DEPTH-ADAPTIVE SUPERVOXELS FOR RGB-D VIDEO SEGMENTATION

David Weikersdorfer
Neuroscientific System Theory
Technische Universität München

Alexander Schick
Fraunhofer IOSB
Karlsruhe

Daniel Cremers
Computer Vision & Image Analysis
Technische Universität München

ABSTRACT

In this paper we present a method for automatic video segmentation of RGB-D video streams provided by combined colour and depth sensors like the Microsoft Kinect. To this end, we combine position and normal information from the depth sensor with colour information to compute temporally stable, depth-adaptive superpixels and combine them into a graph of strand-like spatiotemporal, depth-adaptive supervoxels. We use spectral graph clustering on the supervoxel graph to partition it into spatiotemporal segments. Experimental evaluation on several challenging scenarios demonstrates that our two-layer RGB-D video segmentation technique produces excellent video segmentation results.

Index Terms— Superpixels, Supervoxels, RGB-D, Video Segmentation, Video Analysis

1. INTRODUCTION

Superpixel segmentation is an oversegmentation technique that received increasing attention in the last years. Superpixels align well with object boundaries and are generally of a compact shape. These properties allow using them as input in other algorithms instead of pixels. Their main advantage is the reduction of the input complexity from tens of thousands of pixels to only a couple of hundred superpixels. There exists a large variety of superpixel segmentation algorithms with different properties [1, 2, 3, 4, 5, 6].

Supervoxels extend the planar superpixels into the third dimension by not only clustering the pixels in each image, but by segmenting a stack of images. The image stack can either be composed by frames of volumetric scans (spatial image stack, e.g. medical scans) or by stacking video frames (temporal). Examples include Veksler et al. [7] and Lucchi et al. [8]. A recent survey and evaluation of supervoxel segmentations was given by Xu and Corso [9]. One advantage of using supervoxels is similar to superpixels: reduction of the number of primitives and grouping of similar pixels to one compact representation primitive. The main application area for temporal supervoxels is video segmentation [10].

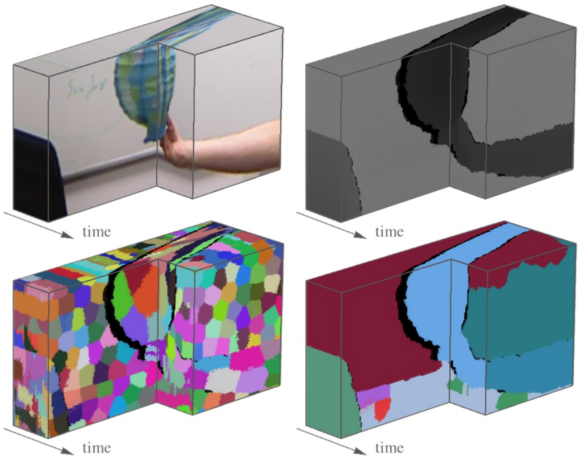

Fig. 1: Our method computes coherent, spatiotemporal depth-adaptive supervoxels (lower left) and consistent temporal segments (lower right) for a RGB-D video stream (upper).

Recently, a method was proposed for computing homogeneous, depth-adaptive superpixels (DASP) [4] for RGB-D images which are uniformly distributed on the surface of the 3D scene geometry. As it considers depth in addition to colour, significantly better superpixels properties and image segmentations can be achieved.

In this paper, we propose depth-adaptive supervoxels (DASV) which are the extension of DASP to the temporal domain. DASV are formed as coherent strands of temporally stable depth-adaptive superpixels and build an oversegmentation of the temporal image stack. We use spectral graph segmentation to segment the DASV into coherent segments to provide classic video segmentations. Our method is thus a hierarchical video segmentation technique with only two layers: supervoxels and segments (see fig. 1). It is unsupervised, model-free, runs in near realtime and shows very good results compared to the state of the art.

This paper is outlined as follows: In §2, we introduce temporally stable, depth-adaptive superpixels. They are used in §3 to construct a spatiotemporal supervoxel graph which is segmented using a spectral segmentation technique. The paper will be concluded with an evaluation of our method in §4.

Thanks to Prof. Jörg Conradt from the Neuroscientific System Theory Group, Technische Universität München, for supporting this project.

