J. Vis. Commun. Image R. 25 (2014) 64–73

Contents lists available at SciVerse ScienceDirect

J. Vis. Commun. Image R.

journal homepage: www.elsevier.com/locate/jvci

Learning of perceptual grouping for object segmentation on RGB-D data

Andreas Richtsfeld [*], Thomas Mörwald, Johann Prankl, Michael Zillich, Markus Vincze

Vienna University of Technology, Automation and Control Institute (ACIN), Gusshausstraße 25-29, 1040 Vienna, Austria

ARTICLE INFO

Article history:
Available online 18 April 2013

Keywords:
Computer vision
Object segmentation
Perceptual organization
RGB-D images
B-spline fitting
Object reconstruction
SVM learning
Graph-based segmentation

ABSTRACT

Object segmentation of unknown objects with arbitrary shape in cluttered scenes is an ambitious goal in computer vision and became a great impulse with the introduction of cheap and powerful RGB-D sensors. We introduce a framework for segmenting RGB-D images where data is processed in a hierarchical fashion. After pre-clustering on pixel level parametric surface patches are estimated. Different relations between patch-pairs are calculated, which we derive from perceptual grouping principles, and support vector machine classification is employed to learn Perceptual Grouping. Finally, we show that object hypotheses generation with Graph-Cut finds a globally optimal solution and prevents wrong grouping. Our framework is able to segment objects, even if they are stacked or jumbled in cluttered scenes. We also tackle the problem of segmenting objects when they are partially occluded. The work is evaluated on publicly available object segmentation databases and also compared with state-of-the-art work of object segmentation.

1. Introduction

Wertheimer, Köhler, Koffka and Metzger were the pioneers of studying Gestalt psychology, when they started to investigate this theory about hundred years ago. Wertheimer [1,2] first introduced *Gestalt principles* and Köhler [3], Koffka [4] and Metzger [5] further developed his theories. A summary and more recent contributions can be found in the modern textbook presentation of Palmer [6]. *Gestalt principles* (also called *Gestalt laws*) aim to formulate the regularities according to which the perceptual input is organized into unitary forms, also referred to as wholes, groups, or Gestalten [7]. In visual perception, such forms are the regions of the visual field whose portions are perceived as grouped or joined together, and are thus segregated from the rest of the visual field. These phenomena are called *laws*, but a more accurate term is *principles of perceptual organization*. The principles are much like heuristics, which are mental short-cuts for solving problems. *Perceptual organization* can be defined as the ability to impose structural organization on sensory data, so as to group sensory primitives arising from a common underlying cause [8]. In computer vision this is more often called *perceptual grouping*, when Gestalt principles are used to group visual features together into meaningful parts, unitary forms or objects.

There is no definite list of *Gestalt principles* defined in literature. The first discussed and mainly used ones are *proximity*, *continuity*, *similarity*, *closure* and *symmetry*, defined by Wertheimer [1], Köhler [3], Koffka [4] and Metzger [5]. *Common region* and *element connectedness* were later introduced and discussed by Rock and Palmer [9–11]. Other principles are *common fate*, considering similar motion of elements, *past experience*, considering former experience and *good Gestalt (form)*, explaining that elements tend to be grouped together if they are part of a pattern, which describes the input as simple, orderly, balanced, unified, coherent and as regular as possible. For completeness we also have to mention the concept of *figure-ground articulation*, introduced by Rubin [12]. It describes a fundamental aspect of field organization but is usually not referred to as a *Gestalt principle*, because this term is mostly used for describing rules of the organization of somewhat more complex visual fields. Some of these rules are stronger than others and may be better described as tendencies, especially when principles compete with each other.

Perceptual grouping has a long tradition in computer vision. But many especially of the earlier approaches suffered from susceptibility to scene complexity. Accordingly scenes tended to be “clean” or the methods required an unwieldy number of tunable parameters and heuristics to tackle scene complexity. A classificatory structure for perceptual grouping methods in computer vision was introduced by Sarkar and Boyer [13] in their review of available systems. They listed representative work for each category at this time and updated it later in [8]. More than ten years after Sarkar and Boyer wrote their status on perceptual grouping, cheap and powerful 3D sensors, such as the Microsoft Kinect or Asus Xtion, became available and sparked a renewed interest in 3D

[*] Corresponding author.
E-mail address: ari@acin.tuwien.ac.at (A. Richtsfeld).

1047-3203 © 2013 The Authors. Published by Elsevier Inc. Open access under CC BY-NC-SA license.
http://dx.doi.org/10.1016/j.jvcir.2013.04.006

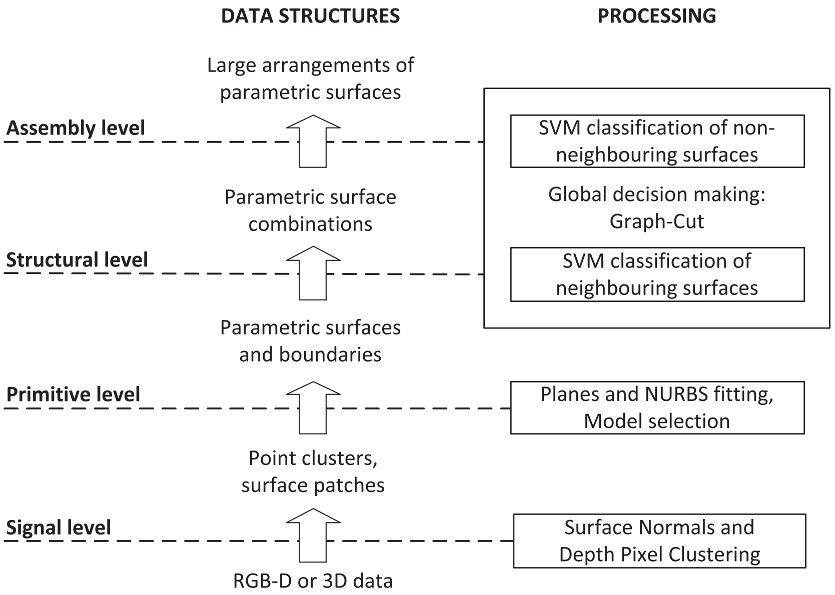
DATA STRUCTURES
PROCESSING
Large arrangements of parametric surfaces
Assembly level
SVM classification of non-neighbouring surfaces
Parametric surface combinations
Global decision making: Graph-Cut
Structural level
SVM classification of neighbouring surfaces
Parametric surfaces and boundaries
Primitive level
Planes and NURBS fitting, Model selection
Point clusters, surface patches
Signal level
Surface Normals and Depth Pixel Clustering
RGB-D or 3D data

Fig. 1. System overview: From raw input data to object hypotheses.

methods throughout all areas of computer vision. Making use of 3D or RGB-D data can greatly simplify the grouping of scene elements, as structural relationships are more readily observable in the data rather than needing to be inferred from a 2D image.

In describing our system we follow the structure of Sarkar and Boyer [13,8], where input data is organized in bottom-up fashion, stratified by layers of abstraction: *signal, primitive, structural* and *assembly level*, see Fig. 1. Raw sensor data, occurring as RGB-D data, is grouped in the signal level to point clusters (surface patches), before the primitive level produces parametric surfaces and associated boundaries. Perceptual grouping principles are learned in the assembly and structural level to form groupings of parametric surface patches. Finally, a globally optimal segmentation is achieved using Graph-Cut on a graph consisting of surface patches and their learned relations.

Signal level – Raw RGB-D images are pre-clustered based on depth information. The relation between 2D image space and the associated depth information of RGB-D data is exploited to group neighboring pixels into patches.

