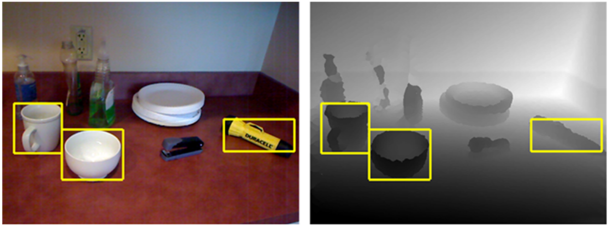
Generating Object Candidates from RGB-D Images and Point Clouds

Helge Wrede

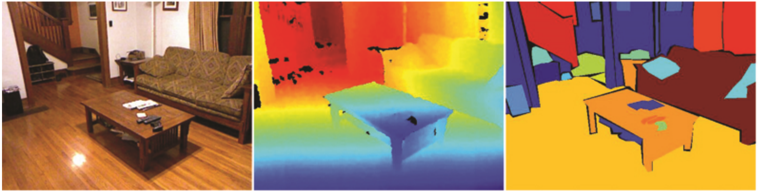
11.05.2017

Outline

Introduction



(a) Bounding box object candidates. [1]



(b) Segmented object candidates. [2]

Figure: Two types of object candidates generated from RGB-D images.

Methods Overview

2D

- BING [3]
- SLIC [4]
- Selective Search [5]

3D

- Objectness [6]
- Shape Analyses [7]
- Spectral Clustering [8]
- Selective Search [9]

Outline

RGB-D Images

Figure: Image from the NYU Depth V2 dataset; RGB on the left and Depth on the right. [2]

RGB-D Images

Structure

- Color in one image
- Depth in another image

RGB-D Images

Structure

- Color in one image
- Depth in another image

RGB image

- RGB camera

RGB-D Images

Structure

- Color in one image
- Depth in another image

RGB image

- RGB camera

Depth image

- Active stereo
- Time-of-flight
- Passive stereo

Point Clouds

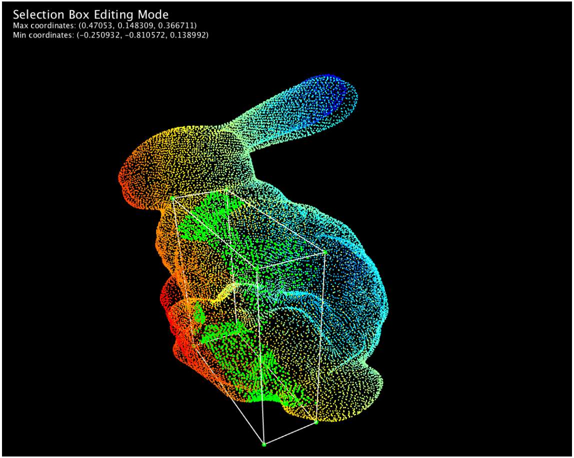


Figure: Example image of a point cloud. [10]

Point Clouds

Structure

- 3D coordinates (X, Y, Z) relative to the camera
- Color (RGB/HSV/LAB/etc.)

Point Clouds

Structure

- 3D coordinates (X, Y, Z) relative to the camera
- Color (RGB/HSV/LAB/etc.)

Generation

- Color information
- Depth information
- Camera position

Microsoft Kinect

(a) Kinect V1 (Xbox 360)

(b) Kinect V2 (Xbox One)

Figure: Both versions of the Microsoft Kinect. [11]

Microsoft Kinect [12]

V1

- RGB camera
- Structured-light camera (active stereo)

Microsoft Kinect [12]

V1

- RGB camera
- Structured-light camera (active stereo)

V2

- RGB camera
- Time-of-flight camera

Outline

Superpixel

Figure: Images segmented into superpixels of size 64, 256 and 1024 pixels (approx.) using SLIC. [4]

Superpixel

Properties

- Region with similar information
- Obtained from images

Generated with SLIC

SLIC (Simple Linear Iterative Clustering) [4]

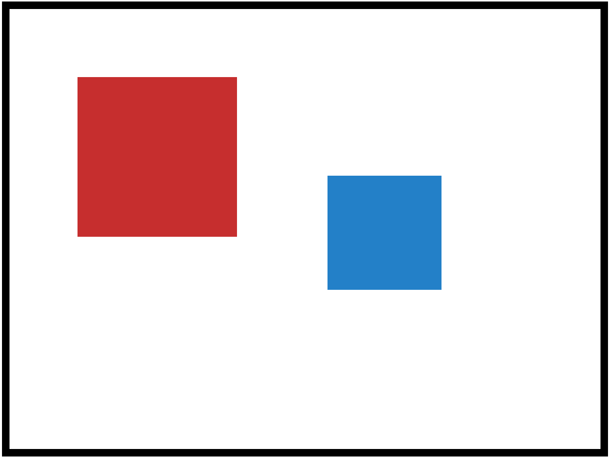


Figure: Initial image.