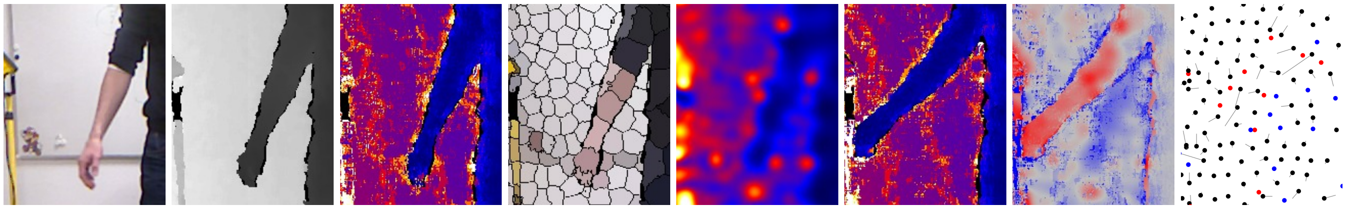

Fig. 2: **From left to right**: Colour and depth input, superpixel density, depth-adaptive superpixels, cluster density, target density for new frame, difference between previous and new density (red resp. blue indicates that clusters have to be removed resp. added) and result of delta density sampling (red: removed superpixel, blue: added superpixel, black: moved superpixel).

2. STABLE DEPTH-ADAPTIVE SUPERPIXELS

Depth-adaptive superpixels are computed frame by frame in three steps (see fig. 2). First, the depth-adaptive superpixel cluster density is computed from the depth input image. Second, a Poisson disc sampling method (e.g. [11]) is used to sample initial cluster centres. Third, sampled cluster centres are used in an density-adaptive local iterative clustering algorithm to assign pixels to cluster centres. For details see [4].

It turns out that the extension of superpixels to the temporal domain is by no means straight-forward. A naive frame-by-frame processing, for example, will lead to a severe jittering of superpixels over time because samples and superpixels are determined independently in each frame. For most applications this jittering is quite undesirable. The key idea of our approach is to induce a stable superpixel distribution by propagating density information over time.

Our method has two main advantages for RGB-D video streams: On the one hand it is straightforward to establish temporal superpixel connections between consecutive frames and on the other hand the number of iterations of the nearest-neighbour pixel assignment step can be reduced for higher processing performance.

The key idea for our temporally stable sampling method is the comparison between the superpixel density ρ_C provided from superpixel centres $C^{(t-1)}$ from previous the frame $t-1$ and the target superpixel density ρ_D computed from the depth image $D^{(t)}$ of the current frame t (see fig. 2). ρ_C is approximated using a classical kernel density estimator of the form

$$\rho_C(x \,|\, C^{(t-1)}) = \sum_{i=1}^{n} k_{\sigma_i}\left(\|x - x_i\|\right) \tag{1}$$

with $C^{(t-1)} = \{(x_i, \sigma_i)\}$, x_i cluster position and σ_i cluster scale. In our case we use a two-dimensional gaussian kernel

$$k_\sigma(d) = \frac{1}{\sigma^2}\, e^{-\pi \frac{d^2}{\sigma^2}} \,. \tag{2}$$

The depth-adaptive superpixels target density ρ_D is

$$\rho_D(x \,|\, D^{(t)}) \propto \left(D^{(t)}(x)\right)^2 \sqrt{\|\nabla D^{(t)}(x)\|^2 + 1} \tag{3}$$

which corresponds to the area of an infinitesimal surface element in 3D space. ρ_D is normalized such that the integral gives the desired number of superpixels. The difference betweenn theses two density functions

$$\Delta\rho^{(t)}(x) = \rho_D(x \,|\, D^{(t)}) - \rho_C(x \,|\, C^{(t-1)}) \,. \tag{4}$$

- a pseudo density function which can be both positive and negative - is used in a hierarchical sampling process like in [11]. The sampling process starts with all cluster centres $C^{(t-1)}$ from the previous frame. During the sampling process there may occur two cases. Either sample points are to be added, this is carried out normally, or sample points are to be removed. In this case the point nearest to the location where a point shall be removed is found in the set of all points currently sampled and removed. Fig. 2 exemplarily shows the result of such a sampling process.

3. SUPERVOXEL GRAPH AND SEGMENTATION

Our video segmentation technique consists of two main steps: construction of the weighted supervoxel graph and its segmentation using spectral methods.

First, for each new frame at time t of the RGB-D video stream, temporal-stable, depth-adaptive superpixels $S^{(t)} = \{s_i^{(t)}\}$ are computed and combined to a graph $G = (V, W)$ of depth-adaptive supervoxels where each supervoxel $v \in V$ consists of a series of superpixels $v = (s_{j_1}^{(T-n+1)}, \dots, s_{j_n}^{(T)})$, thus forms a supervoxel of strand-like shape (see fig. 3).