Primitive level – The task on the primitive level is to create parametric surfaces and boundaries from the extracted pixel clusters of the signal level. Plane and B-spline fitting methods are used to estimate parametric surface representations. Model Selection finds the best representation and therefore the simplest set of parametric models for the given data.

Structural level – Features, derived from Gestalt principles, are calculated between neighboring surface patches (in the 3D euclidean space) and a feature vector is created. During a training period, feature vectors and ground truth data are used to train a support vector machine (SVM) classifier to distinguish between patches belonging to the same object and belonging to different objects. The SVM then provides a value for each feature vector from a neighboring patch pair which represents the probability that two neighboring patches belong together to the same object.

Assembly level – Groups of neighboring parametric surfaces are available for processing. Feature vectors are again constructed from relations derived from Gestalt principles, but now between non-neighboring surface patches of different parametric surface groups. A second SVM is trained to classify based on this type of feature vector. Creating object hypotheses directly from the assembly level is difficult, as the estimated probability values from the SVM are only available between single surfaces, but not between whole groupings of surface patches from the structural level. Wrong classifications by the SVMs (which after all only perform a local decision) pose a further problem, possibly leading to high under-segmentation of the scene for only a few errors.

Global Decision Making – To overcome these problems, the decision about the optimal segmentation has to be made on a global level. To this end we build a graph where parametric surfaces from the primitive level represents nodes and the above relations implementing Gestalt principles represent edges. We then employ Graph-Cut using the probability values from the SVM of the assembly level as well as from the structural level as energy terms of the edges to finally segment the most likely connected parts, forming object hypotheses.

The main contribution of our work is the combination of perceptual grouping with SVM learning following a designated hierarchical structure. The learning approach of the framework enables segmentation of unknown objects of reasonably compact shape and allows segmentation for a wide variety of different objects in cluttered scenes, even if objects are partially occluded. Fig. 2 shows segmentation of a complex scene, processed with the proposed framework. Furthermore, the system provides beside image segmentation a parametric model for each object, enabling efficient storage for convenient further processing of the segmented structures.

The paper is structured as follows: The next section discusses representative related work and sets the work in context. Sections 3, 4, 5 and 6 explain the bottom-up processing for each data abstraction level before Section 7 shows global decision making. Experiments and evaluation results are presented in Section 8 and the work ends with a conclusion and a outlook in Section 9.

2. Related work

Many state-of-the-art approaches in literature formulate image segmentation as energy minimization with an MRF [14–17]. Reasoning on raw sensor data without usage of any constraints is a hard and ill-defined problem. So various approaches added constraints using a shape or a location prior, others exploited active segmentation strategies.

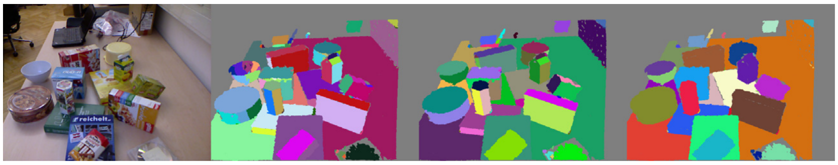

Fig. 2. Original image, pixel clusters, parametric surface patches, segmented scene.

Sala and Dickinson [18,19] perform object segmentation on over-segmented 2D images using a vocabulary of shape models. They construct a graph from the boundaries of region-segments and finds pre-defined simple part models after building a region boundary graph and performing a consistent cycle search. This approach shows interesting results on 2D images, but the system is restricted to a certain vocabulary of 2D-projections from basic 3D shapes. The approach by Hager and Wegbreit [20] is able to segment objects from cluttered scenes in point clouds generated from stereo by using a strong prior 3D model of the scene and explicitly modelling physical constraints such as support. This approach handles dynamic changes such as object appearance/disappearance, but is again limited to predefined parametric models (boxes, cylinders). Silberman and Fergus [21] use superpixels to over-segment RGB-D images. They use a conditional random field (CRF) with a location prior in 2D and 3D to improve segmentation and classification of image regions in indoor scenes.

Kootstra et al. [22–24] use a symmetry detector to initialize object segmentation on pre-segmented superpixels, again using an MRF. Furthermore they developed a quality measure based on Gestalt principles to rank segmentation results for finding the best segmentation hypothesis. Their approach with both, detection and segmentation, was modified by Bergström et al. [25] to overcome the under-segmentation, when objects are stacked or side by side. Bergström formulates an objective function where it is possible to incrementally add constraints generated through human–robot interaction in addition to an appearance model computed from color and texture, which is commonly used to better distinguish foreground from background. Almaddah et al. [26] implement another active vision approach, but without using a MRF. They are able to segment multiple objects, even if they appear side by side. They take advantage of different illumination during active light segmentation. Light with different frequency is projected to the scene, enabling foreground object segmentation and separation of side-by-side objects exploiting different reflectivity of the objects.

Mishra et al. [27,28] show a method to detect and segment compact objects from a scene exploiting border ownership, a concept about knowledge of the object side of a boundary edge pixel. They generate a probabilistic boundary edge map, wherein the intensity of a pixel is the probability to be at the boundary of an object, transfer it to polar space and perform optimal path search to find the best closed contour, representing the object boundary. A drawback of that approach is the lack of object separation in highly cluttered scenes, e.g. when objects are stacked or side by side.

Several approaches perform data abstraction of RGB-D data to form part models before trying to segment objects from the images. Leonardis et al. [29] addressed the problem of fitting higher order surfaces to point clouds. They segmented range images by estimating piecewise linear surfaces, modeled with bivariate polynomials. Furthermore they developed a model selection framework, which is used to find the best interpretation of the range data in terms of Minimum Description Length (MDL). Fisher [30] was a pioneer in perceptual grouping of RGB-D data. He suggests to extract surface patches (pixel clusters) using discontinuity constraints of curvature and depth. Surface clusters are built by linking surface hypotheses based on adjacency and relative surface orientation at the boundaries to reduce the gap between surface patches and object hypotheses. Descriptive features are estimated from boundaries of surfaces, from the surfaces itself and also from surface clusters, enabling object recognition when comparing this features with features of object models from a database. His approach is well structured and theoretically sound, but was more suitable for object recognition rather than for object segmentation.

In our work data abstraction is done, considering the structure of Boyer and Sarkar [13] as well as the suggestions of Fisher [30] and Leonardis [29] by extracting first pixel clusters using discontinuities in the depth image of the sensor level and then estimating parametric surfaces in the primitive level. For the structural and assembly level we propose learning of perceptual grouping principles of these extracted parametric surface patches. Relations between surfaces are derived from perceptual grouping principles and an SVM classifier is trained to distinguish between patches belonging to the same object or different objects. For the generation of object hypotheses we finally employ Graph-Cut to arrive at a globally optimal segmentation for the structural and assembly level.

Compared to our previous work in [31], the perceptual grouping of RGB-D data over different abstraction levels is discussed in detail. In addition, relations based on surface boundaries are introduced to investigate combined 2D edge-based and 3D surface based perceptual grouping, thus improved segmentation results for occluded and concave objects. Furthermore, evaluation and a comparison with another segmentation approach is done.

