Fast Plane Extraction in Organized Point Clouds Using Agglomerative Hierarchical Clustering

Chen Feng[1], Yuichi Taguchi[2], and Vineet R. Kamat[1]

1) University of Michigan, USA

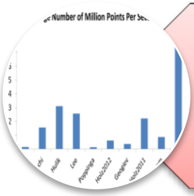
2) Mitsubishi Electric Research Labs, USA

June 4, 2014

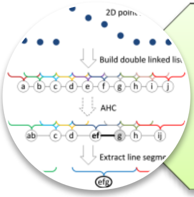
UNIVERSITY OF MICHIGAN

MITSUBISHI ELECTRIC

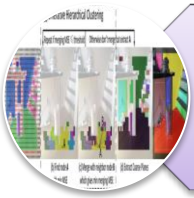
Outline



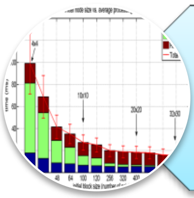
Introduction



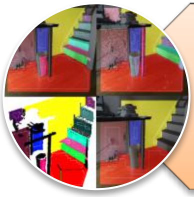
Overview



Breakdown



Experiments



Conclusions

- Real-time plane extraction is crucial to various applications in robotics, computer vision, and 3D modeling:
  - Table-top object manipulation
  - Landmarks for SLAM
  - Compact and semantic scene modeling

- Real-time plane extraction is crucial to various applications in robotics, computer vision, and 3D modeling:
  - Table-top object manipulation
  - Landmarks for SLAM
  - Compact and semantic scene modeling
- We present an efficient and reliable fast plane extraction algorithm applicable to organized point clouds, such as depth maps obtained by Kinect sensors.

- Previous Work

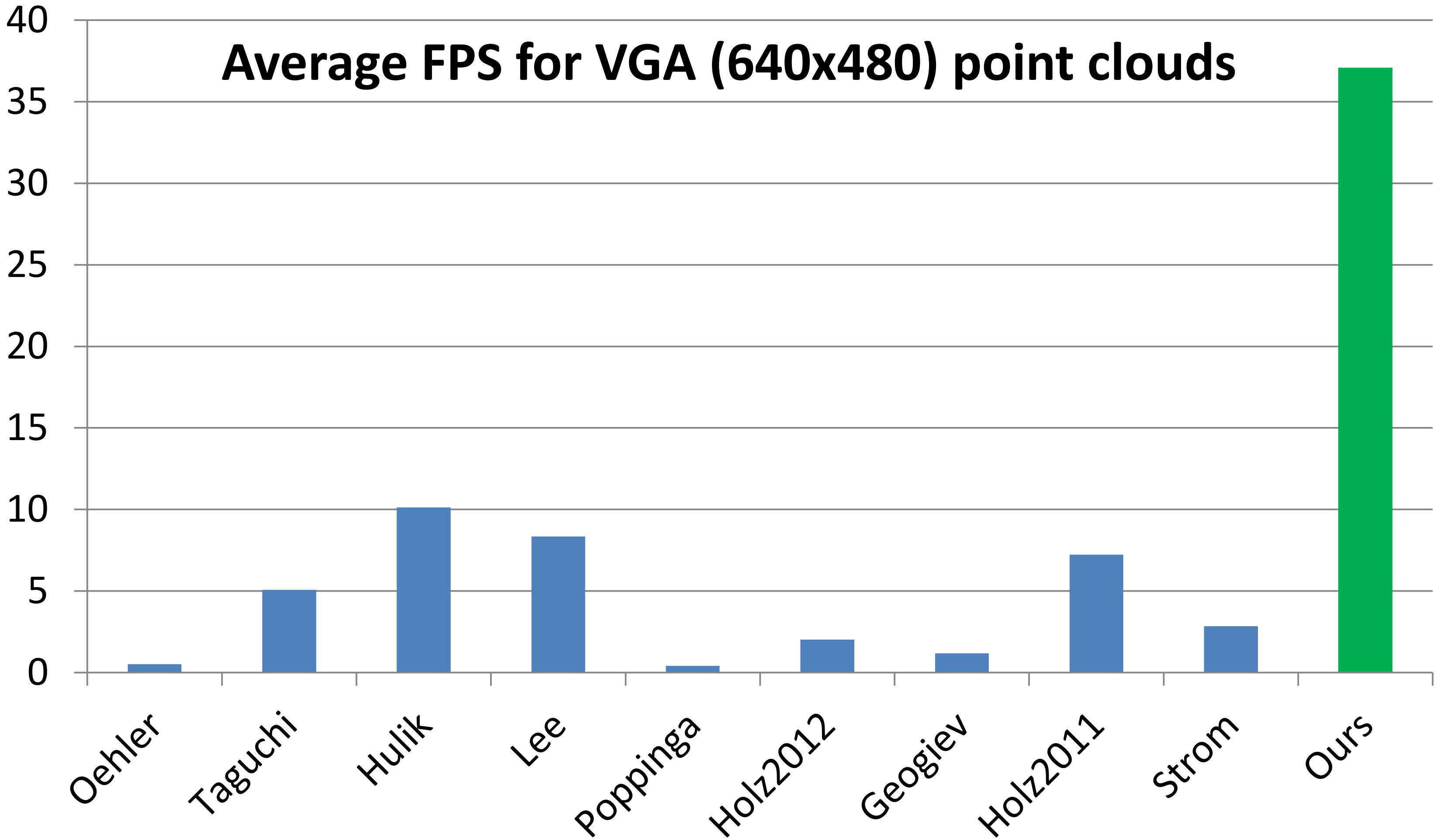
- Previous Work
  - RANSAC-based
    - “Surfels” from Hough Transform (Oehler et al. 2011)
    - RANSAC on local region (Taguchi et al. 2013; Hulik et al. 2012; Lee et al. 2012)

- Previous Work
  - RANSAC-based
    - "Surfels" from Hough Transform (Oehler et al. 2011)
    - RANSAC on local region (Taguchi et al. 2013; Hulik et al. 2012; Lee et al. 2012)
  - Region-grow-based
    - Point-plane distance/MSE threshold (Hahnel et al. 2003; Poppinga et al. 2008)
    - Surface normal deviation threshold (Holz & Behnke 2012)
    - Line segments grow (Georgiev et al. 2011)

- Previous Work
  - RANSAC-based
    - "Surfels" from Hough Transform (Oehler et al. 2011)
    - RANSAC on local region (Taguchi et al. 2013; Hulik et al. 2012; Lee et al. 2012)
  - Region-grow-based
    - Point-plane distance/MSE threshold (Hahnel et al. 2003; Poppinga et al. 2008)
    - Surface normal deviation threshold (Holz & Behnke 2012)
    - Line segments grow (Georgiev et al. 2011)
  - Graph-based (Strom et al. 2010; Wang et al. 2013)

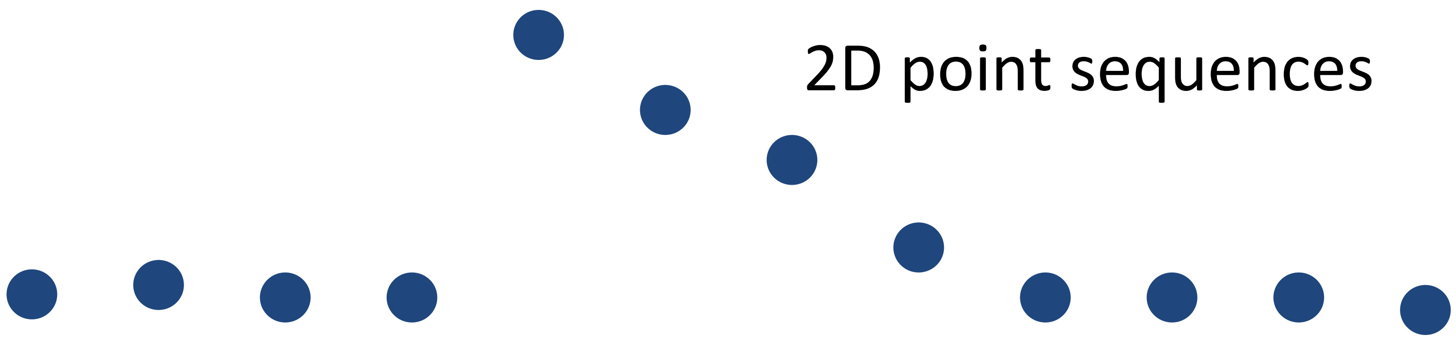
- Previous Work
  - RANSAC-based
    - "Surfels" from Hough Transform (Oehler et al. 2011)
    - RANSAC on local region (Taguchi et al. 2013; Hulik et al. 2012; Lee et al. 2012)
  - Region-grow-based
    - Point-plane distance/MSE threshold (Hahnel et al. 2003; Poppinga et al. 2008)
    - Surface normal deviation threshold (Holz & Behnke 2012)
    - Line segments grow (Georgiev et al. 2011)
  - Graph-based (Strom et al. 2010; Wang et al. 2013)
  - Other
    - Normal space clustering (Holz et al. 2011)
    - Gradient-of-depth feature (Enjarini et al. 2012)

- Previous Work

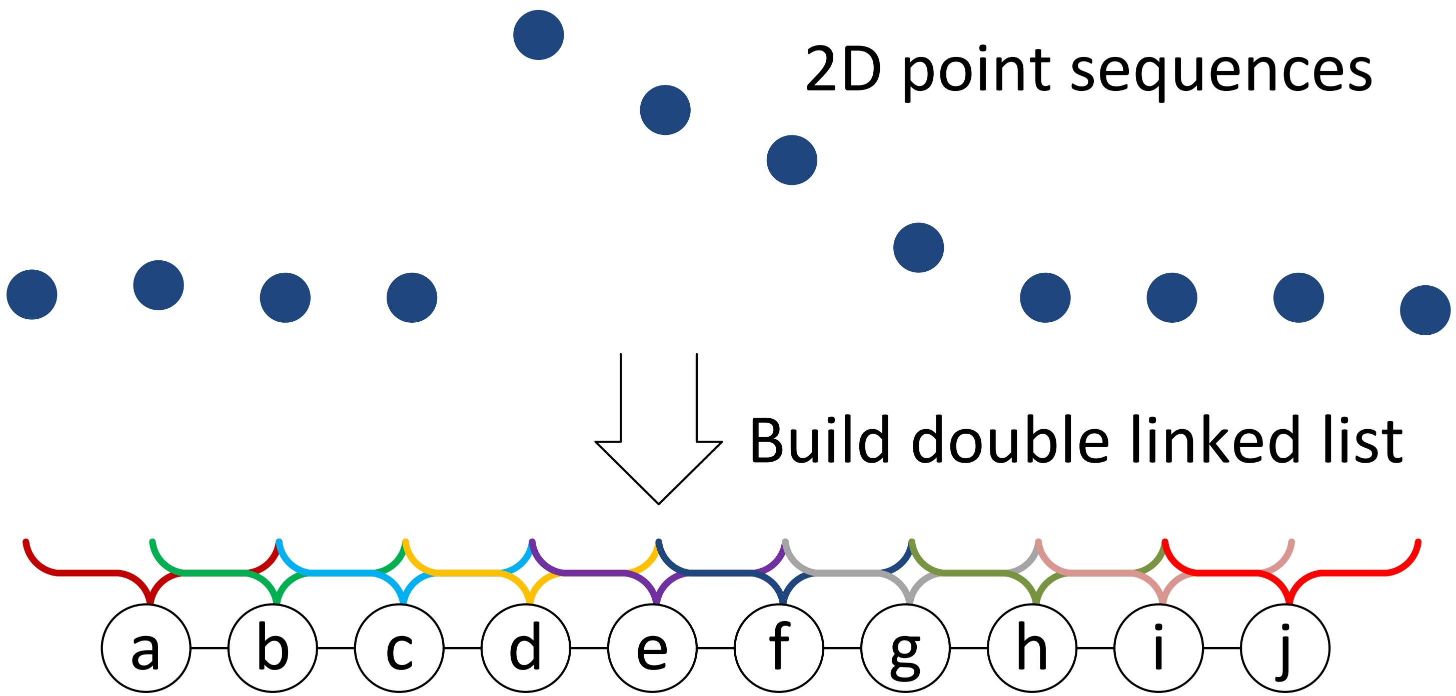


- Analogy to Line Regression (Nguyen et al. 2005; April Robotics Toolkit, 2010)
  - Exploit the neighborhood information given by the order of points