The last superpixels $S^{(t-1)} := \{s_i^{(t-1)}\}$ of active supervoxels, i.e. supervoxels where $T = t-1$, are used to compute the superpixel density $\rho_C(x|S^{(t-1)})$. Supervoxels are continued by trying to attach each new superpixel $s_j^{(t)}$ to the supervoxel which provided the seed point for the sampling and clustering process (see §2). We assure that

$$v = (\dots, s_i^{(t-1)}, s_j^{(t)}) \text{ iff. } \|s_i^{(t-1)} - s_j^{(t)}\|_{\mathcal{D}} \le \theta \tag{5}$$

where $\|\cdot\|_{\mathcal{D}}$ is a metric on the superpixel features, i.e. colour, spatial position and normal. If a supervoxel can not be continued, the superpixel starts a new supervoxel in the supervoxel graph. If during the sampling process a superpixel seed is deleted, the corresponding supervoxel in the supervoxel graph is closed and if a new seed is created, a new supervoxel is started (see fig. 3).

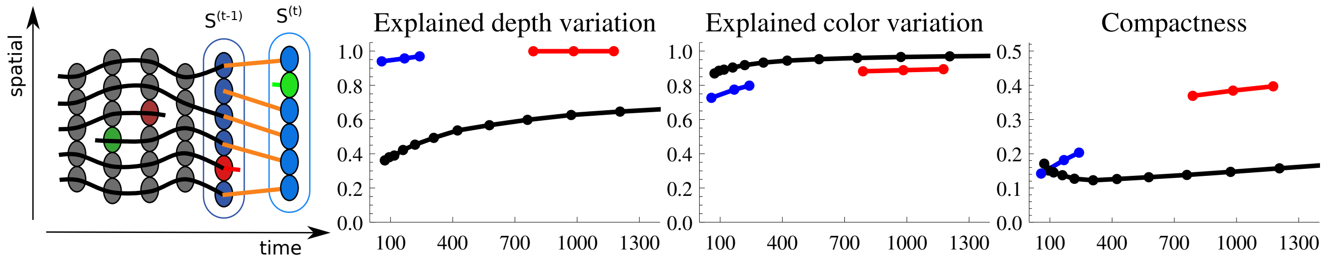

Fig. 3: **Left**: Creation of strand-like supervoxels from frame superpixels. Red/green are superpixels removed/added during sampling. Orange connections are to be tested. **Middle**: Colour and depth explained variation metric for our method (red: supervoxels, blue: supervoxel segments) and StreamGBH (black). **Right**: Compactness metric for our method and StreamGBH.

The weight W_{ij} of an edge connecting now active supervoxels i, j is updated using an exponential decay model:

$$W_{ij}^{(t)} = (1-\alpha)\, W_{ij}^{(t-1)} + \alpha\, n_{ij}^{(t)} \tag{6}$$

where n_{ij} is the corresponding superpixel similarity value from the current superpixel neighbourhood graph (see [4]).

Second, a spectral graph segmentation technique [5, 12] is used to segment the supervoxel graph. The segmentation method is identical to the spectral graph technique for RGB-D image segmentation using depth-adaptive superpixels in [4] where the general eigenvalue problem

$$(D - W)\, x = \lambda\, D\, x \ , \quad D_{ii} = \sum\nolimits_j W_{ij} \tag{7}$$

is solved, with W the symmetric adjacency matrix of the supervoxel graph. We limit the maximum number of supervoxels in the graph by excluding old supervoxels to assure a reasonable runtime of the spectral graph segmentation process.

The eigensystem solution of eq. 7 is used as in [4] to compute graph edge weights which form an ultrametric contour map (UCM) [12]. The UCM is thresholded and processed in an automatic semi-supervised label propagation step as in [10] to compute supervoxel segment labels which are temporally stable relative to the labelling from previous timesteps.

4. EVALUATION

We compare our method against the state-of-the-art hierarchical streaming video segmentation method StreamGBH [10]. While StreamGBH generates a deep hierarchy of superpixels of increasing size, our method only uses two layers: supervoxels and supervoxel segments (see fig. 4).

We report results under two well-established metrics for superpixels and segmentations: Explained variation and compactness. The explained variation metric [9, 13] is

$$R^2 = \frac{\sum_i \|\mu_i - \overline{\mu}\|^2}{\sum_i \|x_i - \mu_i\|^2} \tag{8}$$

with x_i pixel value for pixel i, μ_i mean value for corresponding superpixel, $\overline{\mu}$ mean value over all pixels. It can show a correlation to human annotations in certain scenarios [14] and is reported for colour and depth. Superpixel compactness is computed using the isoperimetric quotient [6].