3. Signal level: pixel clustering

3D cameras, such as Microsoft's Kinect or Asus' Xtion provide RGB-D data, consisting of a color image and the associated depth information for each pixel. From the RGB-D data we compute surface normals and recursively cluster neighboring normals to planar patches. To account for the different noise levels we propose to create an image pyramid and to select clusters of different levels from coarse to fine using a Model Selection criterion. In detail, we create a pyramid by down-sampling the RGB-D data to three levels of detail. Then starting from the two coarsest levels optimized normals are calculated, using the neighborhood reorganization approach by Calderon et al. [32], and recursively clustered to planar surface patches. We then employ Model Selection and a Minimum Description length criterion (MDL) to decide whether a large patch at a coarse level or several smaller patches at a finer level offer a better description of the data. Model Selection is inspired by the framework of Prankl et al. [33] who adapted the work of Leonardis et al. [29] to detect planes from tracked interest points. The idea is that the same data point cannot belong to more than one surface model. Hence an over-complete set of models is generated and the best subset in terms of an MDL criterion is selected. To select the best model, the *savings* S for each surface hypothesis H can be expressed as

$$S_H = S_{data} - \kappa_1 S_m - \kappa_2 S_{err} \quad (1)$$

where S_{data} is the number of data points N explained by the hypothesis H, S_m stands for the cost of coding different models and S_{err} describes the cost for the error incurred by that hypothesis. κ_1 and κ_2 are constants to weight the different terms. As proposed in [29] we use the number of parameters to define S_m (for comparing only planes S_m can simply be set to 1). For the cost S_{err} experiments have shown that the Gaussian error model $\mathcal{N}(\mu_{err}, \sigma_{err}^2)$ and an approximation of the log-likelihood has a superior performance. Hence the cost of the error results in

$$S_{err} = -\log\prod_{i=1}^{N} p(f_i|H) \approx \sum_{i=1}^{N}(1 - p(f_i|H)) \tag{2}$$

and accordingly the substitution of Eq. 2 in Eq. 1 yields the savings of a model

$$S_H = \frac{N}{A_m} - \kappa_1 S_m - \frac{\kappa_2}{A_m}\sum_{i=1}^{N}(1 - p(f_i|H)) \tag{3}$$

where A_m is a normalization value for merging two models. In case

$$S_{l-1} < \sum_{j=1}^{M} S_{j,l} \tag{4}$$

the patch of level $l-1$ is substituted by the overlapping patches $j = 1, \ldots, M$ of level l. This approach allows to detect large surfaces in the noisy background while details in the foreground are preserved. The planar clusters of points are the input for the primitive level where more complex parametric models are estimated.

4. Primitive level: parametric surface model creation

On the primitive level, patches – i.e. pixel clusters – get processed and parametric surface models are created. First, planes and B-spline surfaces are estimated for each cluster, before again Model Selection determines the best explanation for the data in terms of an MDL criterion. Then, greedily two neighboring patches are grouped, B-splines are fitted and again Model Selection is used to decide whether the model of the grouped surface patches or the models of the individual patches better fit to the data.

4.1. B-spline fitting

A common representation of free-form surfaces are B-splines, which are widely used in industry and are the standard for most CAD tools. Due to the definition through the Cox-de-Boor formula they have several beneficial characteristics, like the ability to represent all conic sections, i.e. circles, cylinders, ellipsoids, spheres and so forth. Furthermore refinement through knot insertion allows for representing local irregularities and details, while selecting a certain polynomial degree determines the type of the surface that can be represented.

A good overview of the characteristics and strength of B-splines is summarized in [34]. To reduce the number of parameters to be estimated we set the weights of the control points to 1, that is we actually fit B-Spline surfaces.

The concept of B-Splines would go far beyond the scope of this paper and we therefore want to refer to the well known book by Piegl and Tiller [35]. Instead we want to start from their mathematical definition of B-Spline surfaces in Chapter 3.4.

$$\mathbf{S}(\xi, \eta) = \sum_{i=1}^{u}\sum_{j=1}^{v} N_{i,d}(\xi) M_{j,d}(\eta) \mathbf{B}_{i,j} \tag{5}$$

The basic idea of this formulation is to manipulate the B-spline surface $\mathbf{S} : \mathbb{R}^2 \to \mathbb{R}^3$ of degree d, by changing the entries of the *control grid* $\mathbf{B}$. The i,j-element of the control grid is called *control point* $\mathbf{B}_{i,j} \in \mathbb{R}^3$ which defines the B-spline surface at its region of influence determined by the *basis functions* $N_{i,d}(\xi)$, $M_{j,d}(\eta)$. $(\xi, \eta) \in \Omega$ are called parameters defined on the domain $\Omega \subset \mathbb{R}^2$.

Given a set of points $\mathbf{p}_k \in \mathbb{R}^3$ with $k = 1, \ldots, n$ we want to fit a B-spline surface $\mathbf{S}$ with $u > d$, $v > d$ and $d \geqslant 1$. A commonly used approach is to minimize the squared Euclidean shortest distance e_k from the points to the surface.

$$\begin{aligned} f &= \frac{1}{2}\sum_{k=1}^{n} e_k + w_s f_s \\ e_k &= ||\mathbf{S}(\xi_k, \eta_k) - \mathbf{p}_k||^2 \end{aligned} \tag{6}$$

For regularisation we use the weighted smoothing term $w_s f_s$ to obtain a surface with minimal curvature.

$$f_s = \int_\Omega ||\mathbf{S}''(\xi_k, \eta_k)||^2 d\xi d\eta \tag{7}$$

The weight w_s strongly depends on the input data and its noise level. In our implementation we set $w_s = 0.1$. For minimizing the functional in Eq. (6) the parameters (ξ_k, η_k) are required. We compute them by finding the closest point $\mathbf{S}_k(\xi_k, \eta_k)$ on the B-spline surface to $\mathbf{p}_k$ using Newton's method. The surface is initialised by performing principal-component-analysis (PCA) on the point-cloud (Fig. 3).

4.2. Plane fitting

Although planes are just a special case of B-spline surfaces and could thus be estimated using the above procedure, we chose a more direct approach for this most simple type of surface, because the iterative optimization algorithm in B-spline fitting is computationally more expensive. To this end we use the linear least squares implementation of the Point Cloud Library (PCL) [36].

4.3. Model selection

Algorithm 1. Modelling of surface patches

```
Detect piecewise planar surface patches
for $i = 0 \rightarrow$number of patches do
  Fit B-splines to patch $i$
  Compute MDL savings $S_{i,B-spline}$ and $S_{i,plane}$
  if $S_{i,B-spline} > S_{i,plane}$ then
    Substitute the model $H_{i,plane}$ with $H_{i,B-spline}$
  end if
end for
Create Euclidean neighborhood pairs $P_{ij}$ for surface patches
for $k = 0 \rightarrow$number of neighbors $P_{ij}$ do
  Greedily fit B-splines to neighboring patches $P_{ij}$
  Compute MDL savings $S_{ij}$ to merged patches
  if $S_{ij} > S_i + S_j$ then
    Substitute individual models $H_i$ and $H_j$ with merged B-spline model $H_{ij}$
  end if
end for
```

To optimally represent the data with a minimal set of parameter we again use the Model Selection framework introduced for data abstraction in Section 3. First, we represent the point clusters with planes and B-spline surfaces depending on the savings computed with Eq. (3). To account for the complexity of the surface model now S_m is set to the number of parameters of the models, i.e., three times the number of B-spline control points. Then the savings of neighboring patches S_i and S_j are compared to the savings of a model fitted to a merged patch S_{ij} and in case

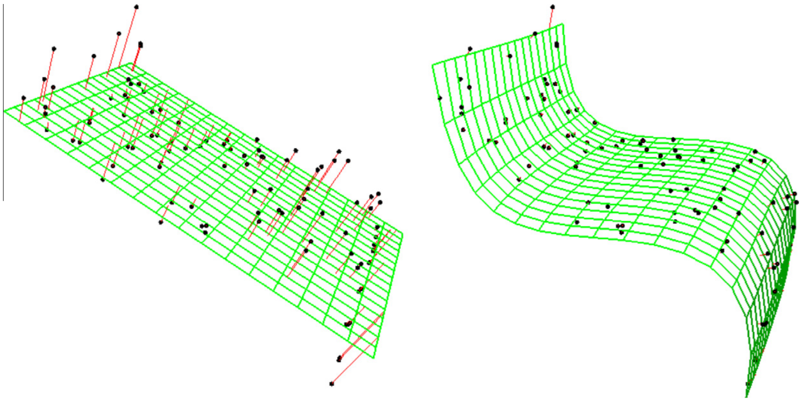

Fig. 3. Left: Initialisation of a B-Spline surface (green) using PCA. Right: The surface is fitted to the point-cloud (black) by minimizing the closest point distances (red) ($m = n = 3,\ p = 2,\ w_a = 1,\ w_r = 0.1$). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

$$S_{ij} > S_i + S_j \tag{8}$$

the individual patches are substituted with the merged patch. Algorithm 1 summarizes the proposed surface modelling pipeline.