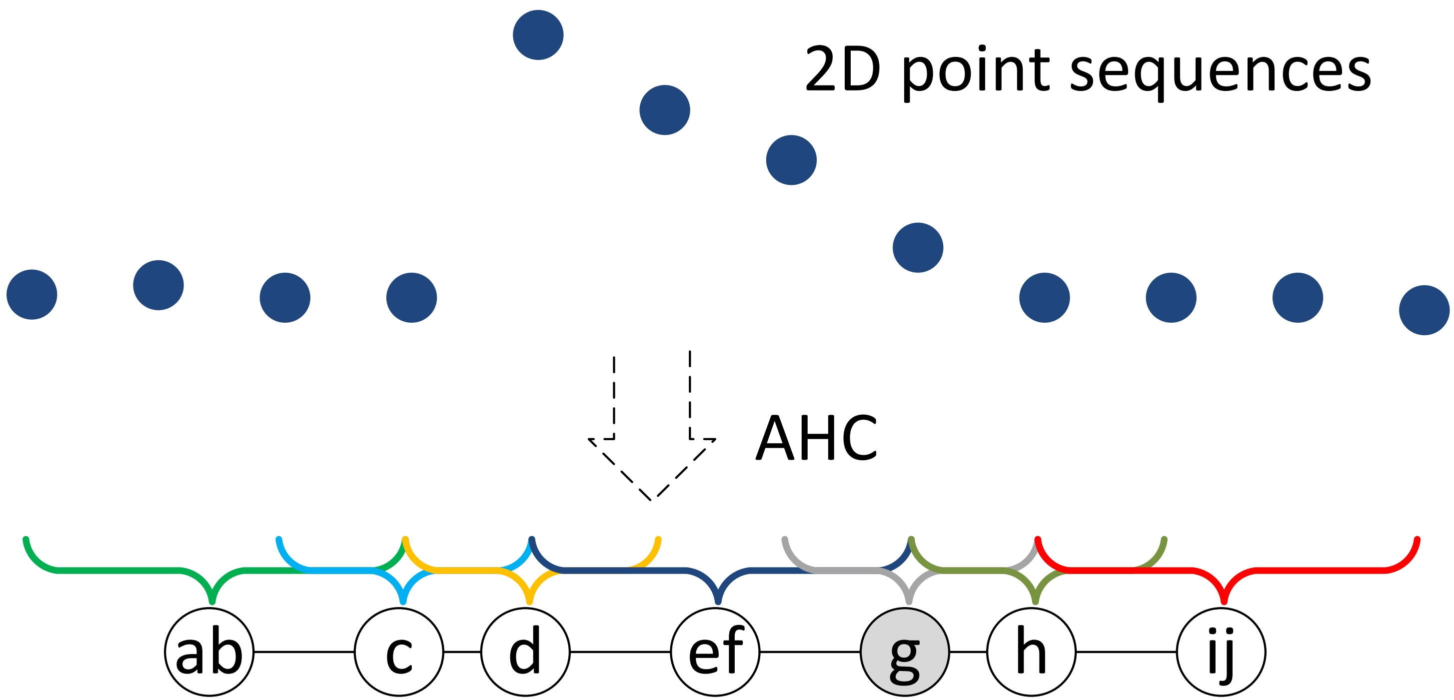
- Analogy to Line Regression (Nguyen et al. 2005; April Robotics Toolkit, 2010)
  - Exploit the neighborhood information given by the order of points



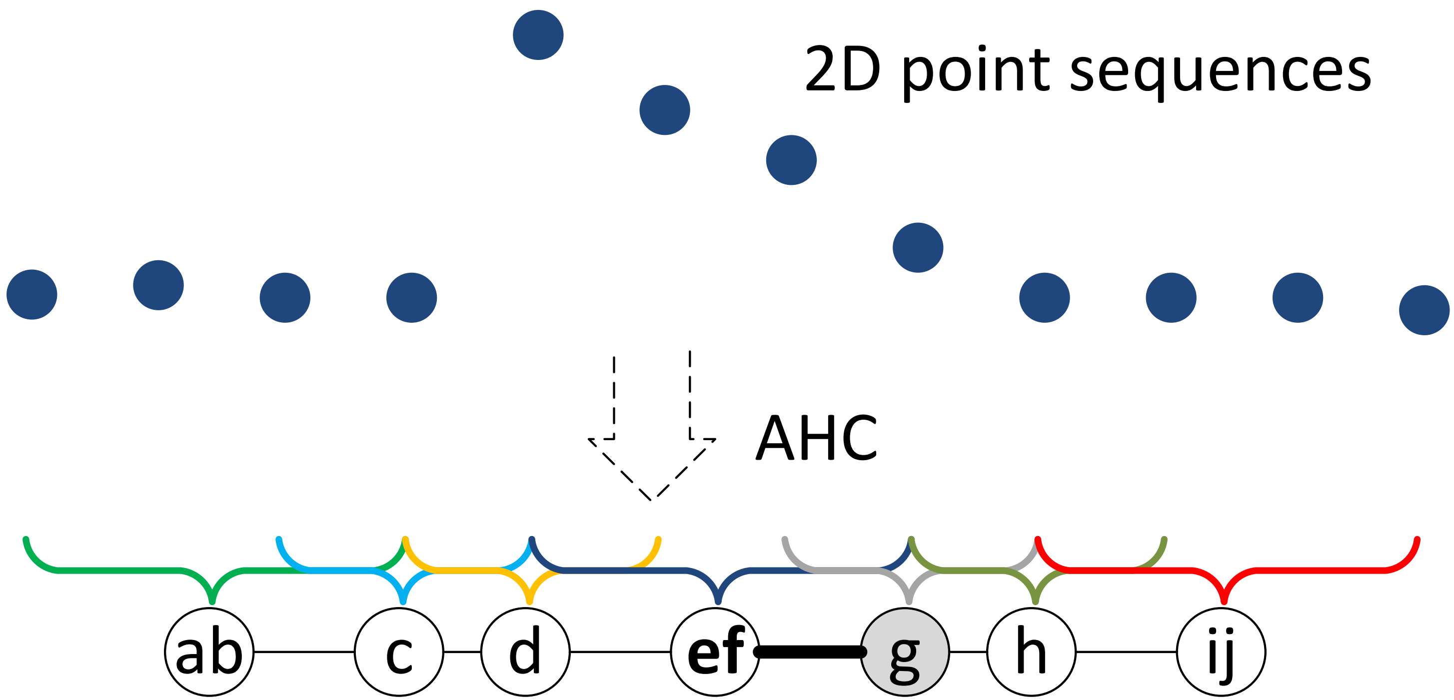
- Analogy to Line Regression (Nguyen et al. 2005; April Robotics Toolkit, 2010)
  - Exploit the neighborhood information given by the order of points



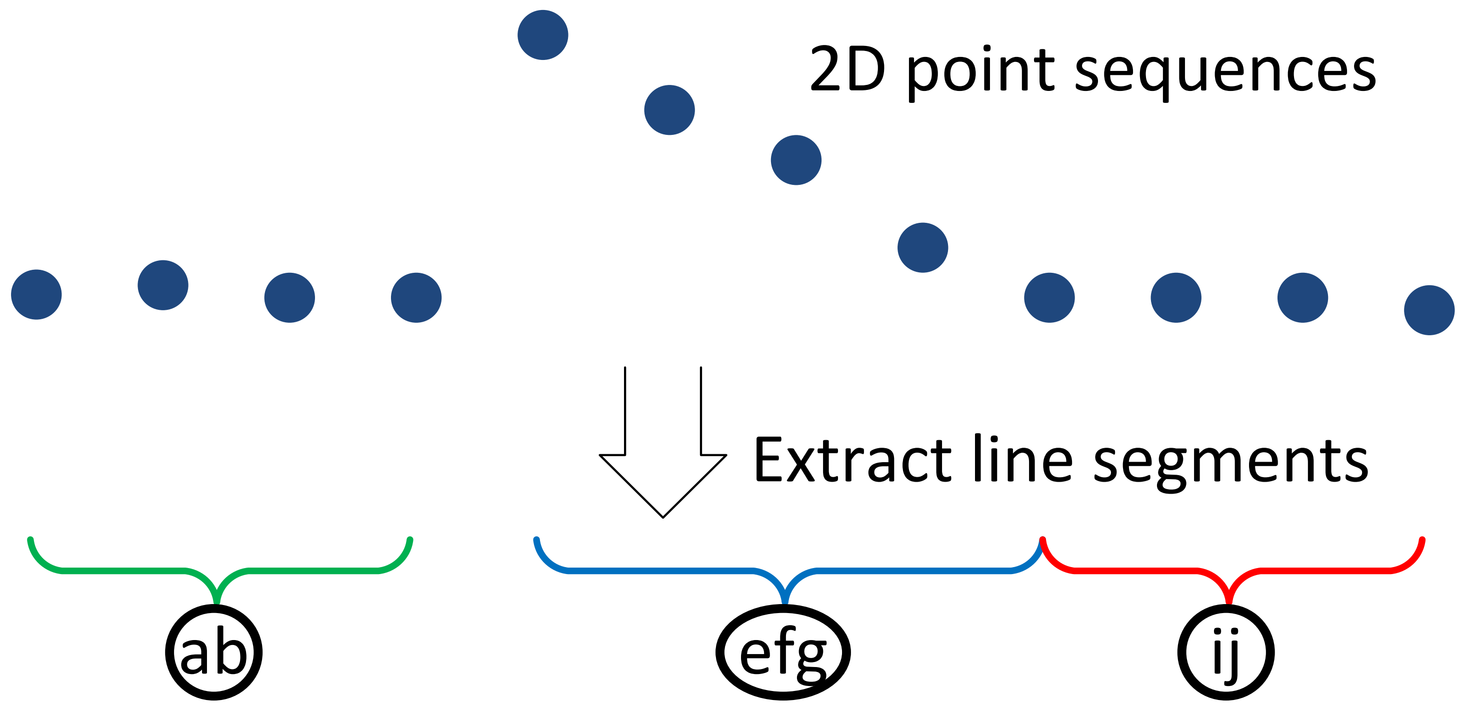
- Analogy to Line Regression (Nguyen et al. 2005; April Robotics Toolkit, 2010)
  - Exploit the neighborhood information given by the order of points