In fig. 3 we compare our two-layer method for varying UCM edge thresholds (blue line) and varying superpixel numbers (red line) against the multi-layer method StreamGBH on a dataset consisting of six video sequences. Our method generates segments which are both compact and have a high value for explained colour variation, in contrast to StreamGBH which overfits on colour information at the cost of compactness. In addition, our method achieves significantly better results for explained depth variation which is a strong indicator for segmentations which respect geometry borders and thus perform better overall. A close-up comparison in fig. 4 demonstrates the shortcomings of StreamGBH regarding overfitted superpixels and unmet depth boundaries which our method does not possess.

In fig. 5 we report segmentations for our method and StreamGBH for three challenging scenarios and demonstrate that our method can handle difficult situations with fast motions, partial occlusions, textured objects and lighting changes.

As the supervoxel graph is changing gradually over time, it is not necessary to compute the graph segmentation step for every frame. Computing segmentations every fourth frame, our method has a near realtime performance of 0.38 seconds per frame compared to 71.4 seconds per frame for StreamGBH.

More labelling results and the source code of our DASV implementation can be found on the project homepage [1].

5. CONCLUSION

We proposed depth-adaptive supervoxels, a segmentation technique for RGB-D video streams which respects both temporal and spatial coherence and applied it to streaming video segmentation. Our evaluation showed that we outperform the state of the art for explained depth variation and compactness with significantly better runtimes.

[1] Project homepage: `https://github.com/Danvil/dasv`

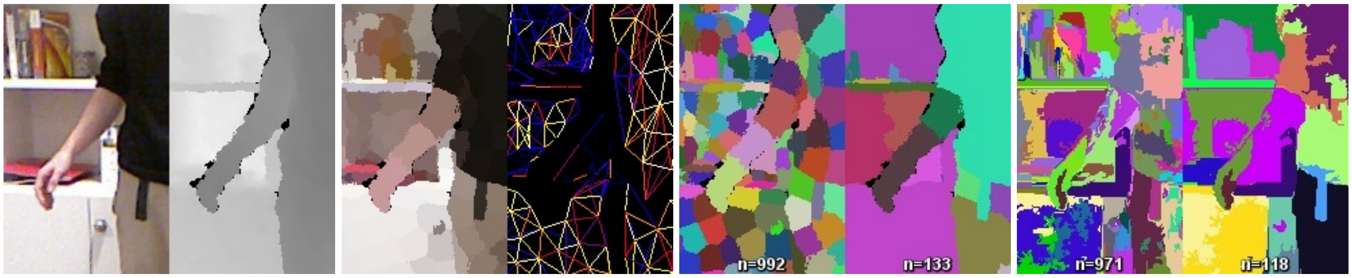

Fig. 4: **Left**: Colour and depth input. **Middle left**: Supervoxels (mean colour) and supervoxel neighbourhood graph (lighter colour is higher similarity). **Middle right**: Supervoxels (random colour) and supervoxel segments generated by our two-layer method. **Right**: Selected layers of the 21-layer hierarchy generated by StreamGBH. n is number of segments in the complete image.