SLIC (Simple Linear Iterative Clustering) [4]

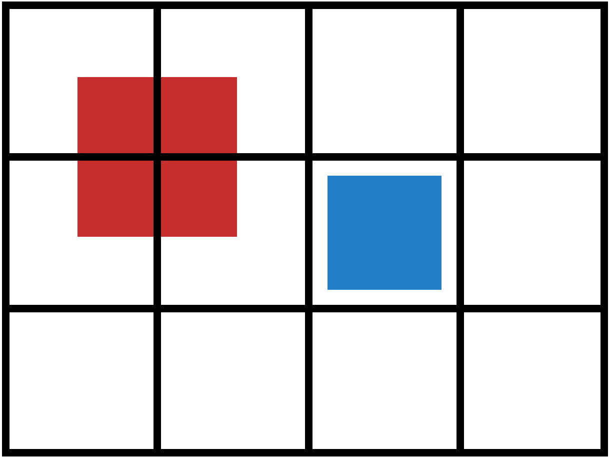


Figure: Seeding grid placed.

SLIC (Simple Linear Iterative Clustering) [4]

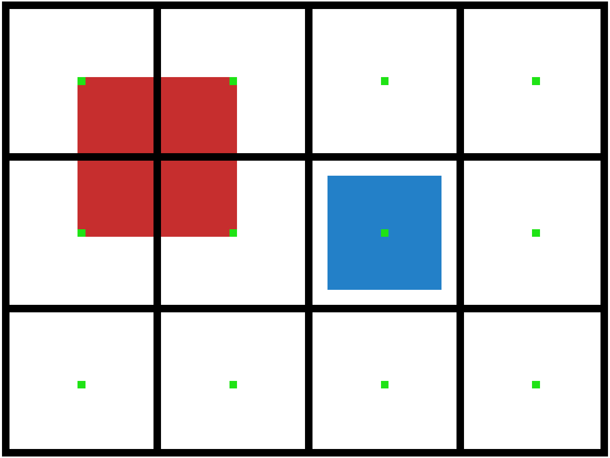


Figure: Seeding centers placed.

SLIC (Simple Linear Iterative Clustering) [4]

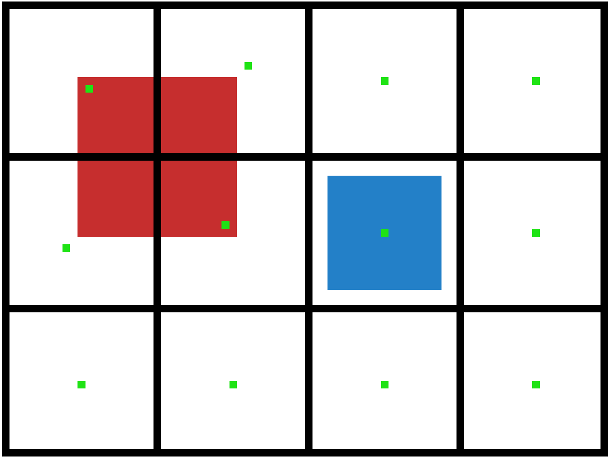


Figure: Centers moved away from edges and corners.

SLIC (Simple Linear Iterative Clustering) [4]

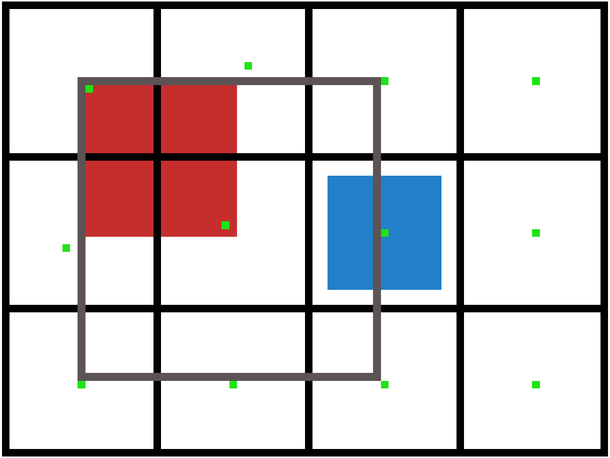


Figure: The area to search for fitting pixels.