- Analogy to Line Regression (Nguyen et al. 2005; April Robotics Toolkit, 2010)
  - Exploit the neighborhood information given by the order of points

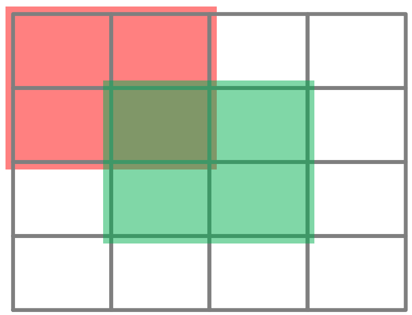


- Analogy to Line Regression (Nguyen et al. 2005; April Robotics Toolkit, 2010)
  - Exploit the neighborhood information given by the order of points

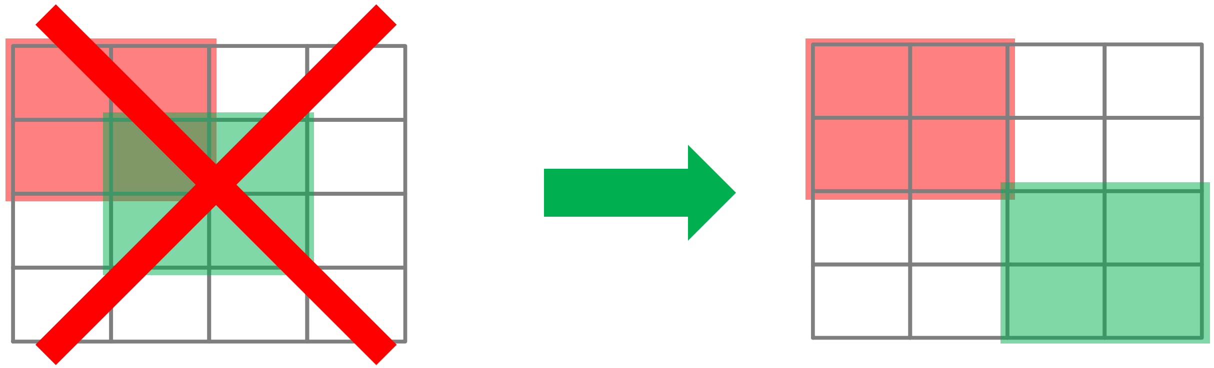


- Non-trivial Generalization to 3D

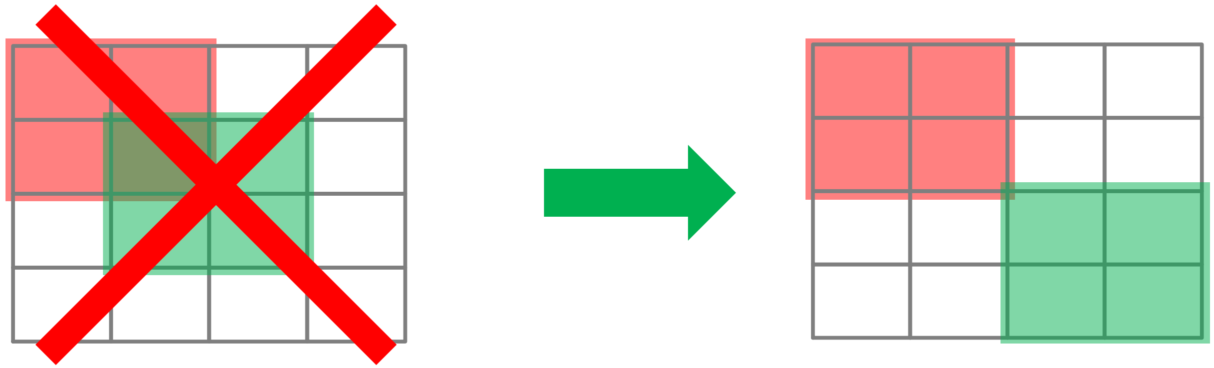
- Non-trivial Generalization to 3D
  - None-overlapping nodes



- Non-trivial Generalization to 3D
  - None-overlapping nodes

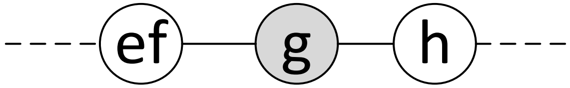


- Non-trivial Generalization to 3D
  - None-overlapping nodes

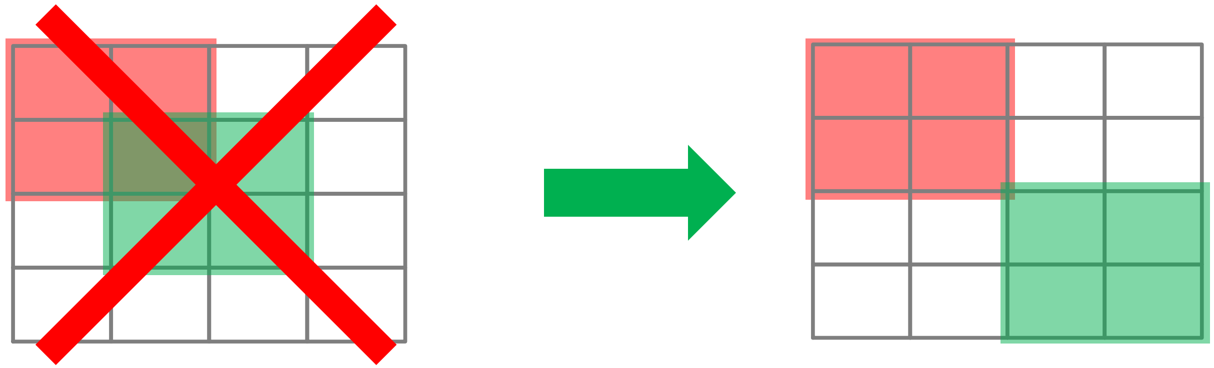


  - Number of merging attempts

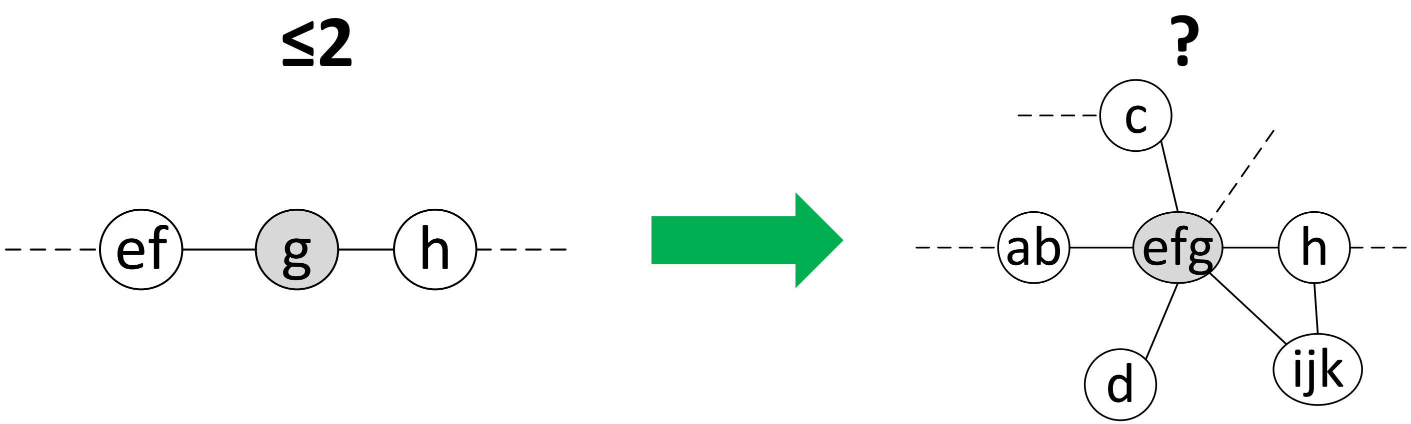
≤2



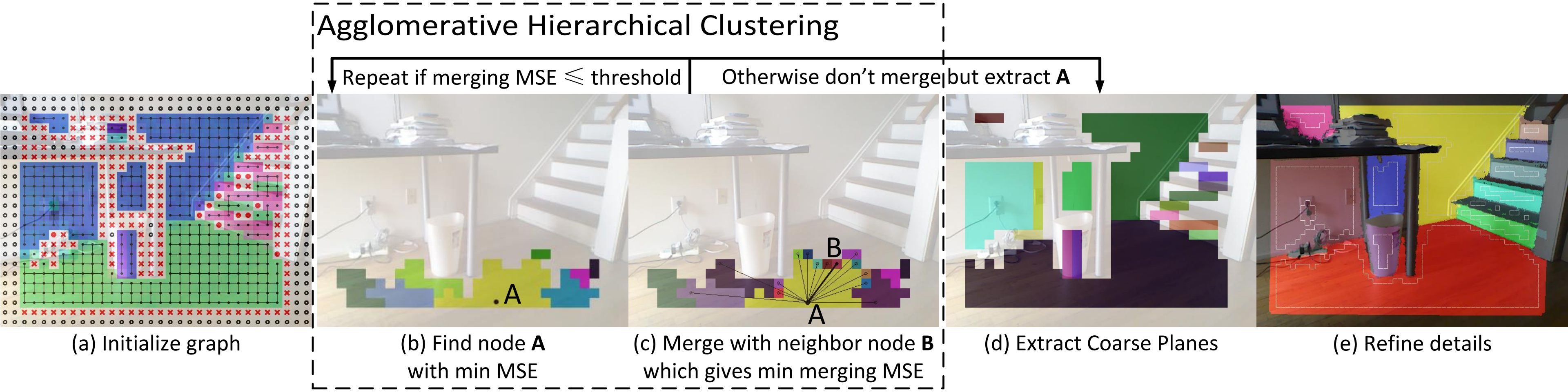
- Non-trivial Generalization to 3D
  - None-overlapping nodes



