

Image Segmentation from Shadow-Hints using Minimum Spanning Trees

Moritz Heep
mheep@uni-bonn.de
University of Bonn
Bonn, Germany

Eduard Zell
e.zell@hotmail.de
University of Bonn
Bonn, Germany

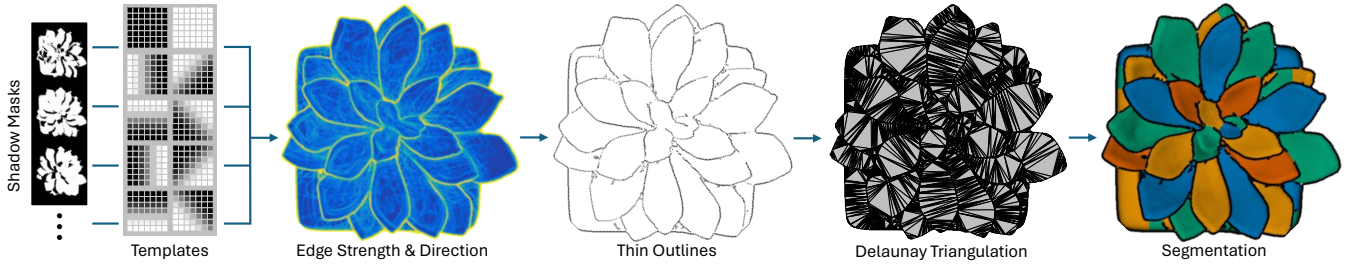


Figure 1: Overview of our Pipeline: Starting from a set of shadow masks, we use templates to extract light-to-shadow transitions. After combining these transitions into an edge strength and direction, we apply non-maximum suppression to obtain thin outlines. The segmentation is performed on a Delaunay triangulation of the detected outline points.

CCS CONCEPTS

• Computing methodologies → Computer graphics.

KEYWORDS

Edge detection, Segmentation

ACM Reference Format:

Moritz Heep and Eduard Zell. 2024. Image Segmentation from Shadow-Hints using Minimum Spanning Trees. In *Special Interest Group on Computer Graphics and Interactive Techniques Conference Posters (SIGGRAPH Posters '24)*, July 27 - August 01, 2024. ACM, New York, NY, USA, 2 pages. <https://doi.org/10.1145/3641234.3671025>

1 INTRODUCTION

Image segmentation in RGB space is a notoriously difficult task where state-of-the-art methods are trained on thousands or even millions of annotated images [Kirillov et al. 2023]. While the performance is impressive, it is still not perfect. We propose a novel image segmentation method, achieving similar segmentation quality but without training. Instead, we require an image sequence with a static camera and a single light source at varying positions, as used in for photometric stereo, for example. Here, foreground objects cast shadows onto background objects, the detection of transitions from light to shadow can be used to reveal the spatial structure of the scene and to trace the contour of an object. Unfortunately, these contours are not water-tight and simple flood-fill approaches fail. Inspired by interactive sketch colouring methods [Parakkat et al.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

SIGGRAPH Posters '24, July 27 - August 01, 2024, Denver, CO, USA

© 2024 Copyright held by the owner/author(s).

ACM ISBN 979-8-4007-0516-8/24/07

<https://doi.org/10.1145/3641234.3671025>

2022], our novel image segmentation algorithm is based on Delaunay triangulations. After converting the pixel grid to a mesh, our algorithm operates on the face graph of the Delaunay triangulation where there is no notion of colour similarity. Instead, we rely on the edge length as a similarity indicator due to the circumcircle property of the Delaunay triangulation. While comparable graph-based image segmentation algorithms [Felzenszwalb and Huttenlocher 2004] cluster pixels according to colour similarity and achieve - by current standards - only mediocre results, our method shows promising results without any training on annotated data.

2 METHOD

Instead of RGB images, we require binary shadow masks for each light position to generate a discontinuity-sensitive image segmentation. Such shadow masks are commonly created in photometric stereo, see e.g. [Heep and Zell 2022].

2.1 Shadow Edge Detection

We apply a template matching procedure to detect shadow-to-light transitions in all shadow masks and merge them into a pixel-based edge strength and direction. We use ten templates (Fig. 1), each 7×7 pixels big: Two for completely lit or shadowed regions and eight for light-to-shadow transitions in the directions $d \in \mathcal{N}$ of the eight pixel neighbourhood. Let $E_{lp,0}$, $E_{lp,1}$ and $E_{lp,d}$ be the resulting L^2 errors at pixel p under lighting l . Fully shadowed regions and transitions that cannot be explained by the current light position, cf. [Raskar et al. 2004], are excluded by setting the binary weight $\omega_{lp,d} \in \{0, 1\}$ to zero. We calculate the edge score

$$b_{p,d} = \frac{\sum_l \omega_{lp,d} \cdot \sigma((E_{lp,1} - E_{lp,d})/\beta)}{\sum_l \omega_{lp,d}} \in [0, 1] \quad (1)$$

for each direction $d \in \mathcal{N}$ and pixel p . The sigmoid function σ smoothly transitions between 0 and 1, depending on whether a

pixel is fully lit or the respective transition direction is a better fit. For each pixel, we choose the edge direction $\theta_p \in \mathcal{N}$ such that $b_{p,d}$ is maximal and use the maximum value g_p as edge strength.

2.2 Subpixel Delaunay Triangulation

To extract thin outlines from edge strength g_p and direction θ_p , we apply non-maximum suppression and double thresholding [Canny 1986]. Furthermore, we refine the location of each edge pixel, by employing a quadratic fit to locate the subpixel positions of the maxima in g_p . This yields smoother outlines and moves maxima away from pixel centres, i.e. pixels are unambiguously *within* one segment. The maxima are then connected through a Delaunay triangulation, completing our shift from the pixel domain to a 2D polygon mesh. We remove any triangles covering the image background.

