Depth-Adaptive Superpixels

David Weikersdorfer, David Gossow, Michael Beetz
Intelligent Autonomous Systems Group, Technische Universität München.
[weikersd—gossow—beetz]@cs.tum.edu

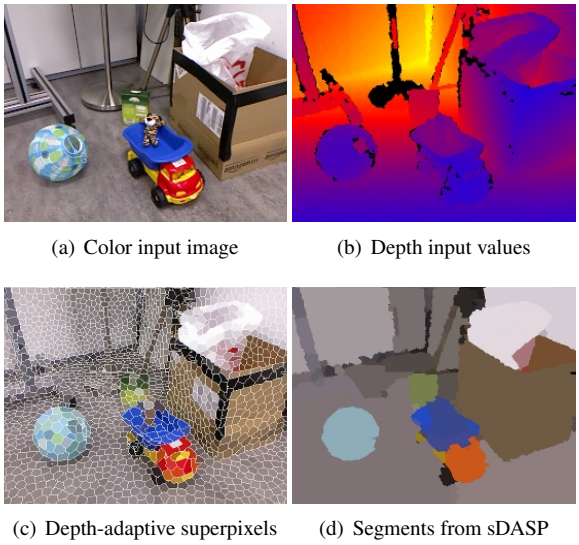
Abstract

We propose a novel oversegmentation technique for RGB-D images. The visible surface of the 3D geometry is partitioned into uniformly distributed and equally sized planar patches. This results in a classic oversegmentation of pixels into depth-adaptive superpixels which correctly reflect deformation through perspective projection. The advantages of depth-adaptive superpixels (DASP) are demonstrated by using spectral graph theory to create image segmentations in near realtime. Our algorithms outperform state-of-the-art oversegmentation and image segmentation algorithms both in quality and runtime.

1 Introduction

Following [7], image segmentation divides an image into regions which fulfill two criteria: intra-region similarity and inter-region dissimilarity. Oversegmentation into superpixels focuses on intra-region similarity, possibly dividing an image into more segments than necessary. An oversegmentation greatly reduces scene complexity and can be used as the basis of advanced and expensive algorithms. Recent oversegmentation algorithms like Turbopixels [5] and SLIC [1] additionally have the property that the image is covered uniformly with superpixels of similar size. Turbopixels use a geometric-flow-based algorithm which in addition assures connectivity and compactness.

Since the advent of the Microsoft Kinect device [9], RGB-D sensors have become available for a wide range of applications. In this paper we contribute two algorithms, oversegmentation (DASP) and segmentation (sDASP), that make use of the additional depth information in order to simplify the segmentation task (see figure 1). In section 2 and 3, we describe an oversegmentation algorithm, which partitions the visible scene geometry into uniformly distributed near-planar surface patches. Due to its efficiency, we keep the notion of



(a) Color input image (b) Depth input values

(c) Depth-adaptive superpixels (d) Segments from sDASP

Figure 1. Superpixels and segments

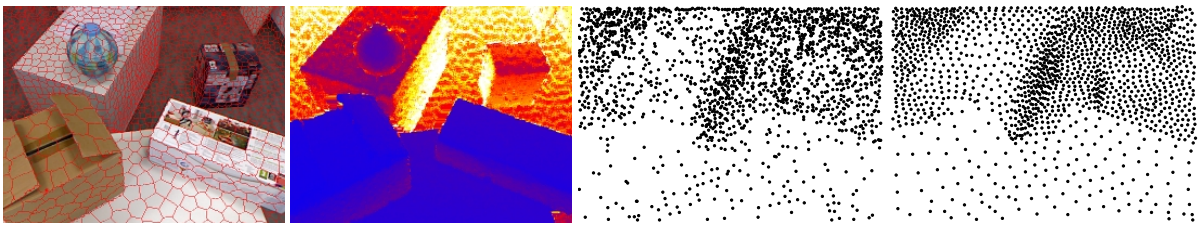
a two-dimensional image and instead deform the shape of local neighborhoods in image space according to the perspective distortion and scaling. In section 4, we show how spectral graph theory [8] can be used on top of depth-adaptive superpixels to generate a full segmentation of the image. Our method is general and can be used to compute segmentations from any oversegmentation. Our evaluation in section 5 shows that both methods achieve significantly better performance than state-of-the art algorithms which only consider pixel color.