SLIC (Simple Linear Iterative Clustering) [4]

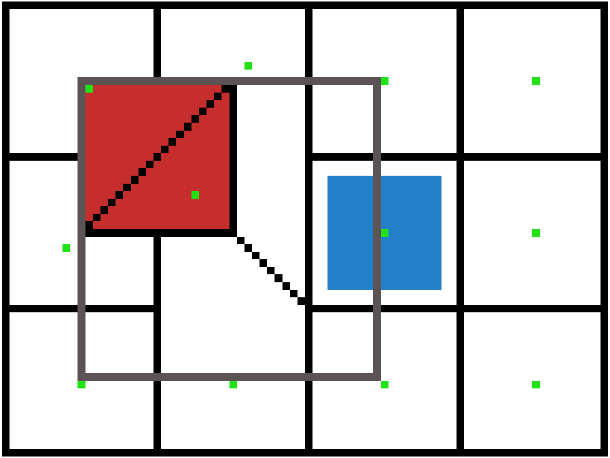


Figure: A few iterations of adding pixels and recentering.

SLIC (Simple Linear Iterative Clustering) [4]

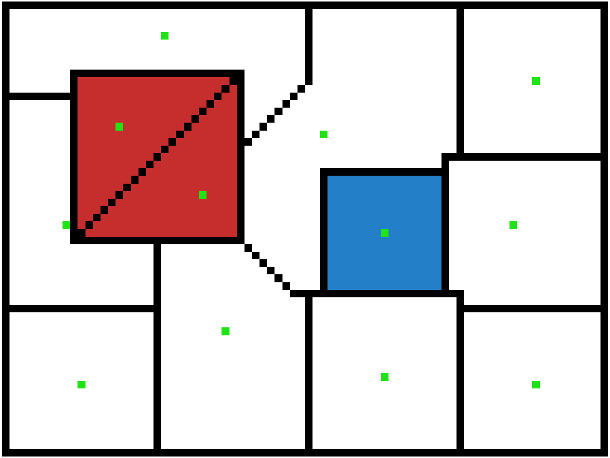


Figure: The superpixels after recentering and adding similar pixels.

SLIC (Simple Linear Iterative Clustering) [4]

Method Summary

- Subdivide the image
- Seed the centers
- Move the centers based on gradients
- Search neighbourhood for similar pixels
- Adjust the centers based on added pixels
- Add more similar pixels and adjust centers until finished

Supervoxel

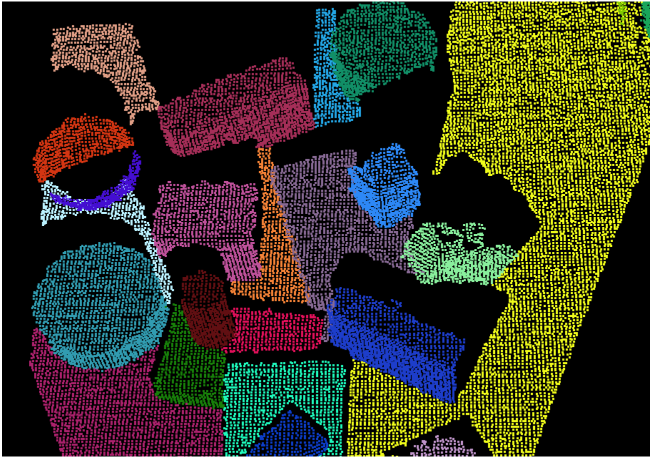


Figure: Spectral supervoxel clustering of cluttered table-top scene. [13]

Supervoxel

Properties

- Superpixel in 3D
- Obtained from point clouds

Supervoxel

Properties

- Superpixel in 3D
- Obtained from point clouds

Generation [9]

- K-means clustering
- Seed voxel in every cluster
- Combine all subsets of seed voxels
- Assign voxel to nearest supervoxel
- Update supervoxel centers

Superpixel and Supervoxel

Advantages

- Reduces complexity
- Potentially increases processing speed

Superpixel and Supervoxel

Advantages

- Reduces complexity
- Potentially increases processing speed

Disadvantages

- One extra step
- Needs to be accurate
- Might be too slow for realtime applications

Selective Search [9]

2D Selective Search

- Uses superpixels
- Iteratively merge adjacent superpixels based on similarity
  - Color (Histogram)
  - Texture (Gradient derivation)
  - Size (Fraction of the image)
  - Fill (Compactness)

Selective Search [9]

2D Selective Search

- Uses superpixels
- Iteratively merge adjacent superpixels based on similarity
  - Color (Histogram)
  - Texture (Gradient derivation)
  - Size (Fraction of the image)
  - Fill (Compactness)