5. Structural level: grouping of parametric surfaces

After the first two levels parametric surfaces and their boundaries are available for further processing in the structural level. A crucial task on this level is to find relations between surface patches, indicating that they belong to the same object and to define them in a way that relations are valid for a wide variety of different objects. Based on the Gestalt principles discussed earlier, we introduce the following relations between neighboring surface patches:

- r_{co} ⋯similarity of patch color,
- r_{rs} ⋯relative patch size similarity,
- r_{tr} ⋯similarity of patch texture quantity,
- r_{ga} ⋯gabor filter match,
- r_{fo} ⋯fourier filter match,
- r_{co3} ⋯color similarity on 3D patch borders,
- r_{cu3} ⋯mean curvature on 3D patch borders,
- r_{cv3} ⋯curvature variance on 3D patch borders,
- r_{di2} ⋯mean depth on 2D patch borders,
- r_{vd2} ⋯depth variance on 2D patch borders.

The first relations are inferred from the *similarity principle*, which can be integrated in many different ways. Similarity of patch color r_{co} is implemented by comparing the 3D-histogram in the YUV color space. The histogram is constructed of four bins in each direction leading to 64 bins in the three-dimensional array. The Fidelity distance a.k.a. Bhattacharyya coefficient ($d_{Fid} = \sum_i \sqrt{P_i * Q_i}$) is then calculated to get a single color similarity value between two different surface patches. Similarity of patch size r_{rs} is again based on the *similarity principle* and is calculated as the relation between two patch sizes.

Texture similarity is realized in three different ways: As difference of texture quantity r_{tr}, as Gabor filter match r_{ga} and as Fourier filter match r_{fo}. Texture quantity is calculated as relation of canny edge pixels to all pixels of a surface patch. The difference of texture quantity is then the difference of those values of two surface patches. The Gabor and Fourier filter are implemented as proposed in [37]. For the Gabor filter six different directions (in 30° steps) with five different kernel sizes $(17, 21, 25, 31, 37)$ are used. A feature vector $\mathbf{g}$ with 60 values is built from the mean and the standard deviation of each filter value. The Gabor filter match r_{ga} is then the minimum difference between these two vectors ($d(\mathbf{g}_1, \mathbf{g}_2) = \min_{k=0,\dots,5,} \sum_{i=1}^{60} \sqrt{(\mu_{1,i} - \mu_{2,i+10k})^2 + (\sigma_{1,i} - \sigma_{2,i+10k})^2}$), when one feature vector gets shifted such that different orientations of the Gabor filter values are matched. This guarantees a certain level of rotation invariance for the filter. The Fourier filter match r_{fo} is calculated as Fidelity distance of five histograms, each consisting of 8 bins filled with the normalized absolute values of the first five coefficients from the discrete Fourier transform (DFT).

Subsequent relations are local feature values along the common border of two patches using the 2D and 3D relationship of neighboring patches. Color similarity r_{co3}, the mean r_{cu3} and the variance of curvature r_{cv3} are calculated along the 3D patch border of surface patches and the mean r_{di2} and the variance of depth r_{vd2} are calculated along borders in the 2D image space. While r_{co3} represents again a relation inferred from *similarity*, r_{cu3} and r_{cv3} representing relations inferred from a mixture of *continuity* as well as *closure*, which could also be interpreted as compactness in the 3D space. The mean of depth r_{di2} and the variance of the depth r_{vd2} along 2D patch borders in the image space describe relations inferred from the *proximity* and the *continuity principle*.

We then define a feature vector, containing all relations between neighboring patches:

$$\mathbf{r_{st}} = \{r_{co}, r_{rs}, r_{tr}, r_{ga}, r_{fo}, r_{co3}, r_{cu3}, r_{cv3}, r_{di2}, r_{vd2}\} \tag{9}$$

Feature vectors r_{st} are calculated between all combinations of neighboring parametric surfaces in the 3D image space. These vectors are then classified as indicating patches belonging to the same object or different objects using a support vector machine (SVM). Feature vectors r_{st} with hand-annotated ground truth segmentation from a set of RGB-D images are used to train the SVM during an offline phase. Feature vectors of patch pairs from the same object represent positive training examples and vectors of pairs from different objects or objects and background represent negative examples. With this strategy, not only the affiliation of patches to the same object, but also the disparity of object patches to other objects or background is learned.

For the offline training and online testing phase we use the freely available *libsvm package* [38]. After training the SVM is not only capable to provide a binary decision *same* or *notsame* for each feature vector $\mathbf{r}$, but also a probability value $p(same|\mathbf{r})$ for each decision, based on the theory introduced by Wu and Lin [39]. As solver C-support vector classification (C-SVC) with $C = 1,\ \gamma = 1/n$ and $n = 9$ is used and as kernel the radial basis function (RBF):

$$\mathbf{K}(x_i, x_j) = e^{\gamma ||x_i - x_j||^2} \tag{10}$$

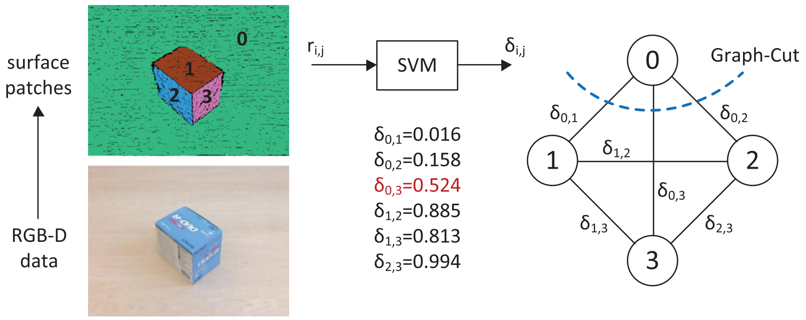

Fig. 4. Simple Graph-Cut example: Input image (bottom left image) is abstracted to parametric surface patches (top left image); The SVMs of the structural and assembly level estimate probability values $\delta_{i,j}$ from the feature vectors $r_{i,j}$ and a graph is constructed (right); Graph-Cut estimates the globally optimal segmentation of objects, correcting single wrong classification ($\delta_{0,3}$).