2 Depth-Adaptive Superpixels

Given depth values $p_\zeta := \zeta(p_j, p_i) \in \mathbb{R}^+$ for every pixel (p_j, p_i), corresponding 3D points p_v in camera coordinates are computed with the pinhole camera model:

$$p_v = p_\zeta \left(\frac{p_j - c_x}{f}, \frac{p_i - c_y}{f}, 1 \right)^T, \qquad (1)$$

where f and (c_x, c_y) are camera parameters.



(a) Image with DASP borders (b) Superpixel density (c) Sampled cluster seeds (d) Centers after 20 iterations

Figure 2. Color image, cluster density (left) and cluster seeds and centers (right)

Using knowledge of the scene geometry surface, the principle of superpixels can be transferred to 3D space. We aim to compute an oversegmentation of the surface into uniformly distributed, planar patches with a fixed radius R. The parameter R corresponds to the minimal size of interesting features and has to be chosen depending on the application domain.

The visible part of the surface can be parametrized by the pixel grid. Thus, a 3D oversegmentation of surface points is dual to a 2D oversegmentation of pixels. When projecting surface patches onto the image plane, they are distorted by perspective projection, corresponding to a non-uniform oversegmentation into depth-adaptive superpixels in the image space. To guarantee equal size of surface patches, the density of superpixels has to increase with distance from the camera and inclination of the surface normal relative to view direction. To assure that surface patches are distributed uniformly, i.e. under a Poisson disk distribution, samples drawn from the density distribution need to have a spectrum with blue noise characteristics [3].

Desirable segment borders include color, texture and geometry edges. While image-based approaches have to rely on the fact that often, geometric edges go along with changes in color and texture, the use of the 3d point and normal corresponding to each pixel makes it possible to respect geometry directly. In our case, it is sufficient to estimate the normal p_n using the depth gradient $\nabla\zeta(p_j, p_i)$ computed from finite differences:

$$\nabla\zeta(p_j, p_i) = \frac{1}{2d_w}\begin{pmatrix}\zeta(p_j+d_p,p_i)-\zeta(p_j-d_p,p_i)\\ \zeta(p_j,p_i+d_p)-\zeta(p_j,p_i-d_p)\end{pmatrix} \quad (2)$$

We choose $d_w = \frac{R}{2}$ and compute d_p as the size of d_w projected into the image plane (see Eq. 3).

3 The DASP algorithm

We proceed in three steps. First, the density of superpixel clusters in the image space is computed from the depth image. Second, we use an efficient method to sample points which will guarantee the blue-noise spectrum property. Third, a clustering algorithm assigns points to superpixels and improves superpixel centers.

The density of superpixel seeds at pixel (p_j, p_i) can be computed by considering a disk with radius R whose center point is at depth $\zeta(p_j, p_i)$ and projected into pixel (p_j, p_i) on the image sensor [4]. The projected radius of this disk is computed using the trivial equation

$$r_p(p_j, p_i) = \frac{f}{\zeta(p_j, p_i)} R\,. \quad (3)$$

For surfaces parts that are not parallel to the image plane, one has to compute the perspective distortion. We locally approximate this with an affine deformation by considering the local depth gradient $\nabla\zeta(p_j, p_i)$. This gives the projected area of the disk as

$$A_p(p_j, p_i) = \frac{r_p(p_j, p_i)^2\,\pi}{\sqrt{\left\|\nabla\zeta(p_j, p_i)\right\|^2 + 1}}\,. \quad (4)$$

The density of superpixels is directly computed as $\rho(p_j, p_i) \propto \frac{1}{A_p(p_j,p_i)}$. Figure 2 shows superpixel density and sampled seeds for an example image. The bumpy pattern results from noisy depth measurements.

Fattal [3] describes an efficient method to draw blue-noise point samples using a multi-scale sampling scheme. Initially, few points are distributed using the lowest density frequency and shifted into an optimal configuration using the Langevin method. This process is repeated iteratively for higher frequencies by using the point configuration of the previous step, splitting points if necessary and optimizing again. In order to achieve realtime performance, we use the basic idea of multi-scale sampling, but do not perform the Langevin step. During the pixel clustering step, superpixel centers are automatically shifted into a cluster configuration which approximately satisfies the blue noise spectrum property in addition to flatness and color constraints (see figure 2).