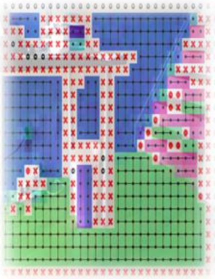
  - Number of merging attempts



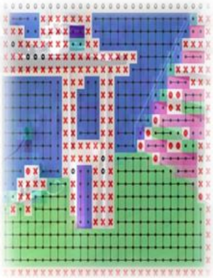
- Algorithm Overview



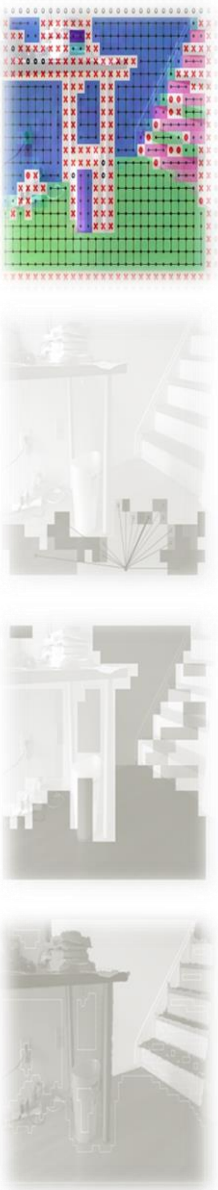
- Algorithm Overview



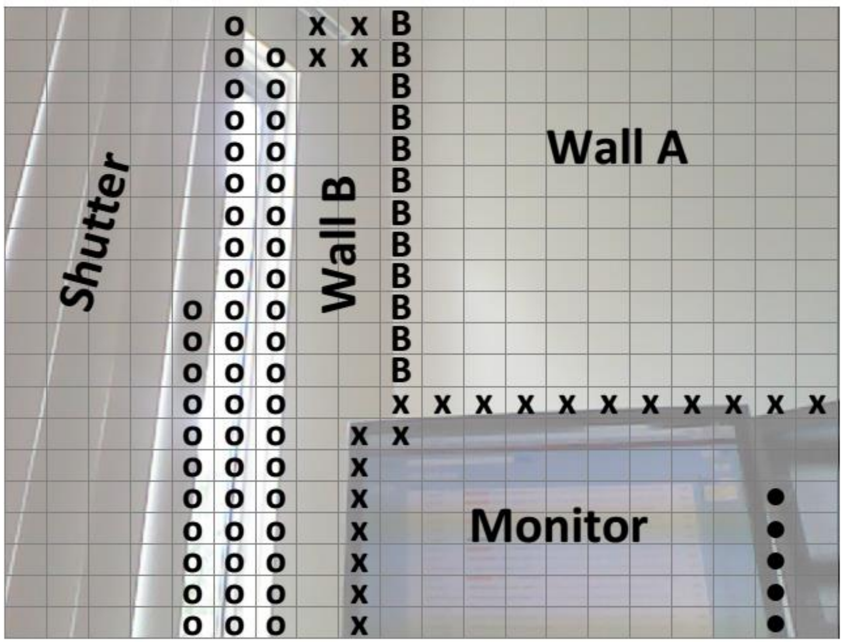
- Graph Initialization

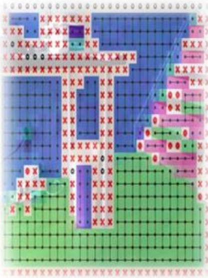


- Graph Initialization
  - Non-overlapping node initialization

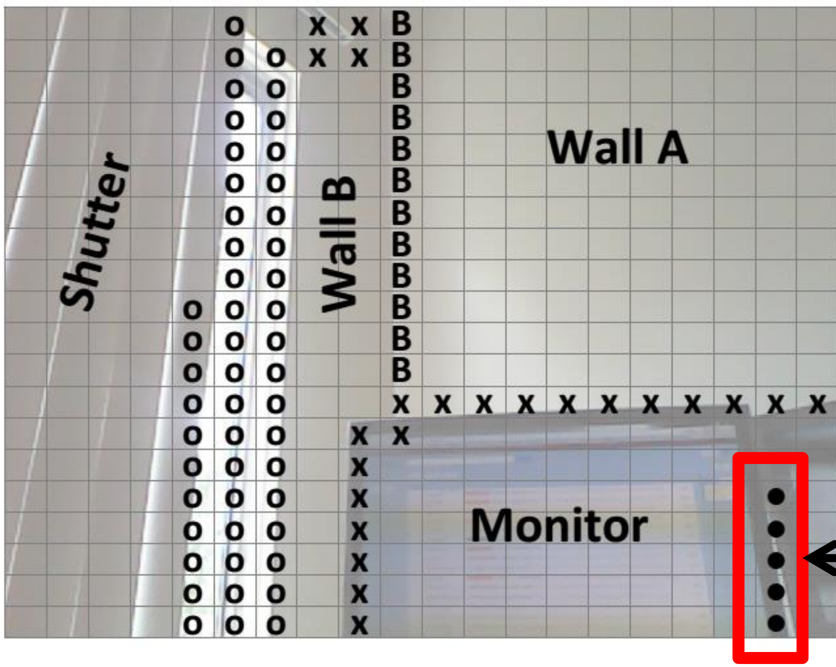


- Graph Initialization
  - Non-overlapping node initialization
  - Rejecting “bad” nodes

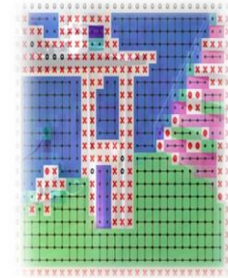




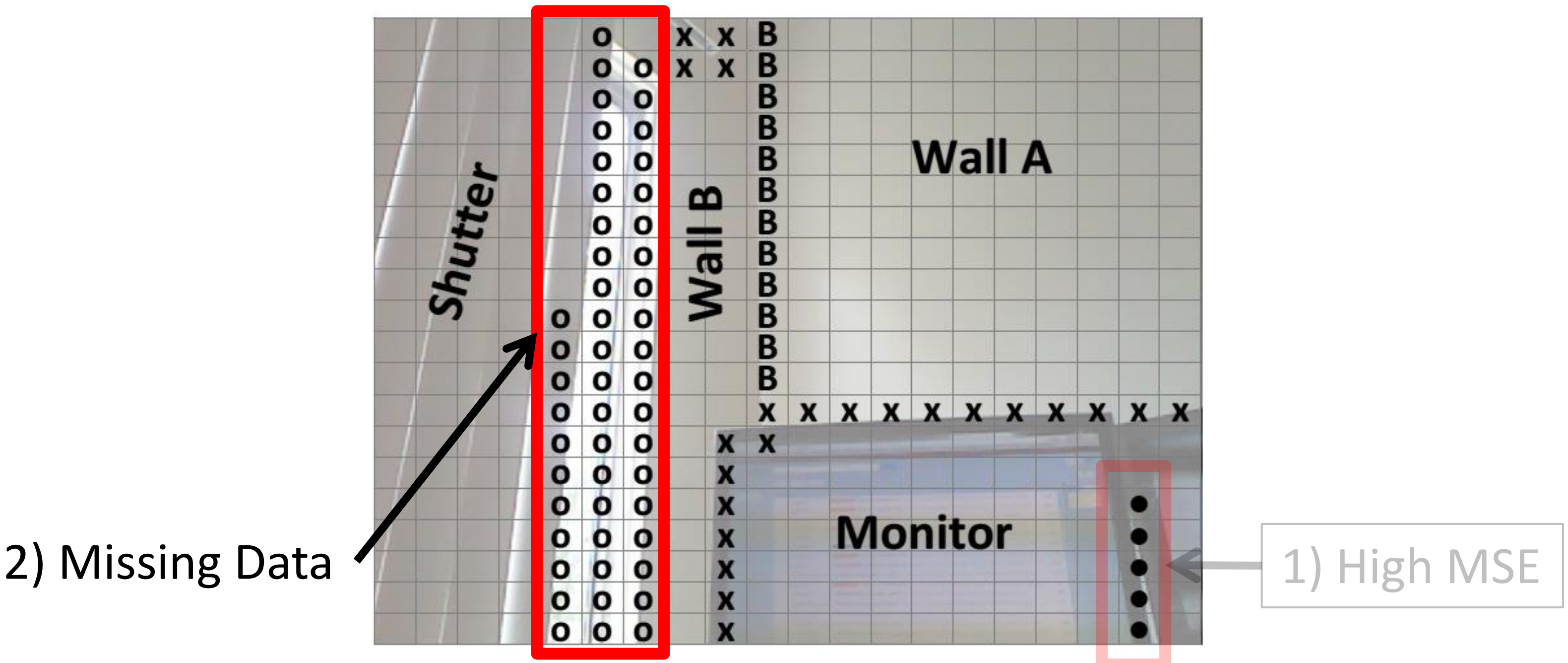
- Graph Initialization
  - Non-overlapping node initialization
  - Rejecting “bad” nodes