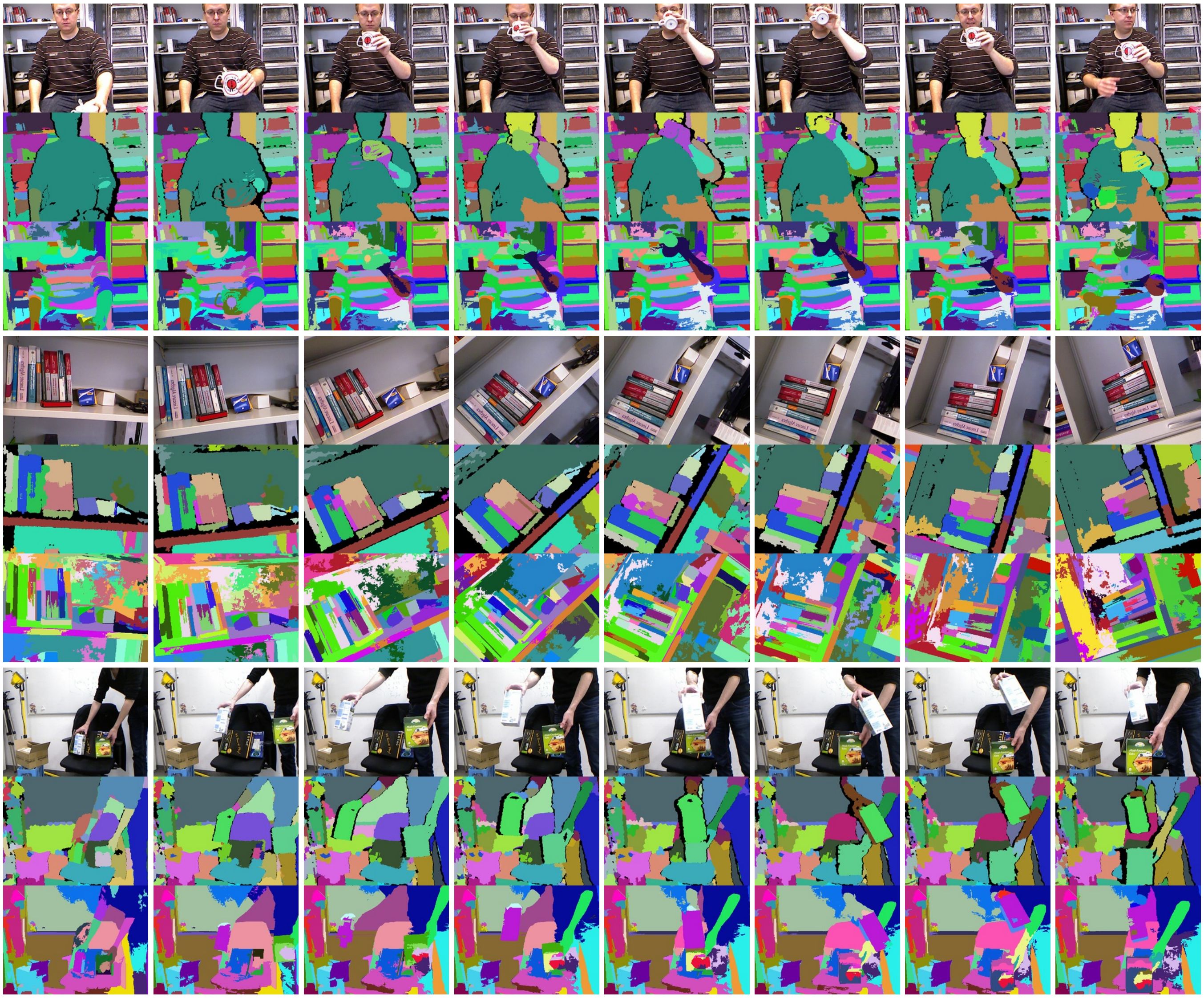

Fig. 5: Input colour images (first row), segmentation results from our method (second row) and from StreamGBH (third row). **First scenario**: Hand and object movements with textured surfaces and cluttered background. **Second scenario**: Camera movement and rotation. **Third scenario**: A challenging scenario with inter-object occlusions and changes in object lighting.

6. REFERENCES

[1] 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.

[2] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, "Slic superpixels," Tech. Rep., EPFL, 2010.

[3] F. Perbet and A. Maki, "Homogeneous superpixels from random walks," in *IAPR Conference on Machine Vision Applications*, 2011.

[4] D. Weikersdorfer, D. Gossow, and M. Beetz, "Depth-adaptive superpixels," in *International Conference on Pattern Recognition*, 2012.

[5] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2000.

[6] A. Schick, M. Fischer, and R. Stiefelhagen, "Measuring and evaluating the compactness of superpixels," in *International Conference on Pattern Recognition*, 2012.

[7] O. Veksler, Y. Boykov, and P. Mehrani, "Superpixels and supervoxels in an energy optimization framework," in *European Conference on Computer Vision*, 2010.

[8] A. Lucchi, K. Smith, R. Achanta, G. Knott, and P. Fua, "Supervoxel-based segmentation of em image stacks with learned shape features," Tech. Rep., EPFL Technical Report, 2010.

[9] C. Xu and J.J. Corso, "Evaluation of super-voxel methods for early video processing," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

[10] C. Xu, C. Xiong, and J.J. Corso, "Streaming hierarchical video segmentation," *Proceedings of European Conference on Computer Vision*, 2012.

[11] R. Fattal, "Blue-noise point sampling using kernel density model," *ACM Transactions on Graphics (SIGGRAPH)*, 2011.

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

[13] A. P. Moore, S. J. D. Prince, J. Warrell, and U. Mohammed, "Superpixel lattices," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

[14] C.E. Erdem, B. Sankur, and A.M. Tekalp, "Performance measures for video object segmentation and tracking," *IEEE Transactions on Image Processing*, 2004.