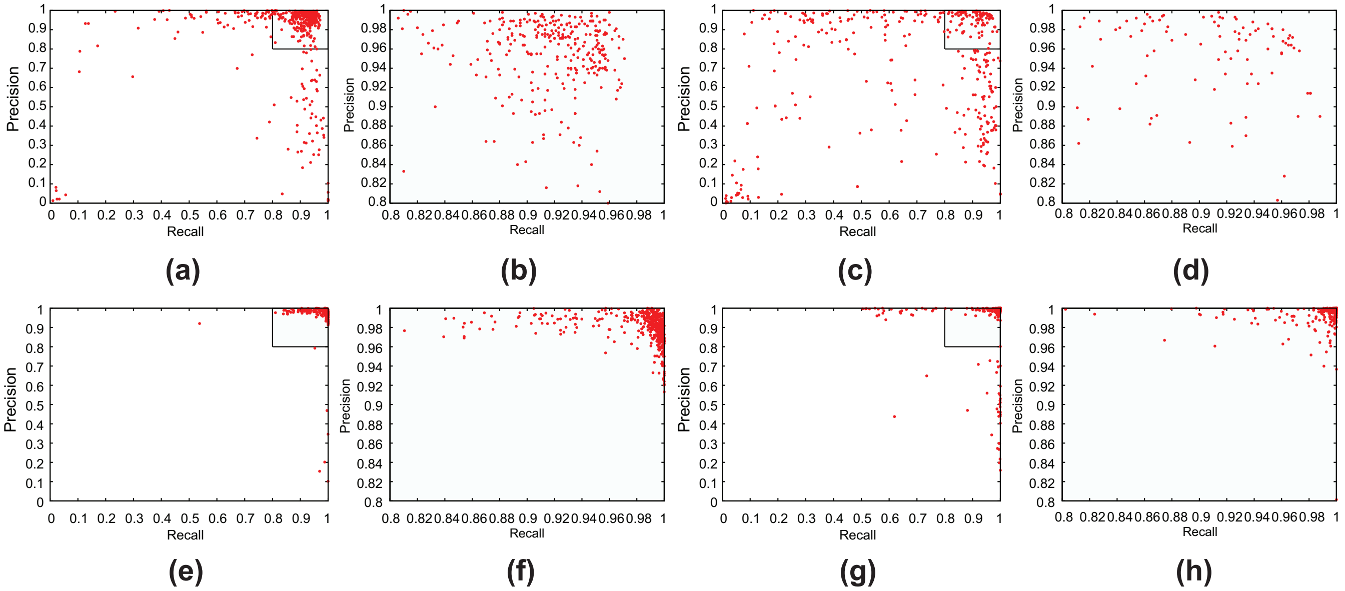

Fig. 5. Precision-Recall for each segmented object. (a)–(d) with Mishra's [28] approach, (e)–(h) with our approach. Plot a and e are showing results from the Willow Garage dataset, (b) and (f) more detailed the upper right corner of (a) and (e). Plot c and g are showing results from the OSD database, (d) and (h) more detailed the upper right corner of (c) and (g).

6. Assembly level: grouping of parametric surface groups

Using the above relations and probabilities we could now already form groups of neighboring surface patches by applying a threshold (e.g. $p(same\,|\,\mathbf{r}) = 0.5$). This would however fail to correctly segment partially occluded objects as the separated object parts would be regarded as independent objects. The assembly level is the last level of grouping and is responsible to group spatially separated surface groupings. Similar to the structural level relations between patches are introduced, derived from the already discussed Gestalt principles:

- r_{co} ⋯similarity of patch color,
- r_{rs} ⋯relative patch size similarity,
- r_{tr} ⋯similarity of patch texture quantity,
- r_{ga} ⋯gabor filter match,
- r_{fo} ⋯fourier filter match,
- r_{md} ⋯minimum distance between patches,

Table 1
The learn- and test-set split in six sub-categories. Columns presenting the numbers of images, objects, relations in the structural level and relations in the assembly level.

	Learn set				Test set			
	Nr.	Obj.	r_{st}	r_{as}	Nr.	Obj.	r_{st}	r_{as}
Boxes	17	38	157	48	16	36	193	290
Stacked boxes	8	20	101	58	8	21	121	315
Occluded obj.	8	16	73	124	7	14	58	159
Cylindric obj.	12	38	104	250	12	42	83	419
Mixed obj.					12	81	412	1913
Complex scene					11	162	754	10149
Total	45	108	435	480	66	356	1621	13240

Table 2
Results on the OSD database [42] for the structural level.

	F_{score}	BER_{svm} (%)	P (%)	R (%)	P^* (%)	R^* (%)
$r_{st} = \{r_{co}\}$	0.124	39.5	16.05	93.9	92.30	95.47
$r_{st} = \{r_{rs}\}$	0.197	41.2	47.24	94.9	93.46	95.49
$r_{st} = \{r_{tr}\}$	0.603	43.7	32.62	93.9	89.42	95.50
$r_{st} = \{r_{ga}\}$	0.379	41.3	17.96	94.1	93.75	95.49
$r_{st} = \{r_{fo}\}$	0.200	44.0	19.07	95.0	93.75	95.48
$r_{st} = \{r_{co3}\}$	0.266	38.6	41.15	88.8	93.74	95.48
$r_{st} = \{r_{cu3}\}$	1.703	21.1	92.21	95.9	66.85	94.58
$r_{st} = \{r_{cv3}\}$	0.053	43.7	13.80	90.9	93.75	95.50
$r_{st} = \{r_{di2}\}$	0.506	30.3	23.77	95.7	92.92	95.74
$r_{st} = \{r_{vd2}\}$	0.737	28.9	23.08	95.4	94.48	95.50
r_{st}		14.0	93.75	95.48		

Table 3
Results on the OSD database [42] for the assembly level.

	F_{score}	BER_{svm} (%)	P (%)	R (%)	$P*$ (%)	$R*$ (%)
$r_{st}, r_{as} = \{r_{co}\}$	47.4e−3	50.0	93.75	95.48	87.16	96.49
$r_{st}, r_{as} = \{r_{rs}\}$	13.8e−3	50.0	93.75	95.48	86.89	96.52
$r_{st}, r_{as} = \{r_{tr}\}$	20.4e−3	50.0	93.75	95.48	90.19	96.58
$r_{st}, r_{as} = \{r_{ga}\}$	25.3e−3	50.0	93.75	95.48	83.82	97.00
$r_{st}, r_{as} = \{r_{fo}\}$	17.9e−3	50.0	93.75	95.48	90.93	96.87
$r_{st}, r_{as} = \{r_{md}\}$	38.4e−3	50.0	93.75	95.48	92.33	96.46
$r_{st}, r_{as} = \{r_{nm}\}$	34.7e−3	50.0	93.75	95.48	91.08	96.21
$r_{st}, r_{as} = \{r_{nv}\}$	3.98e−3	50.0	93.75	95.48	86.19	97.21
$r_{st}, r_{as} = \{r_{ac}\}$	18.5e−3	50.0	93.75	95.48	93.78	96.13
$r_{st}, r_{as} = \{r_{dn}\}$	27.3e−3	50.0	93.75	95.48	83.52	96.42
$r_{st}, r_{as} = \{r_{cs}\}$	25.7e−3	50.0	93.75	95.48	89.83	96.81
$r_{st}, r_{as} = \{r_{od}\}$	8.35e−3	50.0	93.75	95.48	83.98	96.76
$r_{st}, r_{as} = \{r_{ls}\}$	7.01e−3	50.0	93.75	95.48	83.49	96.85
$r_{st}, r_{as} = \{r_{as}\}$	7.27e−3	50.0	93.75	95.48	82.76	96.60
$r_{st}, r_{as} = \{r_{lg}\}$	0.12e−3	50.0	93.75	95.48	59.97	69.97
$r_{st} + r_{as}$		41.1	89.98	97.05		

Table 4
Precision and recall on the OSD and Willow Garage dataset with our approach using only the SVM of the structural level SVM_{st} or using both SVM_{st+as} and results by Mishra et al. [28].