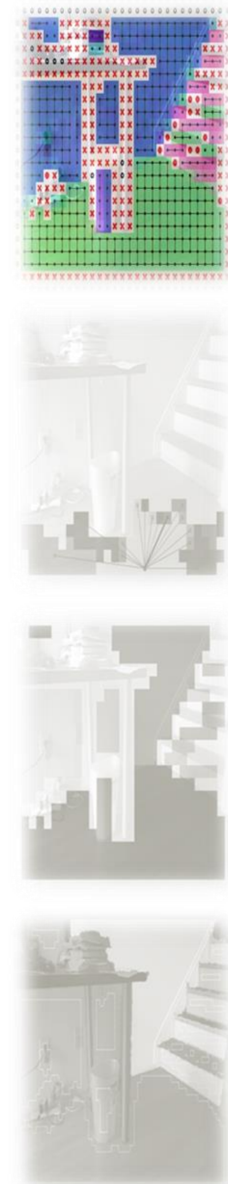


1) High MSE

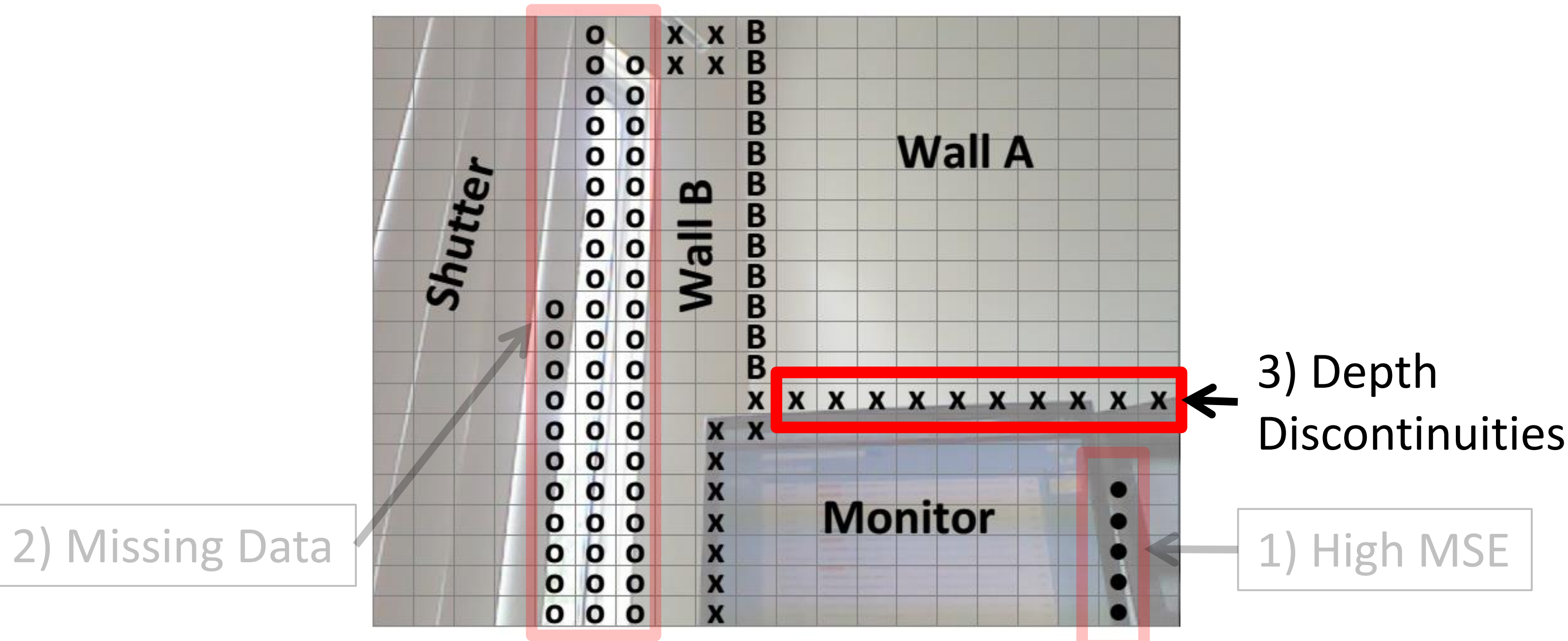


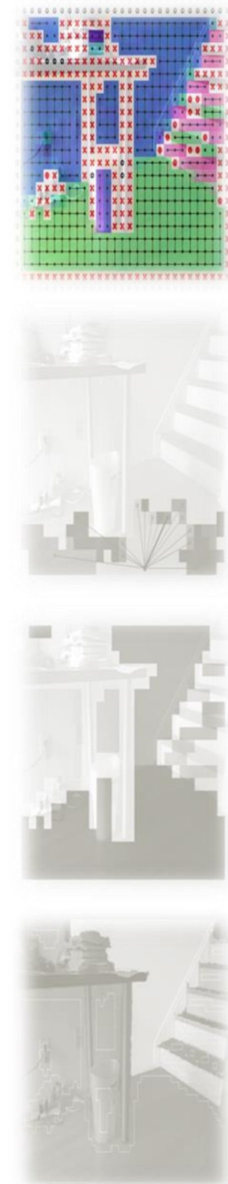
- Graph Initialization
  - Non-overlapping node initialization
  - Rejecting “bad” nodes



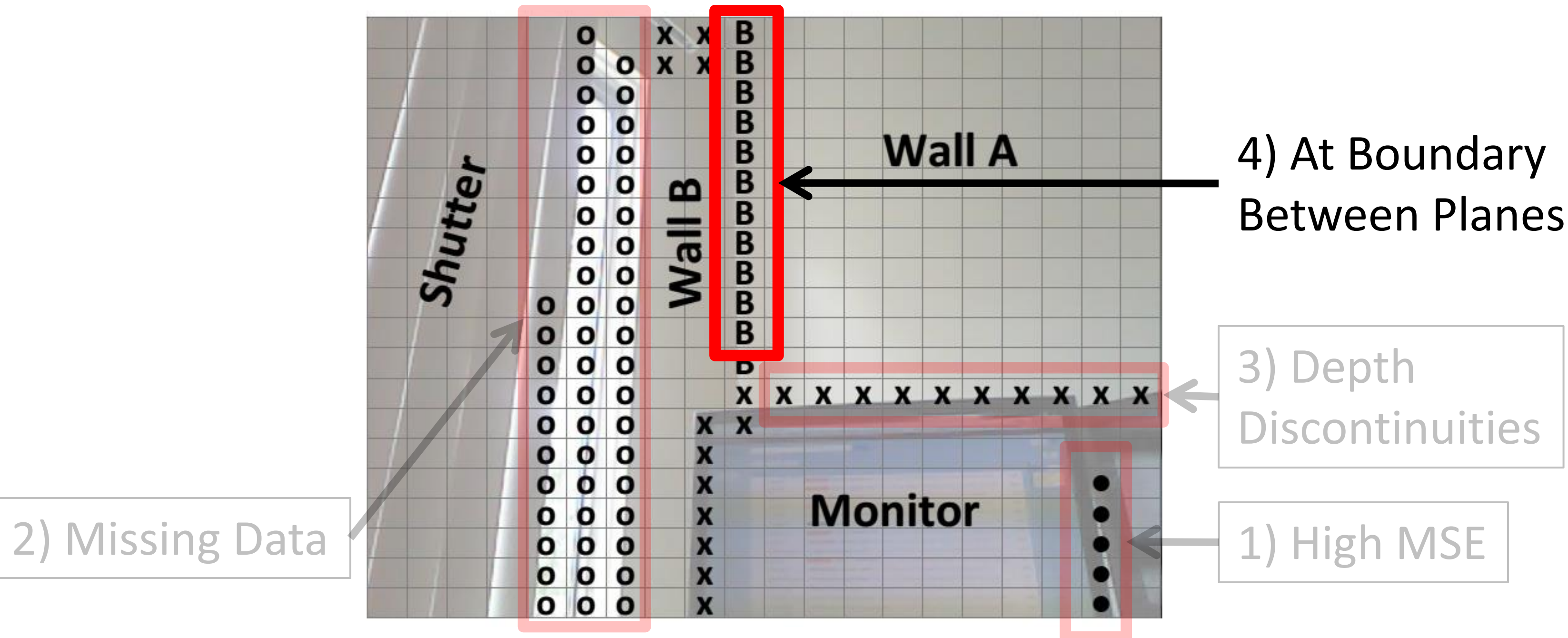


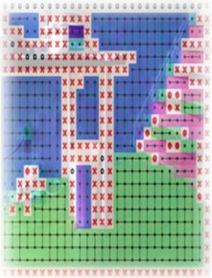
- Graph Initialization
  - Non-overlapping node initialization
  - Rejecting “bad” nodes



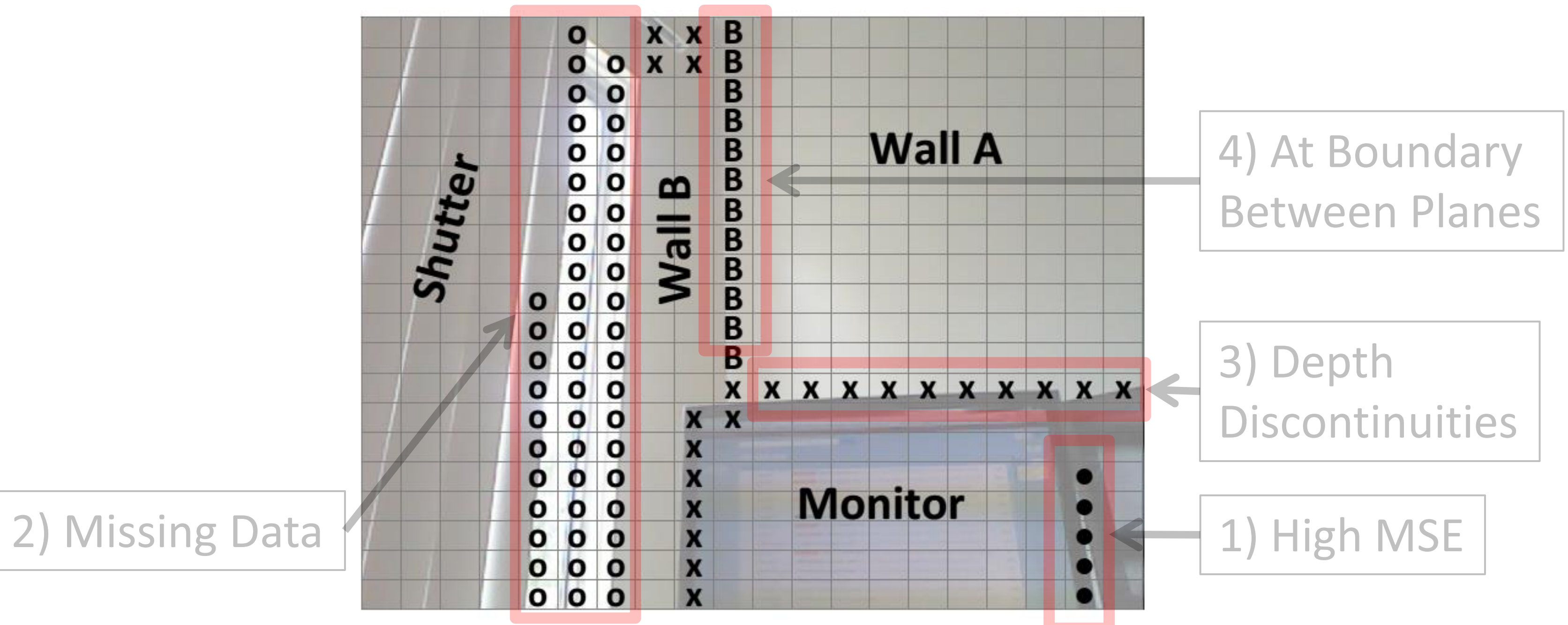


- Graph Initialization
  - Non-overlapping node initialization
  - Rejecting "bad" nodes



