: Left color image. Middle panels: Initial disparity with final segment boundaries. Right panels: Fitted disparities using surface models.

GC [1d]	66.5	1.67 51	3.75 66	6.20 50	0.83 61	2.37 74	6.12 62	9.72 82	16.8 85	17.2 74	4.51 70	15.0 80	11.2 67	6.45
FLTG-DDE [90]	68.6	2.07 59	3.98 68	10.7 65	1.60 74	3.00 78	16.1 79	7.35 69	15.3 76	18.8 79	3.37 49	13.2 72	9.52 55	8.75
CSBP [82]	70.7	1.74 54	3.84 67	9.10 58	1.09 67	2.52 75	12.9 75	8.18 77	17.3 82	20.6 82	4.07 65	14.2 78	11.3 68	8.91
BioPsyASW [80]	71.5	3.17 76	4.91 78	12.5 73	2.50 80	3.41 80	15.6 78	8.97 80	14.1 72	16.9 72	3.56 57	11.3 64	8.78 48	8.82
H-Cut [76]	72.1	2.48 68	4.42 72	12.5 73	1.18 71	2.35 73	14.4 77	7.02 64	15.9 78	18.6 78	4.07 64	13.9 76	12.1 71	9.08
Bipartite [78]	74.2	2.24 63	4.02 70	12.0 69	1.92 77	2.61 77	12.6 74	8.49 78	14.0 71	19.9 81	6.68 81	13.5 73	13.8 76	9.31
DP [1b]	76.0	3.43 78	4.23 71	9.85 63	6.50 90	7.43 88	17.4 82	7.11 66	14.9 75	13.4 58	6.52 79	15.1 82	15.1 80	10.1
SAD-IGMCT [52]	76.2	4.61 85	5.80 84	18.9 87	1.99 78	2.58 76	20.3 84	7.92 76	11.2 65	20.9 84	4.25 68	9.85 53	12.8 74	10.1
DPVI [67]	77.6	4.21 83	5.23 81	14.0 78	3.41 83	4.07 82	19.0 83	7.70 75	10.6 60	16.7 69	7.00 82	12.3 69	18.7 86	10.2
YOUR METHOD	79.2	5.59 90	7.15 90	23.4 92	0.89 62	1.74 62	9.53 66	7.40 70	11.3 66	18.9 80	11.0 91	19.8 91	25.0 91	11.8
PhaseBased [31]	80.4	3.05 75	5.28 82	12.8 75	4.94 85	6.17 85	21.4 86	9.18 81	17.9 83	16.8 71	6.64 80	16.3 84	14.8 78	11.3
IMCT [62]	82.0	3.59 79	4.91 77	17.0 84	1.64 75	2.18 70	17.0 81	13.1 88	16.2 79	28.5 91	9.91 89	15.0 81	23.2 90	12.7
SO [1c]	82.4	4.23 84	6.21 85	10.7 64	5.14 86	6.58 86	17.0 80	11.3 84	20.2 86	18.6 77	8.76 86	18.8 89	16.1 82	12.0
RegionalSup [38]	83.5	2.86 72	4.78 75	10.8 66	6.12 87	7.55 89	30.7 91	14.0 90	22.7 91	24.7 86	7.30 83	17.4 88	16.8 84	13.8
STICA [16]	84.1	4.95 88	6.60 87	18.3 86	4.69 84	5.89 84	27.2 88	10.5 83	17.2 81	25.8 88	6.36 78	13.8 75	19.1 87	13.4
SSD+MF [1a]	84.2	4.70 86	6.48 86	21.9 90	3.41 82	4.63 83	10.7 70	12.0 85	20.4 87	24.2 85	7.33 84	16.5 85	20.3 88	12.7
PhaseDiff [23]	86.4	3.19 77	5.38 83	13.2 77	6.22 88	7.42 87	21.1 85	15.2 92	23.3 92	20.8 83	15.6 92	24.2 92	20.6 89	14.7
LCDM+AdaptWgt [75]	87.1	5.04 89	6.80 89	18.9 87	9.87 92	10.7 92	30.7 92	15.1 91	21.6 89	28.1 89	6.18 77	14.3 79	14.9 79	15.2
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]