	SVM_{st} (%)		SVM_{st+as} (%)		Mishra et al. [28] (%)	
	P	R	P	R	P	R
Boxes	99.85	98.11	99.85	98.85	71.46	78.28
Stacked boxes	93.26	98.61	93.34	99.93	64.37	76.91
Occluded obj.	99.84	83.38	99.80	95.57	76.69	56.59
Cylindric obj.	99.63	97.79	99.64	98.26	66.13	84.67
Mixed obj.	94.04	98.08	88.46	98.28	62.19	74.12
Complex scene	89.12	92.99	82.72	94.34	55.61	62.5
OSD total	93.75	95.48	89.98	97.05	62.14	70.93
Willow Garage	92.70	97.12	90.59	97.12	87.03	86.01

- r_{nm} ⋯angle between mean surface normals,
- r_{nv} ⋯difference of variance of surface normals,
- r_{ac} ⋯mean angle of normals of nearest contour p.,
- r_{dn} ⋯mean distance in normal direction of nearest contour points,
- r_{cs} ⋯collinearity continuity,
- r_{oc} ⋯mean collinearity occlusion distance,
- r_{ls} ⋯closure line support,
- r_{as} ⋯closure area support,
- r_{lg} ⋯closure lines to gap relation.

The first five relations are equal to the relations used in the structural level and characterize again the *similarity* between patches. The implementation details were already discussed, see Section 5.

The minimum distance between patches r_{md} is inferred from the *proximity principle*. In the structural level this was given implicitly as only neighboring surface patches were considered. Now it is explicitly considered as relation between non-neighboring patches. r_{nm} and r_{nv} are the difference of the mean and variance of the normals of a surface patch and roughly represent shape *similarity* between two patches. For the last two relations the nearest twenty percent of contour (boundary) points of two patches are calculated. r_{ac} and r_{dn} compare the mean angle between the surface normals of the boundary points and the mean distance in normal direction of the boundary points. These principles are inferred from the *continuity principle*.

The last five relations of the assembly level are created from the boundary of the surfaces. We are using the framework introduced in [40,41] to fit lines to the boundary of segments in the 2D image space. With the concept of search lines, intersections between line segments from different boundaries are found and can be categorized as L-Junctions or Collinearities. For relation r_{cs} are collinearities in the 2D image space estimated. Collinearity continuation r_{cs} is then calculated as the sum of the angle between the two lines and the distance in normal direction from one line-endpoint to the other line, both calculated in the 3D image space. The feature with the lowest value is chosen, if more than one collinearity between two surface patches is found. Due to the processing of non-neighboring surface patches in the assembly level is a gap between the end-points of collinearities and hence a hypothesized line can be calculated in between. Relation r_{oc} measures the mean distance between this hypothesized line and the points of the surface (s) in between and measures therefore a possible occlusion. The rest of the boundary relations are based on closures (closed convex contours), found with a shortest path search, when considering the L-junctions and collinearities as connections between lines. Three different relations are calculated, r_{ls} describing the line support, r_{as} the area support and r_{lg} the relation between line length and gap length. Again is a representative feature vector defined from the relations:

$$\mathbf{r_{as}} = \{r_{co}, r_{rs}, r_{tr}, r_{ga}, r_{fo}, r_{md}, r_{nm}, r_{nv}, r_{ac}, r_{dn}, r_{cs}, r_{oc}, r_{ls}, r_{as}, r_{lg}\} \qquad (11)$$

The feature vector describes the relation between non-neighboring surface patches from different surface patch groupings of the structural level. Similar to the structural level, the feature vector is used to train an SVM for classification, to provide again after a training period a probability value $p(same|\mathbf{r})$ for connectedness, but now for two non-neighboring surface patches.

Depending on the number of surface patches in the groupings, there are several probability values between two groupings of the structural level and optimal object hypotheses cannot be created by simple thresholding these values. Instead, we try to find a globally optimal solution by building a graph and performing Graph-Cut segmentation.

7. Global decision making: graph cut segmentation

After SVM classification in the structural and assembly level some probability estimates may contradict when trying to form object hypotheses. A globally optimal solution has to be found to overcome vague or wrong local predictions from the two SVMs at the structural and assembly level. To this end we define a graph, where surface patches represent nodes and edges are represented by the probability values of the two SVMs. A simple example is shown in Fig. 4. We employ graph-cut segmentation on the graph, introduced by Felzenszwalb and Huttenlocher [17], using the probability values as the pairwise energy terms to find a global optimum for object segmentation.

8. Experiments and results

A database for evaluation was created, consisting of table top scenes organized in several learn- and test-sets with various types of objects and with different complexities of the scenes. Ground truth data for object segmentation is available for all learn- and test-sets. The *Object Segmentation Database (OSD)* is published at [42], an overview of the content and the number of images and objects within are shown in Table 1. Furthermore, the number of extracted relations from the structural and assembly level (r_{st}, r_{as}) are stated for each learn- and test-set.

Evaluation of the relations is done by calculation of the F-score for each relation. F-score is a technique which measures the discrimination of two sets of real numbers, usually used for feature selection, see Chen et.al [43]. Given training vectors $r_k,\ k = 1, \ldots, m$, if the number of positive and negative instances

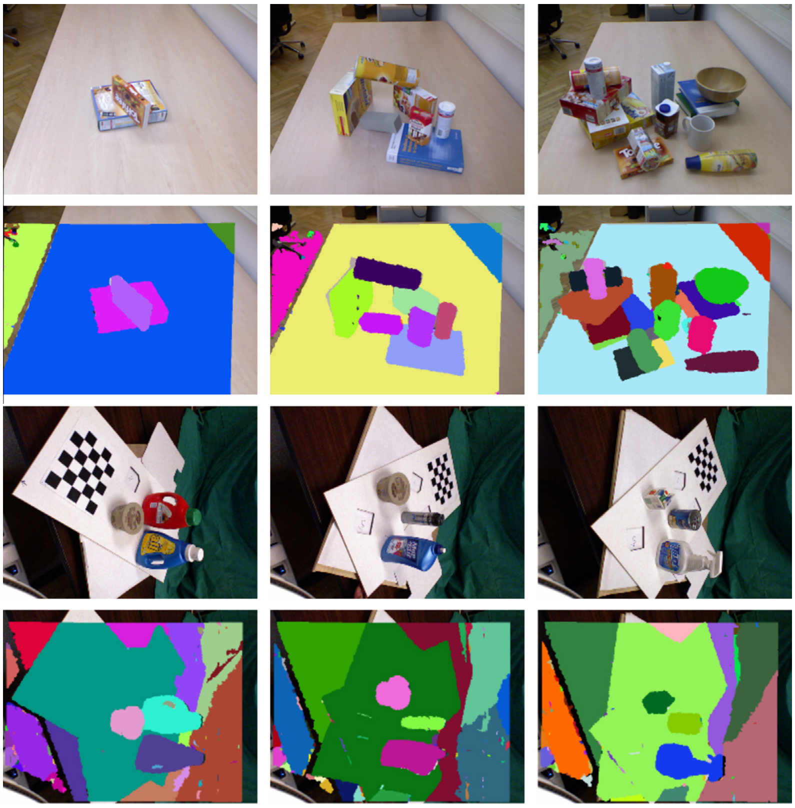

Fig. 6. Example segmentation from the OSD (first two rows) and from the Willow Garage dataset (second two rows), both learned on the OSD learn-set.

are n_+ and n_-, respectively, then the F-score of the ith feature is defined as:

$$F(i) = \frac{(\bar{r}_i^{(+)} - \bar{r}_i)^2 + (\bar{r}_i^{(-)} - \bar{r}_i)^2}{\frac{1}{n_+-1}\sum_{k=1}^{n_+}(r_{k,i}^{(+)} - \bar{r}_i^{(+)})^2 + \frac{1}{n_--1}\sum_{k=1}^{n_-}(r_{k,i}^{(-)} - \bar{r}_i^{(-)})^2} \qquad (12)$$

where $\bar{r}_i, \bar{r}_i^{(+)}$ and $\bar{r}_i^{(-)}$ are the average of the ith feature of the whole, positive and negative data sets, respectively, $r_{k,i}^{(+)}$ is the ith feature of the kth positive instance and $r_{k,i}^{(-)}$ is the ith feature of the kth negative instance. The numerator indicates the discrimination between the positive and negative sets, and the denominator indicates the one within each of the two sets. The larger the F-score, the more likely this feature is more discriminative, but unfortunately it does not reveal mutual information among features.