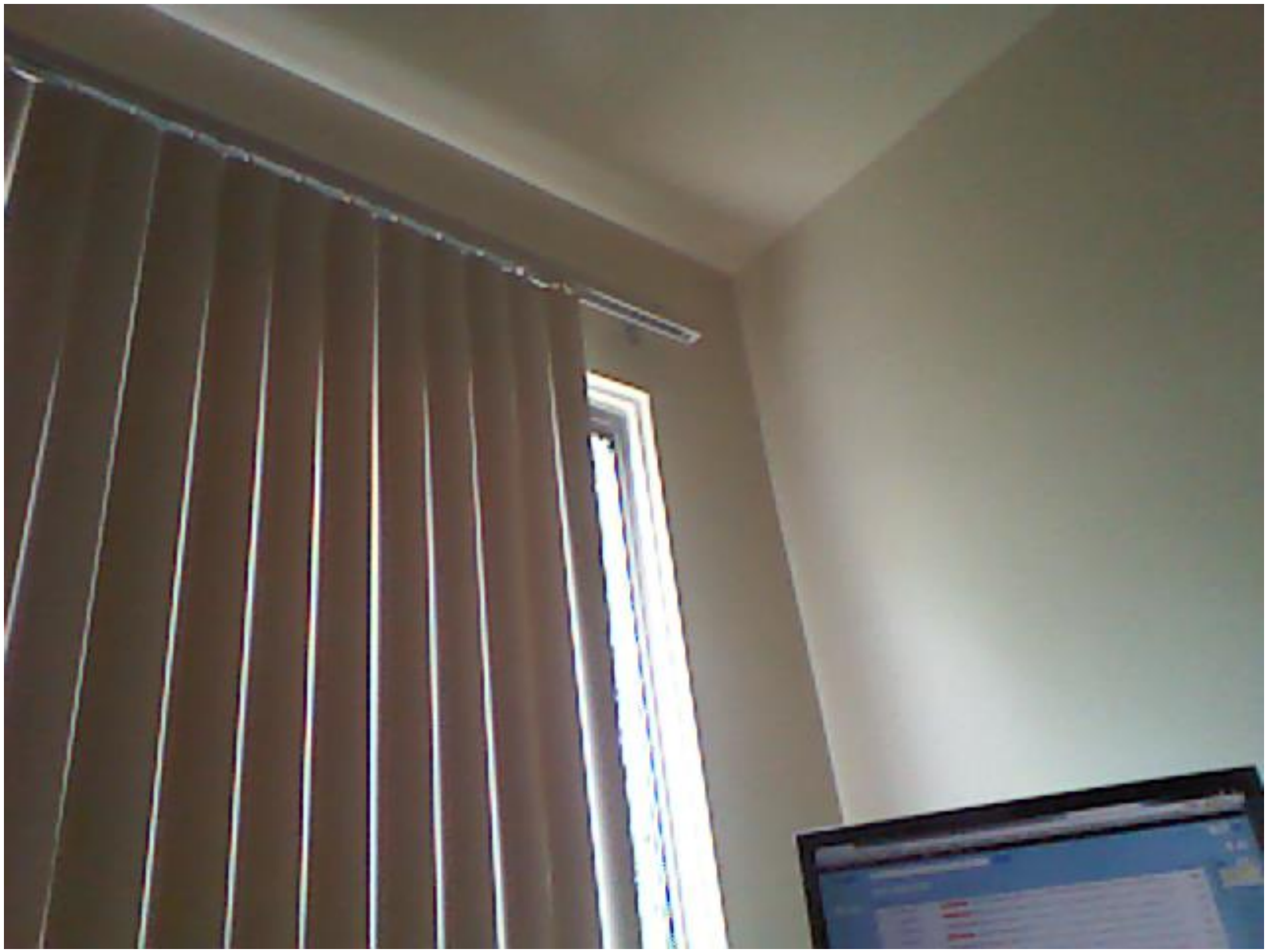


- Graph Initialization
  - Non-overlapping node initialization
  - Rejecting "bad" nodes