Similar to SLIC, pixels are assigned to superpixel clusters using an iterative k-means algorithm with a customized metric. As superpixel clusters only change locally, the search radius during a k-means iteration can

(a) T=1 (b) T=3 (c) T=50 (d) Turbo (e) DASP with 1000 superpixels (f) sDASP contours (g) sDASP segment labels

Figure 3. Far Left: DASP superpixels after a varying number of iterations T (a) - (c) are compared against Turbopixels (d). Figures (e) - (g) show sDASP segmentation results.

be reduced to a window around the cluster center. This results in a runtime linear in the number of pixels and is likewise used in [1] and [3]. We consider a feature space $\mathcal{F} \subset \mathbb{R}^9$ where a pixel is represented by a feature vector consisting of the point position p_v, the pixel color p_c and the point normal p_n. A metric d_F on feature vectors $\mathbf{f} = (p_v, p_c, p_n)$ and $\mathbf{f}' = (p'_v, p'_c, p'_n)$ is defined by a linear combination of metrics defined on the components.

$$d_F(\mathbf{f}, \mathbf{f}') := \sum_{\circ=v,c,n} \mu_\circ \, d_\circ(p_\circ, p'_\circ) \tag{5}$$

The Euclidean distance of points is normalized with respect to the disk radius R: $d_v(p_v, p'_v) := \frac{1}{R}\|p_v - p'_v\|$. Color distance d_c is measured by the Euclidean distance in the CIELAB color space. d_n measures the angle between normals: $d_n(p_n, p'_n) := 1 - p_n \circ p'_n$. As weights we used $\mu_v = 1$, $\mu_c = 2$ and $\mu_n = 3$ for all results.

4 Image segmentation with DASP

Superpixels provide a sparse image representation, making them suitable as a preprocessing stage in various computer vision algorithms. We demonstrate this for the case of image segmentation. Spectral graph theory forms the basis of the normalized cuts algorithm [8], which extracts global shape information using local pixel affinity. This idea is transferred to superpixels by analyzing the superpixel neighborhood graph and considering superpixel affinity.

Let N the number of superpixels and B_{kl} the set of pixels on the border between two superpixels $1 \leq k, l \leq N$, represented by their mean feature vectors $\bar{\mathbf{f}}_k$ and $\bar{\mathbf{f}}_l$. The superpixel affinity matrix $W \in \mathbb{R}^{N \times N}$ is constructed by using a metric similar to d_F:

$$W_{kl} := \begin{cases} e^{-d'_F(\bar{\mathbf{f}}_k, \bar{\mathbf{f}}_l)}, & B_{kl} \neq \emptyset \\ 0, & B_{kl} = \emptyset \end{cases} . \tag{6}$$

To avoid strong disconnections at geometry edges we use $d'_v := \min(d_v, 5)$. In image segmentation, convex geometry edges usually do not indicate a segment border. Thus $d'_n := d_n$ if two superpixel surface patches form a concave surface and 0 otherwise. d'_c is d_c scaled with the constant $\sqrt{N}/100$. Using the diagonal matrix $D_{kk} := \sum_{l=1}^{N} W_{kl}$, we solve the sparse general eigenvalue problem

$$(D - W)\,\mathbf{x} = \lambda\, D\,\mathbf{x}\,. \tag{7}$$

We proceed in a similar way as the contour detector sPb from [2] to extract borders from eigenvectors v_t and eigenvalues λ_t. However, we only need to consider pixels on borders between superpixels, which greatly simplifies the process. The border intensity image is computed as

$$I(i,j) := \sum_{t=1}^{C} \frac{1}{\sqrt{\lambda_t}} |v_{tk} - v_{tl}|, \; (i,j) \in B_{kl}\,. \tag{8}$$

Segment contours are extracted by using a threshold on I. Segment labels are received by computing connected components on the superpixel graph reduced to edges below the threshold (see figure 3). This method can be applied to any oversegmentation algorithm like Turbopixels or SLIC (named sSLIC in figure 4).

5 Evaluation

We compare our algorithm against state-of-the-art using a RGB-D dataset consisting of 11 images which were captured with the Microsoft Kinect sensor. Ground truth labels were generated manually.

The quality of an oversegmentation is generally measured through boundary recall and undersegmentation error [5]. The results in figure 4 show that DASP outperforms Turbopixels and SLIC with respect to boundary recall. The undersegmentation error is bigger for DASP due to noisy depth values near geometry edges.