2.3 Segmentation

Our segmentation algorithm progressively fuses triangles into larger segments, cf. [Kruskal 1956]. Since vertices are more densely placed along detected outlines, the shared edge length between two triangles is a suitable proxy for how likely these two triangles belong to the same segment. We exploit this by placing the edges in a priority queue with non-increasing edge length. Running any minimum spanning tree algorithm with these lengths would fuse all triangles into a single segment. Instead, we calculate the aspect ratio for each segment S as the ratio between its area $|S|$ and the length $|e|$ of its shortest edge:

$$l_S := \frac{|S| - A_{\min}}{\min_{e \in S}(|e|)}. \quad (2)$$

Reducing the segment area by A_{\min} ensures that segments smaller than A_{\min} are fused in the further course of the algorithm. When processing edge e , we check if

$$|e| > \kappa \cdot \min(l_S, l_{S'}) \quad (3)$$

where $\kappa > 0$ is a parameter to control the segment shape. If true, segments S and S' are fused along the edge e . Over time, the segments become larger and contain increasingly shorter edges. Hence, l_S increases as segments grow until no more fusions occur. The algorithm terminates when all edges have been processed. Segments smaller than A_{\min} are always fused.

3 EXPERIMENTS

We tested our algorithm on different objects against classic and learning-based image segmentations: FH04 [Felzenszwalb and Huttenlocher 2004] operates on a nearest-neighbour graph built from screen position and RGB colour. In contrast, the Segment Anything Model (SAM23) [Kirillov et al. 2023] is a state-of-the-art deep learning approach. FH04 leads to over-segmentation in textured regions (e.g. the wooden branch, Fig. 2) and under-segmentation for similarly coloured segments. SAM23 and our method are more robust and create comparable results in many cases. SAM23 is prone to over-segmentation in high-contrast textures (e.g. the origami butterfly, Fig. 2) while our approach is prone to under-segmentation if transitions between objects are too smooth to cast a shadow.

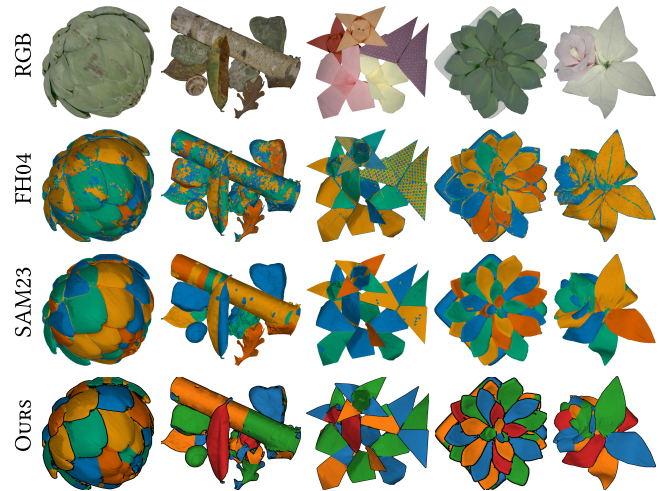


Figure 2: From Top to Bottom: RGB input used to generate segmentations with FH04 and SAM23 as well as our segmentation from shadow-hints. For our method, detected outline points are overlaid to visualize where completions occur.

4 CONCLUSION

Given the quality of the results without depending on annotated data, our method offers an alternative to create annotated datasets to train learning-based image segmentation algorithms. The granularity of the segmentation can be controlled in real-time through the user parameter κ and manual refinement is feasible at the segment level, instead of the pixel level.

ACKNOWLEDGMENTS

Special thanks to Amal Dev Parakkat for fruitful discussions. This work has been funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany’s Excellence Strategy, EXC-2070 – 390732324 (PhenoRob).

REFERENCES

- John Canny. 1986. A Computational Approach To Edge Detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on PAMI-8* (Dec. 1986), 679–698. <https://doi.org/10.1109/TPAMI.1986.4767851>
- Pedro F. Felzenszwalb and Daniel P. Huttenlocher. 2004. Efficient Graph-Based Image Segmentation. *International Journal of Computer Vision* 59 (2004), 167–181.
- Moritz Heep and Eduard Zell. 2022. ShadowPatch: Shadow Based Segmentation for Reliable Depth Discontinuities in Photometric Stereo. *Computer Graphics Forum* 41, 7 (Oct. 2022), 635–646. <https://doi.org/10.1111/cgf.14707>
- Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. 2023. Segment Anything. In Proceedings of the IEEE/CVF International Conference on Computer Vision. *arXiv:2304.02643*, 4015–4026. [arXiv:2304.02643](https://arxiv.org/abs/2304.02643)
- Joseph Bernard Kruskal. 1956. On the Shortest Spanning Subtree of a Graph and the Traveling Salesman Problem. *Proceedings of the American Mathematical Society* 7, 1 (1956), 48–50. <http://www.jstor.org/stable/2033241>
- Amal Dev Parakkat, Pooran Memari, and Marie-Paule Cani. 2022. Delaunay Painting: Perceptual Image Colouring from Raster Contours with Gaps. *Computer Graphics Forum* 41, 6 (Sept. 2022), 166–181. <https://doi.org/10.1111/cgf.14517>
- Ramesh Raskar, Kar-Han Tan, Rogerio Feris, Jingyi Yu, and Matthew Turk. 2004. Non-Photorealistic Camera: Depth Edge Detection and Stylized Rendering Using Multi-Flash Imaging. *ACM Transactions on Graphics (TOG)* 23, 3 (2004), 679–688.