Table 2 shows evaluation results, first of individual relations and in the last row of Table 2 from the feature vector with all relations of the structural level. This reveals the importance of each introduced relation for further processing in the system. The first column of the table shows the F-score for each relation and the second column presents the balanced error rate BER_{svm} of the results from the SVM, which is computed from the *true positive* t_p, *true negative* t_n, *false positive* f_p and *false negative* f_n decisions:

$$BER_{svm} = \frac{1}{2} * \left(\frac{fp}{tp+fp} + \frac{fn}{tn+fn}\right) \qquad (13)$$

It can be seen that a higher F-score leads to fewer wrong decisions of the SVM, resulting in a lower BER. The following columns show *PrecisionP* and *RecallR* of segmentation summed up over the whole testset and finally P^* and R^* show Precision and Recall when using the whole feature vector r_{st} without considering the given relation. A decrease here shows the importance of each relation to the overall performance of the segmentation framework.

Table 3 shows the same evaluation as shown in Table 2, but now for relations of the assembly level (when using them additionally to the structural level). A comparison of Tables 2 and 3 shows higher F-scores for relations of feature vector r_{st}, indicating a higher discrimination compared to the relations of feature vector r_{as}. Therefore, neighboring patches can be easier connected correctly than non-neighboring patches what shows also the strength of the *proximity principle* which is implicitly implemented due to the splitting of the framework structure into structural and assembly level.

It is noticeable that a single relation never leads to a decision that two non-neighboring patches belong together, because the low prior probability (0.0123) of positive decisions always results in negative decisions of SVM_{as}. This is shown by the 50.0% for the balanced error rate BER of the SVM_{as} and also by the values of precision and recall. When using more than one relation for r_{as}, the SVM_{as} decides also sometimes positive and starts connecting non-neighboring patches, resulting finally in the overall results shown in the last row.

The usage of the assembly level leads to better results of recall R, because partially occluded and non-compact object shapes may now be segmented correctly, but the chance of sometimes wrongly connecting surface patches increases and leads to lower precision P. The decision of using the assembly level is left to the user who decides which error is more important for a certain application.

The evaluation of the relations from the structural level shows that relations based on the *similarity principle* are more relevant to connect patches, specifically r_{co}, r_{tr} and r_{fo}. Other relations are more relevant to separate patches, e.g. r_{cu3} and r_{di2}, which are inferred mainly from the *continuity* and *closure principle*.

Table 4 shows a comparison of our approach with state-of-the-art segmentation method by Mishra et al. [28]. For all experiments in Table 4 we trained our system with the learning sets of the OSD. The columns show again *PrecisionP* and *RecallR*, first when using our framework up to the structural level (SVM_{st}), then for the whole framework including the assembly level (SVM_{st+as}) and finally for the approach by Mishra. In addition both methods have been evaluated on the *Willow Garage database*[1] for which we provide the created ground truth data at [42]. Examples from the Willow Garage database will be shown in Fig. 6 in the last section. Evaluation on the Willow Garage dataset shows the generalization of our approach with respect to other objects and scenes during training, because our framework was trained with the OSD learning sets.

The results of Table 4 show that our approach works significantly better than the approach by Mishra for all test sets of the OSD. For the occluded object set recall is much higher when using the assembly level, while precision remains constant on a high level. Precision decreases when scenes are becoming more complex, because the assembly level accidentally connects more non-neighboring surface patches of different objects and therefore produces more errors.

For the Willow Garage dataset recall remains stable for our approach when using the assembly level in addition, because objects are not stacked, occluded or jumbled. Mishra's method performs for that reason also better on the Willow Garage dataset than on the OSD database. Fig. 5 shows finally precision over recall for each segmented object in the OSD and also for each object in the Willow Garage dataset. Compared to Mishra's method there are only a few objects where both, precision *and* recall are worse. This means that objects are mainly over – *or* under-segmented when they are not segmented correctly.

9. Discussion and conclusion

We presented a framework for segmenting unknown objects in cluttered table top scenes in RGBD-images. Raw input data is abstracted in a hierarchical framework by clustering pixels to surface patches in the primitive level. Parametric surface models are estimated from the surface clusters, represented as planes and B-spline surfaces and Model Selection finds the combination which explains the input data best. In the structural and assembly level relations between neighboring and non-neighboring surface patches are estimated which we infer from Gestalt principles. Instead of matching geometric object models, more general perceptual grouping rules are learned with a SVM. With this approach we address the problem of segmenting objects when they are stacked, side by side or partially occluded, as shown in Fig. 6.

The presented object segmentation approach works well for many scenes with stacked or jumbled objects, but there are still open issues which are not yet handled or could be revised. A major limitation of our approach is the inability of the grouping approach to split wrong pre-segmented surface patches. If objects are stacked or side-by-side and surface parts of different objects are aligned to one co-planar plane, pre-segmentation will wrongly detect one planar patch and the following grouping approach is not able to split it again. Considering color as additional cue during pre-segmentation would be one solution to solve this issue. Another, rather obvious limitation of our approach is the resolution of the sensor, causing errors when small objects or object parts cannot be abstracted to surfaces.

The current implementations of relations delivers better segmentation results for convex objects compared to concave objects due to the fact that concave objects may have self-occlusion which leads to splitting and non-neighboring surface patches of the same object. Usually cylindrical objects, such as mugs and bowls show this nicely when the inner and outer part is decomposed into separate parts. Therefore, concave objects have to be treated similar to occluded objects, but evaluation results have shown that relations of the assembly level are far weaker what causes more errors for this types of objects. Another reason why results at the assembly level are weaker is the noise on depth boundaries of the sensor what causes wrong relation values for the relations based on boundary edges of surfaces. Reducing the noise by depth image enhancement and further investigation of relations based on boundaries could increase the quality of results of the assembly level.

However, the presented grouping framework demonstrates that learning of generic perceptual grouping rules is a method which enables object segmentation of unknown objects when data is initially abstracted to meaningful parts. The examples shown in Fig. 6 demonstrate that the knowledge about the learned rules can be transferred to other object shapes. The turned camera position shows that no prior assumptions about the camera pose is needed. Evaluation of the proposed framework has shown that the approach is promising due to the expandability of the relations in the framework. The proposed method has the ability for usage in several indoor robotic tasks where identifying unknown objects or grasping plays a role.

Acknowledgments

The research leading to these results has received funding from the European Community's Seventh Framework Programme [FP7/2007-2013] under Grant agreement No. 215181 (CogX) and by the Austrian Science Fund (FWF) under Grant agreement No. I513-N23 (Vision@home) and No. TRP 139-N23 (InSitu).