Figure 6. Middlebury evaluation (main columns from left to right: Tsukuba, Venus, Teddy, and Cones) for an error threshold of 2 pixels (for further explanation of the table, see URL: vision.middlebury.edu/stereo/). The percentage of bad pixels is given for nonoccluded areas, all areas, and discontinuities, in the subcolumns from left to right.

surfaces are fitted to the color-segment areas using sparse depth data, and a new segmentation is found by minimizing the total fitting error while giving preference to segments at lower resolutions. The method has been demonstrated to segment a variety of images of domestic objects and plants into their composite surfaces and shown to be applicable to different kinds of depth information, i.e., time-of-flight and stereo. The method showed robust results for the given parameter set and also was demonstrated to work well for images of plants that contain many depth layers, occlusions, and leaf boundaries of weak contrast. Textured areas could be treated to some extent. Our future goal is to apply the algorithm in robotic settings using color images combined with time-of-flight depth for recognition of domestic objects and plants. The method can also be used to describe objects in a viewpoint-invariant manner through relational graphs of surface segments (nodes), where the edges are surface relations, e.g., relative orientation, as proposed by Fan *et al* (1989) [6].

Acknowledgements

This research is partially funded by the EU GARNICS project FP7-247947, the Consolider-Ingenio project CSD2007-00018, and the Catalan Research Commission under 2009SGR155. G. Alenyà and S. Foix were supported by CSIC under a Jae-Doc and Jae-Pre-Doc fellowship, respectively. B. Dellen acknowledges support from the Spanish Ministry for Science and Innovation via a Ramon y Cajal fellowship.

References

[1] A. Bab Hadiashar and N. Gheissari. Range image segmentation using surface selection criterion. *IP*, 15(7):2006–2018, July 2006. 1

[2] P. J. Besl and R. C. Jain. Segmentation through variable-order surface fitting. *IEEE Trans. Pattern Anal. Mach. Intell.*, 10(2):167–192, 1988. 1

[3] A. Bleiweiss and M. Werman. Fusing time-of-flight depth and color for real-time segmentation and tracking. In *Dyn3D '09: Proceedings of the DAGM 2009 Workshop on Dynamic 3D Imaging*, pages 58–69, 2009. 2

[4] R. Crabb, C. Tracey, A. Puranik, and J. Davis. Real-time foreground segmentation via range and color imaging. In *TOF-CV08*, pages 1–5, 2008. 2

[5] B. Dellen and F. Woergoetter. Disparity from stereo-segment silhouettes of weakly-textured images. In *BMVC09*, 2009. 2, 5, 6

[6] T.-J. Fan, G. Medioni, and R. Nevatia. Recognizing 3-d objects using surface descriptions. *IEEE Trans. Pattern Anal. Mach. Intell.*, 11(11):1140–1157, 1989. 1, 8

[7] O. Faugeras, M. Hebert, and E. Pauchon. Segmentation of range data into planar and quadratic patches. In *CVPR83*, pages 8–13, 1983. 1

[8] B. Parvin and G. Medioni. Segmentation of range images into planar surfaces by split and merge. In *CVPR86*, pages 415–417, 1986. 1

[9] S. Sabatini, G. Gastaldi, F. Solari, J. Diaz, E. Ros, K. Pauwels, M. van Hulle, and N. Krüger. Compact and accurate early vision processing in the harmonic space. In *Int. Conf. on Comp. Vis. Theory and Applications (Visapp)*, 2007. 5, 6

[10] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame correspondence algorithms. *International Journal of Computer Vision*, 47(1-3):7–42, 2002. 6

[11] G. Taylor and L. Kleeman. Robust range data segmentation using geometric primitives for robotic applications. In *Proceedings of the 5th IASTED International Conference on Signal and Image Processing*, pages 467–472, 2003. 1

[12] F. Wallhoff, M. Rub, G. Rigoll, J. Gobel, and H. Diehl. Improved image segmentation using photonic mixer devices. In *ICIP07*, pages VI: 53–56, 2007. 2