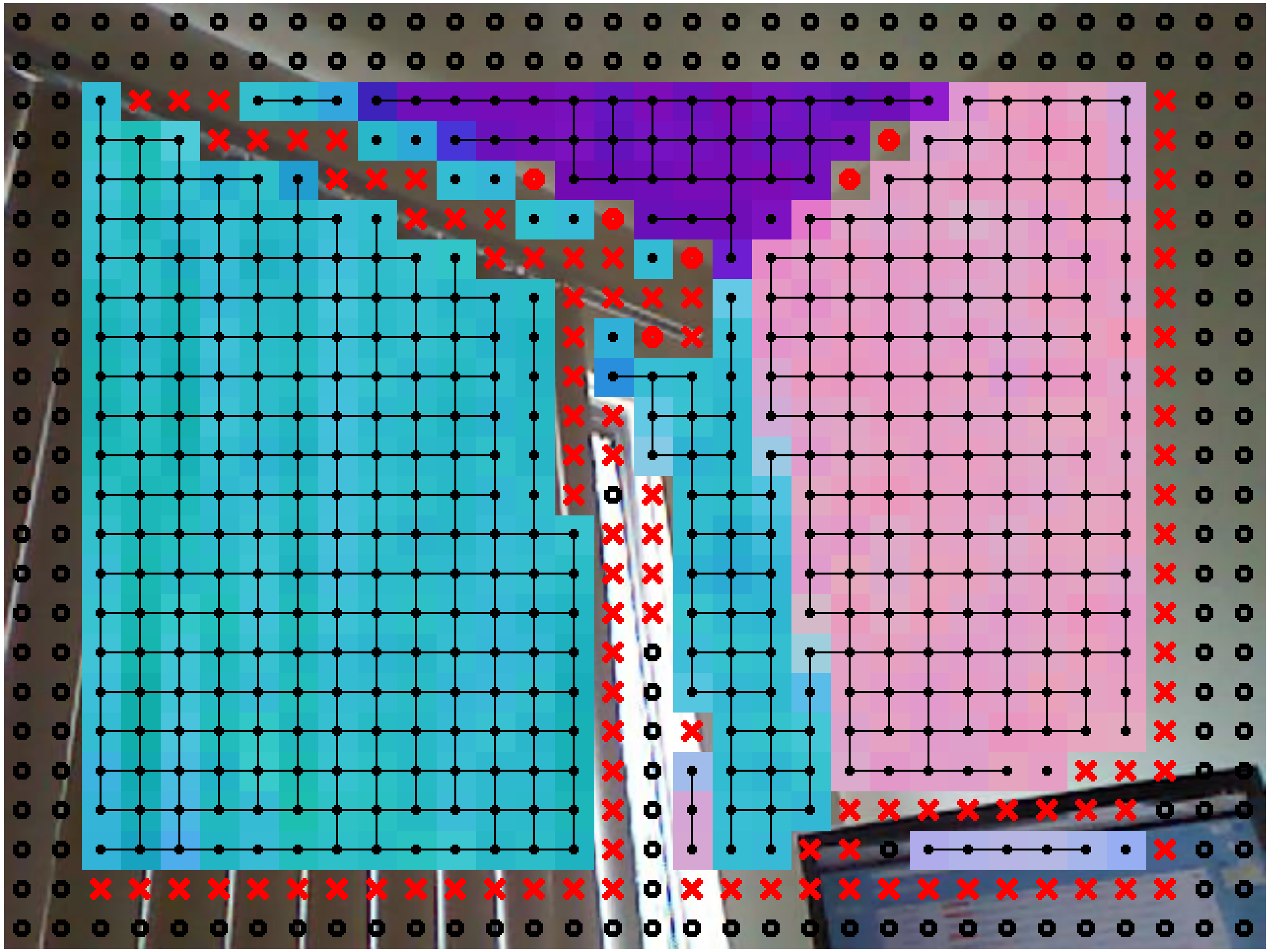


  - Good! Avoid per-point normal estimation

- Agglomerative Hierarchical Clustering

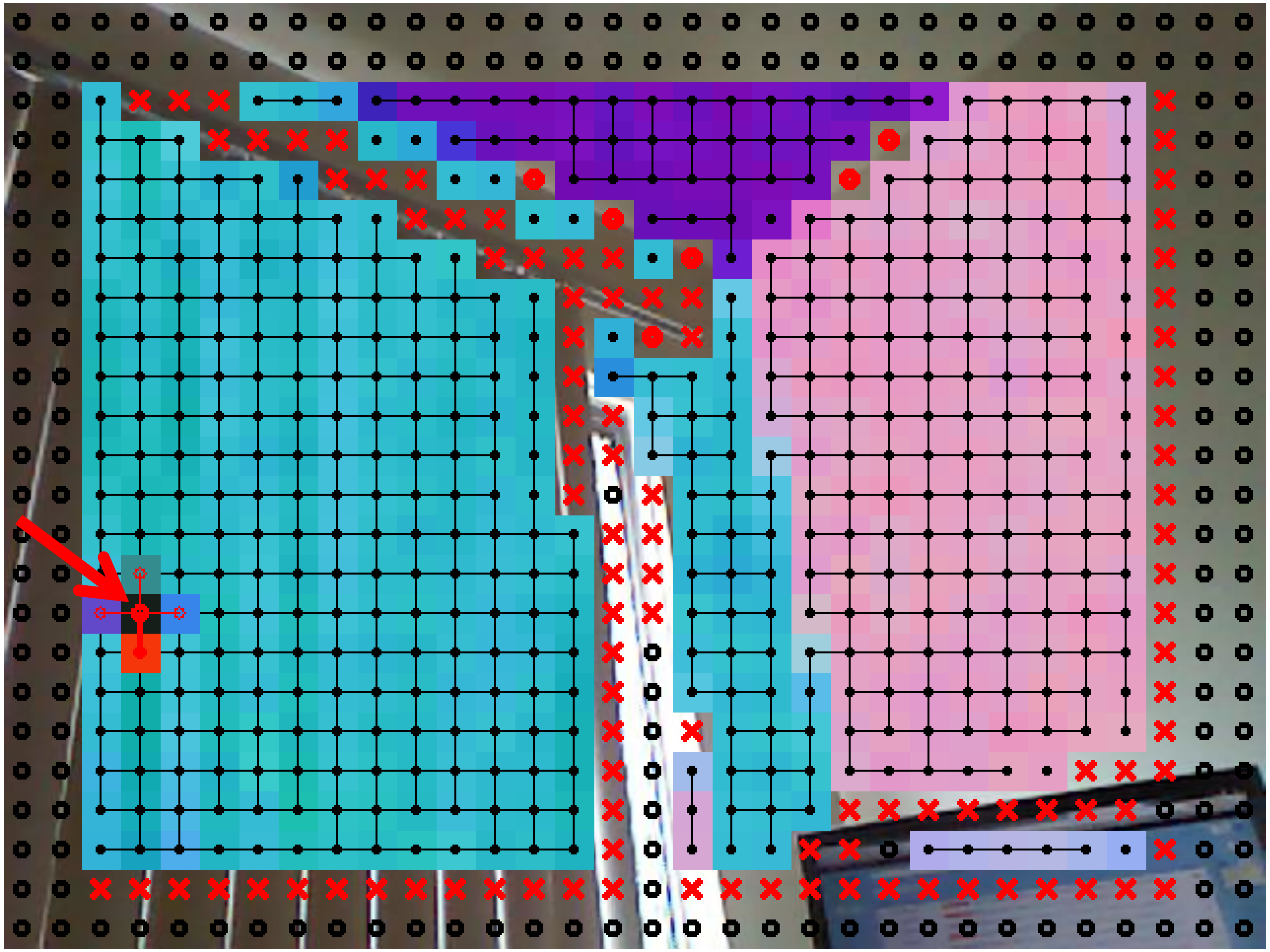


- Agglomerative Hierarchical Clustering



- Agglomerative Hierarchical Clustering

1 Cluster Step(s)

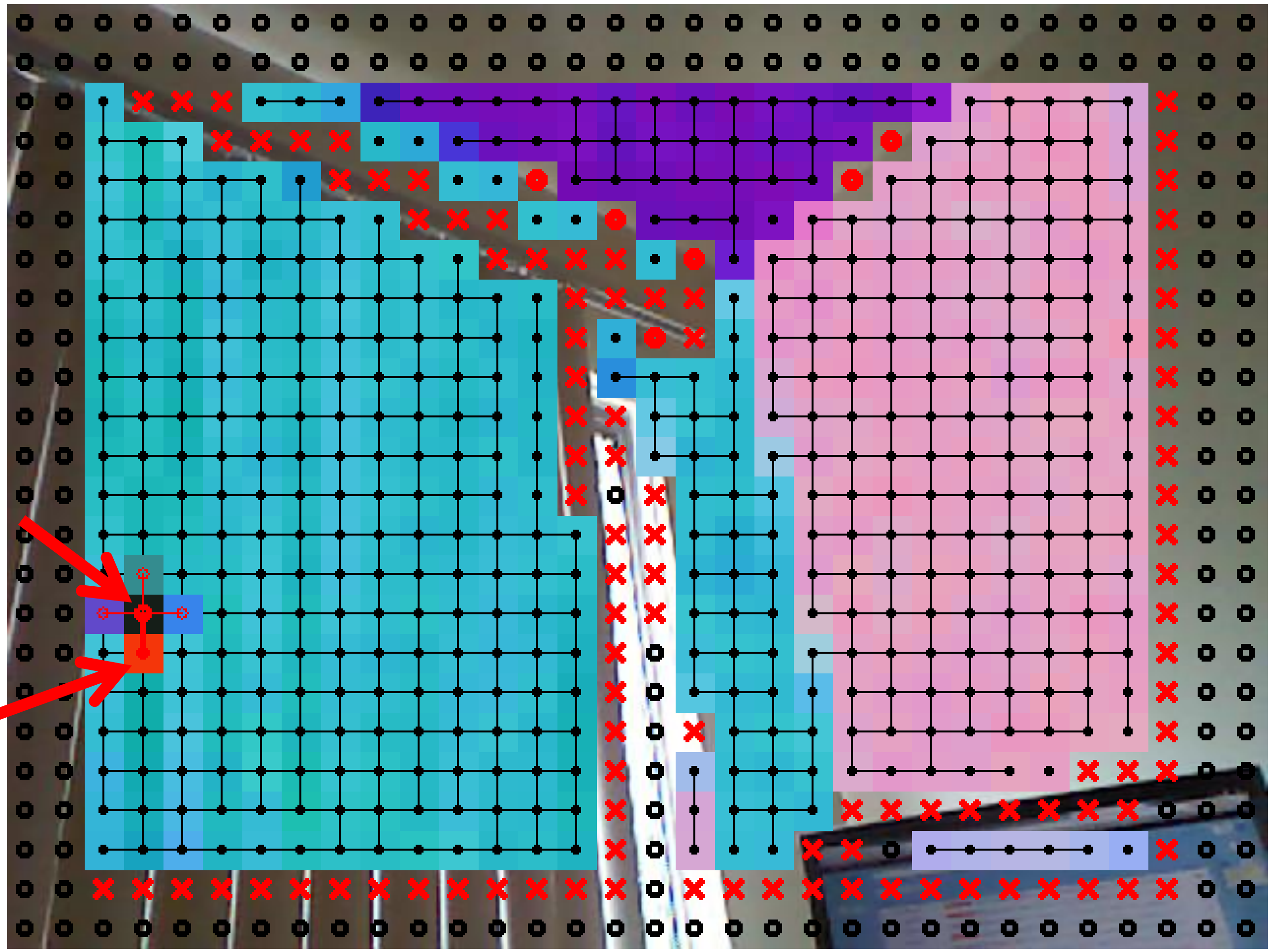


Current min MSE node

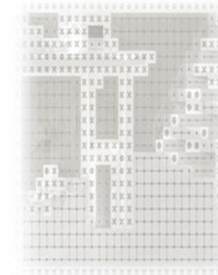
- Agglomerative Hierarchical Clustering

1 Cluster Step(s)

Current min MSE node

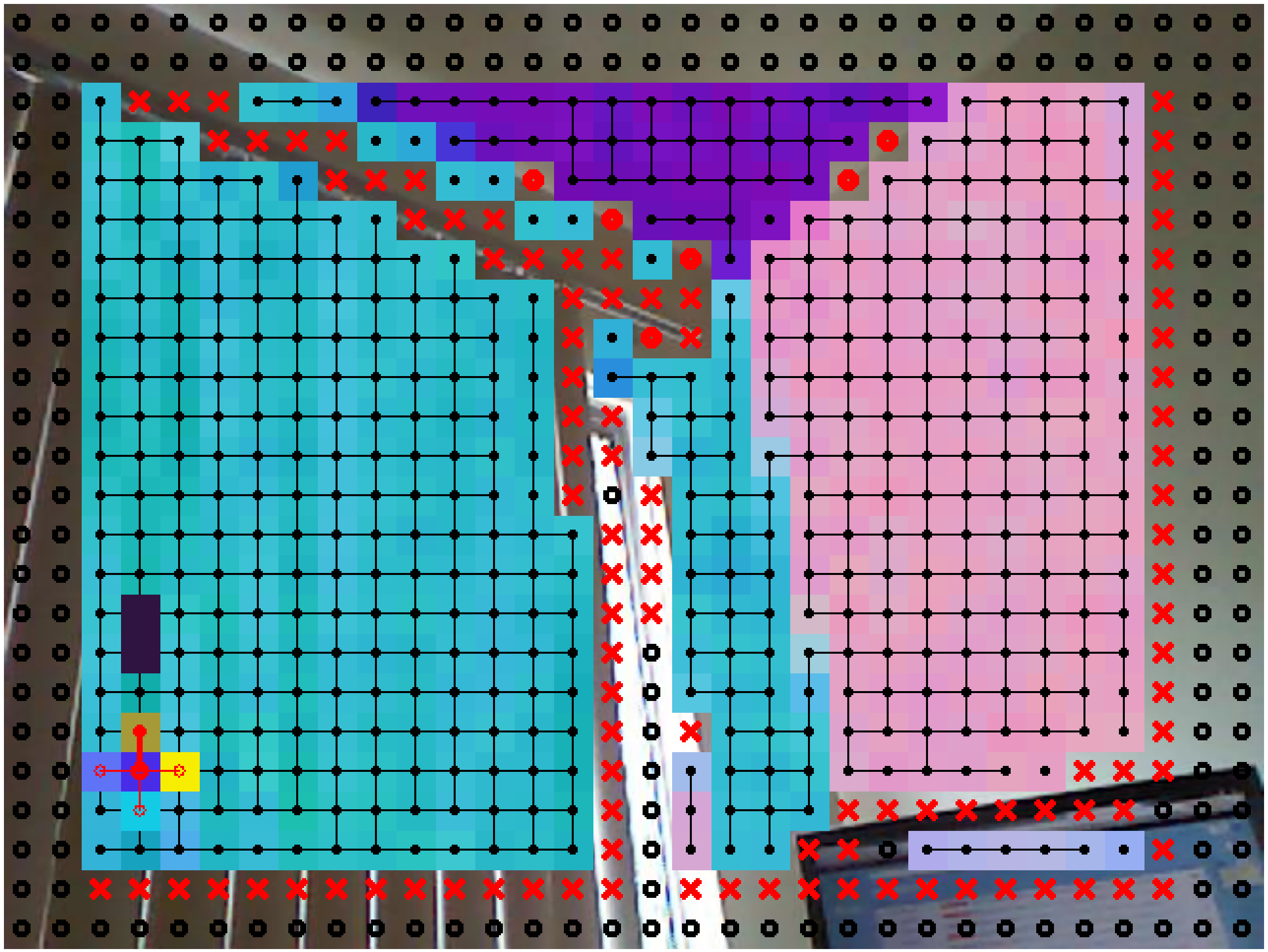


Best node to merge