3D Selective Search

- Uses supervoxels
- Iteratively merge adjacent supervoxels based on similarity
  - Color (Histogram)
  - Volume (Fraction of the point cloud)

Outline

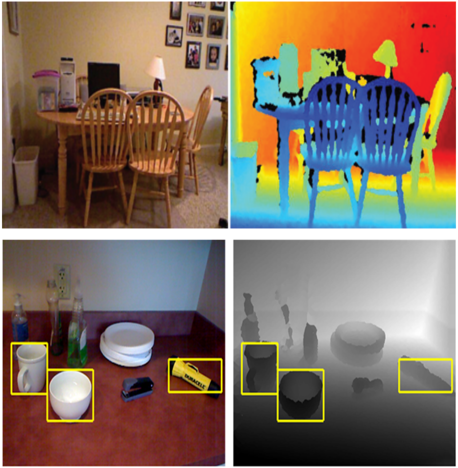
Uses

Collision Detection

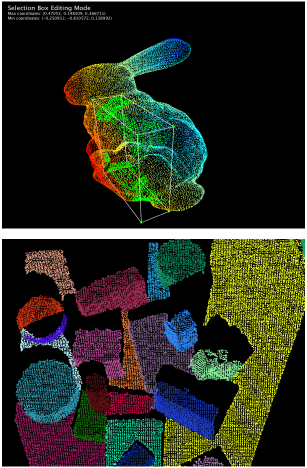
Building Block for

- Tracking
- Classification
- Recognition
- Orientation

Summary



(a) RGB-D images and the generated object candidates.



(b) Point clouds and the generated supervoxels.

References I

[1] Xiaoping Lou, Mingli Dong, Jun Wang, Peng Sun, and Yimin Lin.
Invariant Hough random ferns for RGB-D-based object detection.
Optical Engineering, 55(9):091403, 2016.

[2] Pushmeet Kohli Nathan Silberman, Derek Hoiem and Rob Fergus.
Indoor Segmentation and Support Inference from RGBD Images.
In *ECCV*, 2012.

References II

[3] Ming-Ming Cheng, Ziming Zhang, Wen-Yan Lin, and Philip Torr.
BING: Binarized Normed Gradients for Objectness Estimation at 300fps.
In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014.

[4] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Ssstrunk.
SLIC Superpixels Compared to State-of-the-Art Superpixel Methods.
IEEE Transactions on Pattern Analysis and Machine Intelligence, 34(11):2274–2282, Nov 2012.

[5] J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, and A. W. M. Smeulders.
Selective Search for Object Recognition.
International Journal of Computer Vision, 104(2):154–171, 2013.

[6] Aitor Aldoma, Markus Vincze, Federico Tombari, and Walter Kropatsch.
Localizing and Segmenting Objects with 3D Objectness.
In *Proc. the 18th Computer Vision Winter Workshop (CVWW 2013)*, 2013.

[7] A. Karpathy, S. Miller, and Li Fei-Fei.
Object discovery in 3D scenes via shape analysis.
In *2013 IEEE International Conference on Robotics and Automation*, pages 2088–2095, May 2013.

References IV

[8] Ulrike von Luxburg.
A tutorial on spectral clustering.
Statistics and Computing, 17(4):395–416, 2007.

[9] A. Kanezaki and T. Harada.
3D Selective Search for obtaining object candidates.
In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 82–87, Sept 2015.

[10] Image from the homepage of the 3D Point Cloud Editor.
`http://paradise.caltech.edu/~yli/software/pceditor.html`.
[Online; accessed 06.05.2017].

[11] Images from the Wikipedia article about the Kinect.
`https://en.wikipedia.org/wiki/Kinect`.
[Online; accessed 06.05.2017].

References V

[12] Presentation explaining the Microsoft Kinect functionality.
http://pages.cs.wisc.edu/~dyer/cs534/slides/17_kinect.pdf.
[Online; accessed 07.05.2017].

[13] Image from a blog about Point Clouds and Supervoxels.
http://www.pointclouds.org/blog/tocs/alexandrov/index.php.
[Online; accessed 07.05.2017].

[14] Jeremie Papon, Alexey Abramov, Markus Schoeler, and Florentin Worgotter.
Voxel Cloud Connectivity Segmentation - Supervoxels for Point Clouds.
In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2013.

References VI

[15] R. B. Rusu, N. Blodow, and M. Beetz.
Fast Point Feature Histograms (FPFH) for 3D registration.
In *2009 IEEE International Conference on Robotics and Automation*, pages 3212–3217, May 2009.