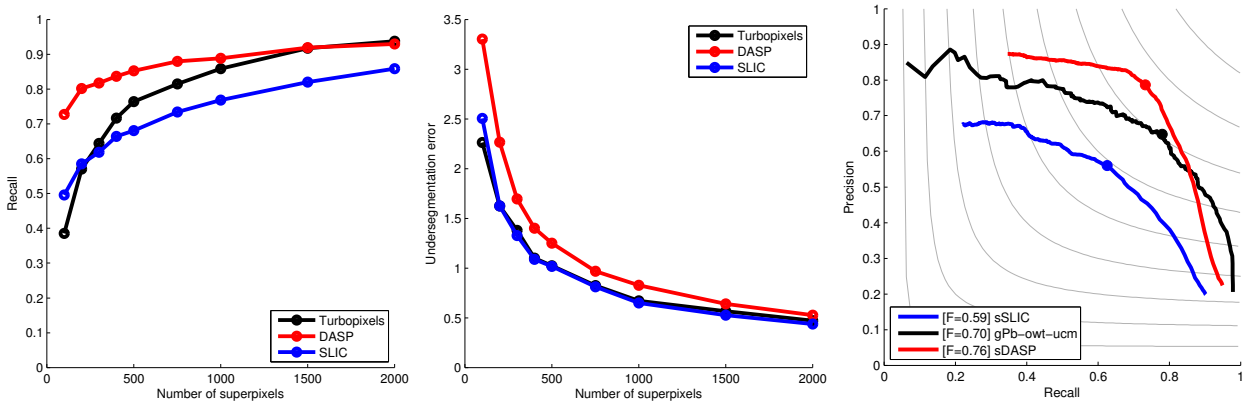


Figure 4. Boundary recall (left) and undersegmentation error (middle) for oversegmentation algorithms and contour detector quality (right) are compared on our dataset.

To evaluate segmentation quality we compare our algorithm against the algorithm gPb-owt-ucm from [2] under the measure of segment boundary quality. gPb-owt-ucm ranks highest on the Berkeley Segmentation Dataset (BSDS300) Benchmark [6] with a maximal F-score of 0.71. On our dataset, gPb-owt-ucm achieves a maximal F-score of 0.70, while sDASP scores 0.76. Figure 4 shows the precision/recall graph of gPb-owt-ucm and sDASP and sSLIC with 1000 superpixels.

In table 1 we report execution times for a 2.53 GHz single-core CPU. Our algorithms outperforms Turbopixels and gPb-owt-ucm by several orders of magnitudes. All results are with 20 k-means clustering iterations. Reducing the number of iterations to 3 results in a runtime of ca. 10 fps with slightly reduced quality.

	Turbo	DASP	gPb	sDASP
n=200	16.5 s	**0.53 s**		**0.54 s**
n=1000	13.6 s	**0.58 s**	218 s	**1.64 s**
n=2000	13.8 s	**0.62 s**		**7.36 s**

Table 1. Comparison of algorithm runtime

6 Conclusions

DASP is a novel oversegmentation algorithm for RGB-D images. In contrast to previous approaches it uses 3D information in addition to color. With depth-adaptive superpixels, previously costly segmentation methods are available to realtime scenarios, opening a variety of new applications. We demonstrate this by using spectral graph theory for an image segmentation which outperforms state-of-the-art algorithms both in quality and speed.

The source code and data set used for this paper are available at *https://github.com/Danvil/dasp*.

References

[1] R. Achanta, A. Shaji, K. Smith, and A. Lucchi. Slic superpixels. *Technical Report 149300, EPFL*, 2010.

[2] P. Arbeláez, M. Maire, C. Fowlkes, and J. Malik. Contour detection and hierarchical image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2011.

[3] R. Fattal. Blue-noise point sampling using kernel density model. *SIGGRAPH*, 2011.

[4] A. Lagae and P. Dutré. A Comparison of Methods for Generating Poisson Disk Distributions. *Computer Graphics Forum*, 2008.

[5] A. Levinshtein, A. Stere, K. N. Kutulakos, D. J. Fleet, S. J. Dickinson, and K. Siddiqi. TurboPixels: fast superpixels using geometric flows. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2009.

[6] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A Database of Human Segmented Natural Images and its Application to Evaluating Segmentation Algorithms and Measuring Ecological Statistics. *IEEE International Conference on Computer Vision*, 2004.

[7] X. Ren and J. Malik. Learning a classification model for segmentation. *IEEE International Conference on Computer Vision*, 2003.

[8] J. Shi and J. Malik. Normalized Cuts and Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2000.

[9] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake. Real-Time Human Pose Recognition in Parts from Single Depth Images. *IEEE Conference on Computer Vision and Pattern Recognition*, 2011.