References

[1] M. Wertheimer, Untersuchungen zur Lehre von der Gestalt. II, Psychological Research 4 (1) (1923) 301–350.
[2] M. Wertheimer, Principles of perceptual organization, in: D.C. Beardslee, M. Wertheimer (Eds.), A Source Book of Gestalt Psychology, Van Nostrand, Inc., 1958, pp. 115–135.
[3] W. Köhler, Gestalt psychology today, American Psychologist 14 (12) (1959) 727–734.
[4] K. Koffka, Principles of Gestalt Psychology, International Library of Psychology, Philosophy, and Scientific Method, vol. 20, Harcourt, Brace and, World, 1935.
[5] W. Metzger, Laws of Seeing, first ed., The MIT Press, 1936.
[6] S. Palmer, Photons to Phenomenology, A Bradford Book, 1999.
[7] D. Todorovic, Gestalt principles, Scholarpedia 3 (12) (2008) 5345.
[8] K.L. Boyer, S. Sarkar, Perceptual organization in computer vision: status, challenges, and potential, Computer Vision and Image Understanding 76 (1) (1999) 1–5.
[9] I. Rock, S. Palmer, The legacy of Gestalt psychology, Scientific American 263 (6) (1990) 84–90.
[10] S.E. Palmer, Common region: a new principle of perceptual grouping, Cognitive Psychology 24 (3) (1992) 436–447.
[11] S. Palmer, I. Rock, Rethinking perceptual organization: the role of uniform connectedness, Psychonomic Bulletin & Review 1 (1) (1994) 29–55.
[12] E. Rubin, Visuell Wahrgenommene Figuren, Copenhagen Gyldendals.
[13] S. Sarkar, K.L. Boyer, Perceptual organization in computer vision – a review and a proposal for a classificatory structure, IEEE Transactions On Systems Man and Cybernetics 23 (2) (1993) 382–399.
[14] S. Vicente, V. Kolmogorov, C. Rother, Joint optimization of segmentation and appearance models, in: 2009 IEEE 12th International Conference on Computer Vision, IEEE Computer Society, 2009, pp. 755–762. No. ICCV.
[15] Y.Y. Boykov, M.-P. Jolly, Interactive graph cuts for optimal boundary and region segmentation of objects in N-D images, International Conference on Computer Vision (ICCV), vol. 1, 2001, pp. 105–112.
[16] C. Rother, V. Kolmogorov, A. Blake, GrabCut: interactive foreground extraction using iterated graph cuts, ACM Transactions on Graphics (SIGGRAPH) 23 (3) (2004) 309–314.
[17] P.F. Felzenszwalb, D.P. Huttenlocher, Efficient graph-based image segmentation, International Journal of Computer Vision 59 (2) (2004) 167–181.
[18] P. Sala, S.J. Dickinson, Model-based perceptual grouping and shape abstraction, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, 2008, pp. 1–8.

[1] http://vault.willowgarage.com/wgdata1/vol1/solutions_in_perception/Willow_Final_Test_Set/.

[19] P. Sala, S. Dickinson, Contour grouping and abstraction using simple part models, in: European Conference on Computer Vision (ECCV), Lecture Notes in Computer Science, vol. 6315, 2010, pp. 603–616.

[20] G.D. Hager, B. Wegbreit, Scene parsing using a prior world model, The International Journal of Robotics Research 30 (12) (2011) 1477–1507.

[21] N. Silberman, R. Fergus, Indoor scene segmentation using a structured light sensor, in: 2011 IEEE International Conference on Computer Vision Workshops ICCV Workshops, Dept. of Computer Science, Courant Institute, New York University, USA, IEEE, 2011, pp. 601–608.

[22] G. Kootstra, N. Bergström, D. Kragic, Gestalt principles for attention and segmentation in natural and artificial vision systems, in: Semantic Perception, Mapping and Exploration (SPME), ICRA 2011 Workshop, Shanghai, 2011, pp. 1–8.

[23] G. Kootstra, N. Bergström, D. Kragic, Fast and automatic detection and segmentation of unknown objects, in: Proceedings of the IEEE-RAS International Conference on Humanoids Robotics (Humanoids), Bled, 2010, pp. 442–447.

[24] G. Kootstra, D. Kragic, Fast and bottom-up object detection, segmentation, and evaluation using Gestalt principles, in: International Conference on Robotics and Automation (ICRA), 2011, pp. 3423–3428.

[25] N. Bergström, M. Björkman , D. Kragic, Generating object hypotheses in natural scenes through human–robot interaction, in: Intelligent Robots and Systems (IROS), 2011, pp. 827–833.

[26] A. Almaddah, Y. Mae, K. Ohara, T. Takubo, T. Arai, Visual and physical segmentation of novel objects, in: 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems, Osaka University, Japan, 2011, pp. 807–812.

[27] A.K. Mishra, Y. Aloimonos, Visual segmentation of simple objects for robots, Robotics: Science and Systems VII (2011) 1–8.

[28] A.K. Mishra, A. Shrivastava, Y. Aloimonos, Segmenting simple objects using RGB-D, in: International Conference on Robotics and Automation (ICRA), 2012, pp. 4406–4413.

[29] A. Leonardis, A. Gupta, R. Bajcsy, Segmentation of range images as the search for geometric parametric models, International Journal of Computer Vision 14 (3) (1995) 253–277.

[30] R.B. Fisher, From Surfaces to Objects: Computer Vision and Three Dimensional Scene Analysis, vol. 7, John Wiley and Sons, 1989.

[31] A. Richtsfeld, T. Mörwald, J. Prankl, M. Zillich, M. Vincze, Segmentation of unknown objects in indoor environments, in: 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2012.

[32] F. Calderon, U. Ruiz, M. Rivera, Surface-normal estimation with neighborhood reorganization for 3D reconstruction, in: Proceedings of the Congress on Pattern Recognition 12th Iberoamerican Conference on Progress in Pattern Recognition, Image Analysis and Applications, CIARP'07, Springer-Verlag, Berlin, Heidelberg, 2007, pp. 321–330.

[33] J. Prankl, M. Zillich, B. Leibe, M. Vincze, Incremental model selection for detection and tracking of planar surfaces, in: Proceedings of the British Machine Vision Conference, BMVA Press, 2010, pp. 87.1–87.12.

[34] J.A. Cottrell, T.J.R. Hughes, Y. Bazilevs, Isogeometric analysis, Continuum 199 (5–8) (2010) 355.

[35] L. Piegl, W. Tiller, The NURBS Book, Computer-Aided Design 28 (8) (1997) 665–666.

[36] R. Rusu, S. Cousins, 3D is here: point cloud library (pcl), in: 2011 IEEE International Conference on Robotics and Automation (ICRA), IEEE, 2011, pp. 1–4.

[37] A.A. Ursani, K. Kpalma, J. Ronsin, Texture features based on Fourier transform and Gabor filters: an empirical comparison, in: 2007 International Conference on Machine Vision, 2007, pp. 67–72.

[38] C.-c. Chang, C.-j. Lin, LIBSVM: a library for support vector machines, ACM Transactions on Intelligent Systems and Technology 2 (3) (2011) 27:1–27:27.

[39] T. Wu, C. Lin, Probability estimates for multi-class classification by pairwise coupling, The Journal of Machine Learning Research 5 (2004) 975–1005.

[40] M. Zillich, Incremental Indexing for Parameter-Free Perceptual Grouping, in: 31st Workshop of the Austrian Association for Pattern Recognition, 2007, pp. 25–32.

[41] A. Richtsfeld, M. Vincze, 3D shape detection for mobile robot learning, in: Torsten Kröger, Friedrich M. Wahl (Eds.), Advances in Robotics Research, Springer Berlin Heidelberg, Braunschweig, 2009, pp. 99–109.

[42] A. Richtsfeld, The Object Segmentation Database (OSD), 2012. <http://www.acin.tuwien.ac.at/?id=289>.

[43] Y.-w. Chen, C.-j. Lin, Combining SVMs with various feature selection strategies, in: Feature Extraction, Studies in Fuzziness and Soft Computing, vol. 324, Springer, 2006, pp. 315–324. Ch. 12.