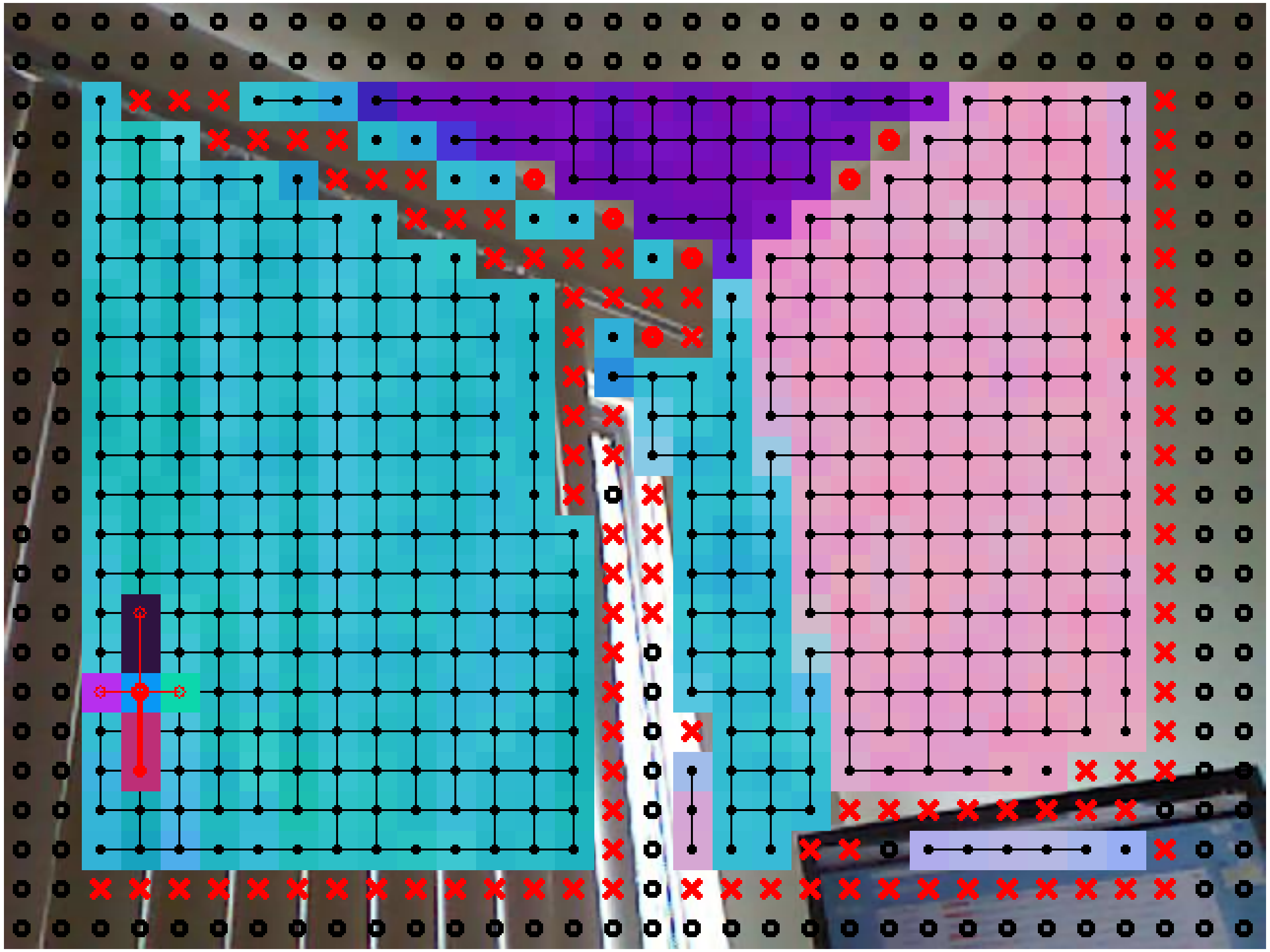

- Agglomerative Hierarchical Clustering

2 Cluster Step(s)



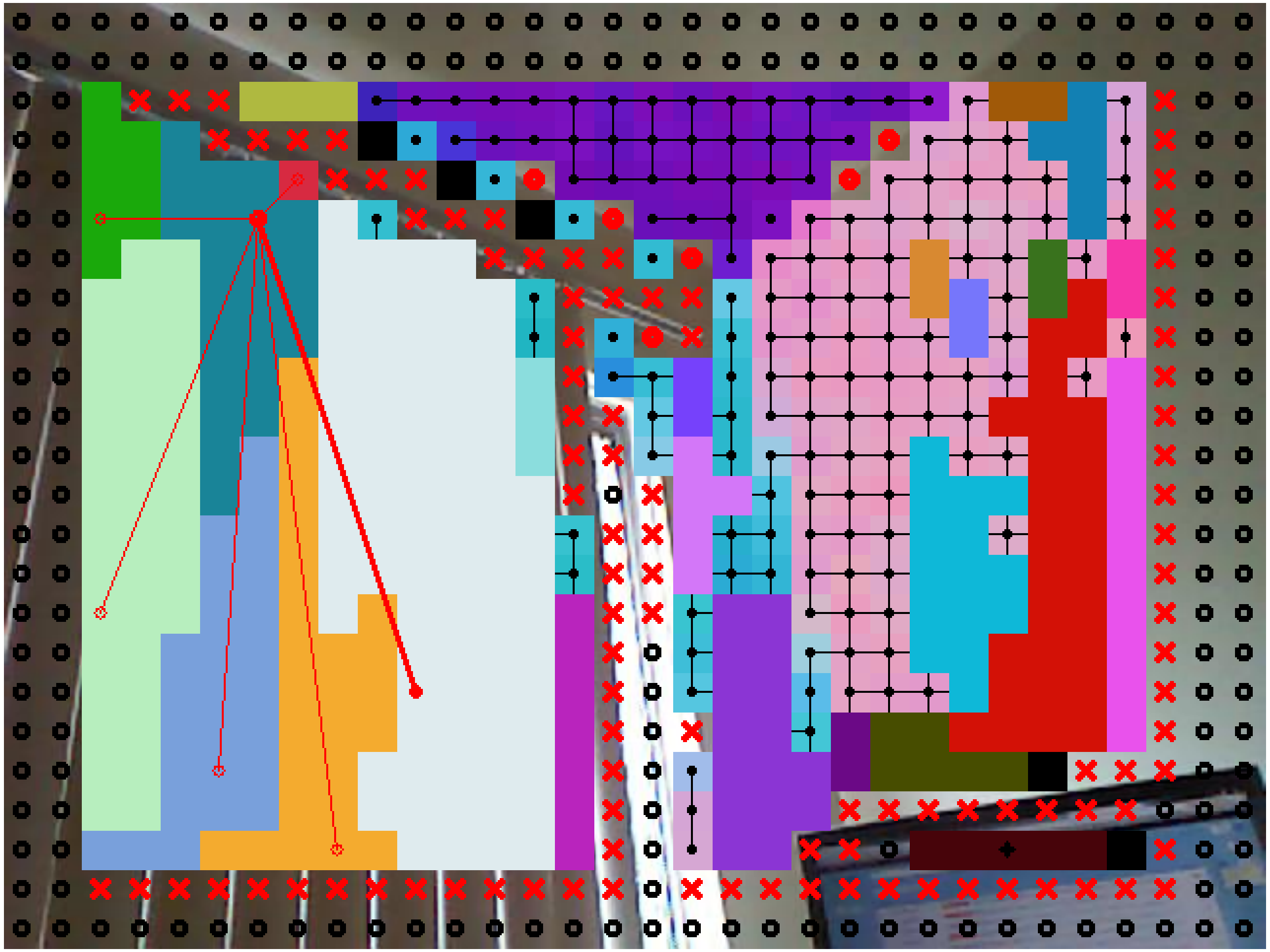
- Agglomerative Hierarchical Clustering

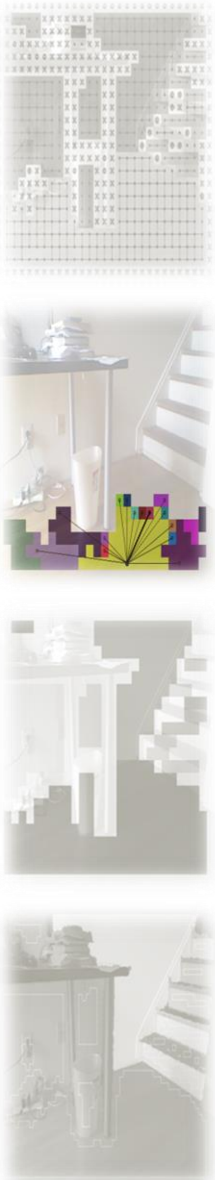
3 Cluster Step(s)



- Agglomerative Hierarchical Clustering

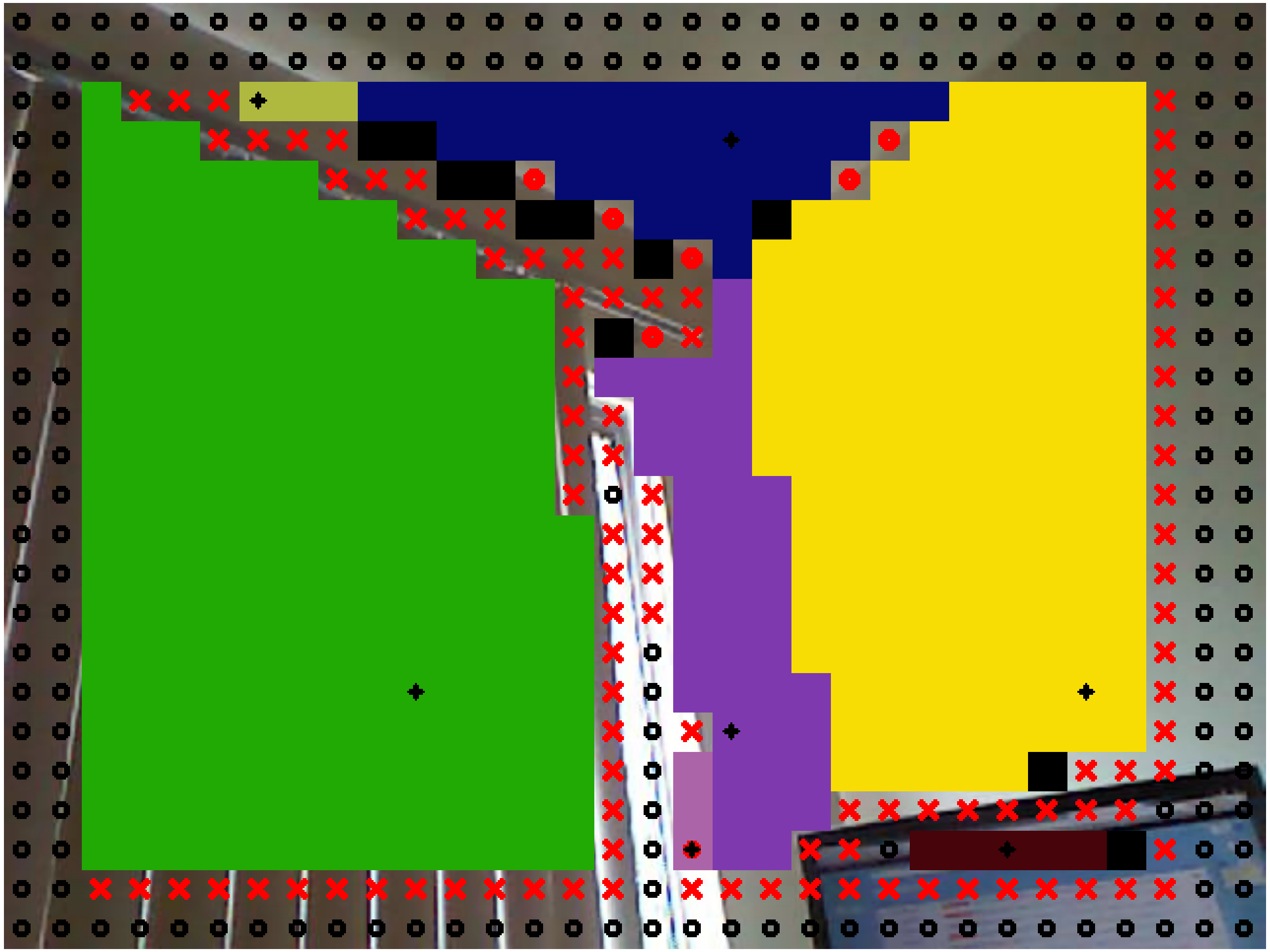
300 Cluster Step(s)



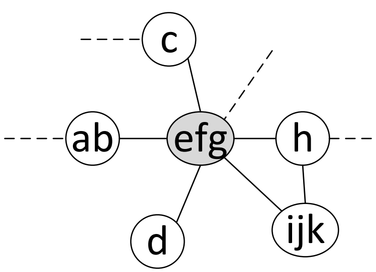


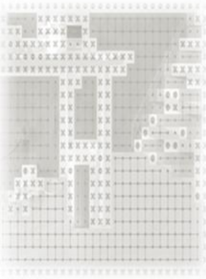
- Agglomerative Hierarchical Clustering

472 Cluster Step(s)

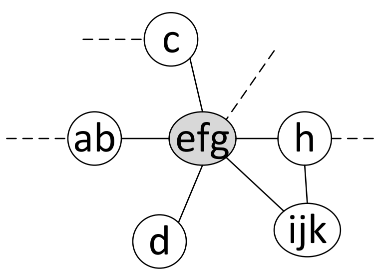


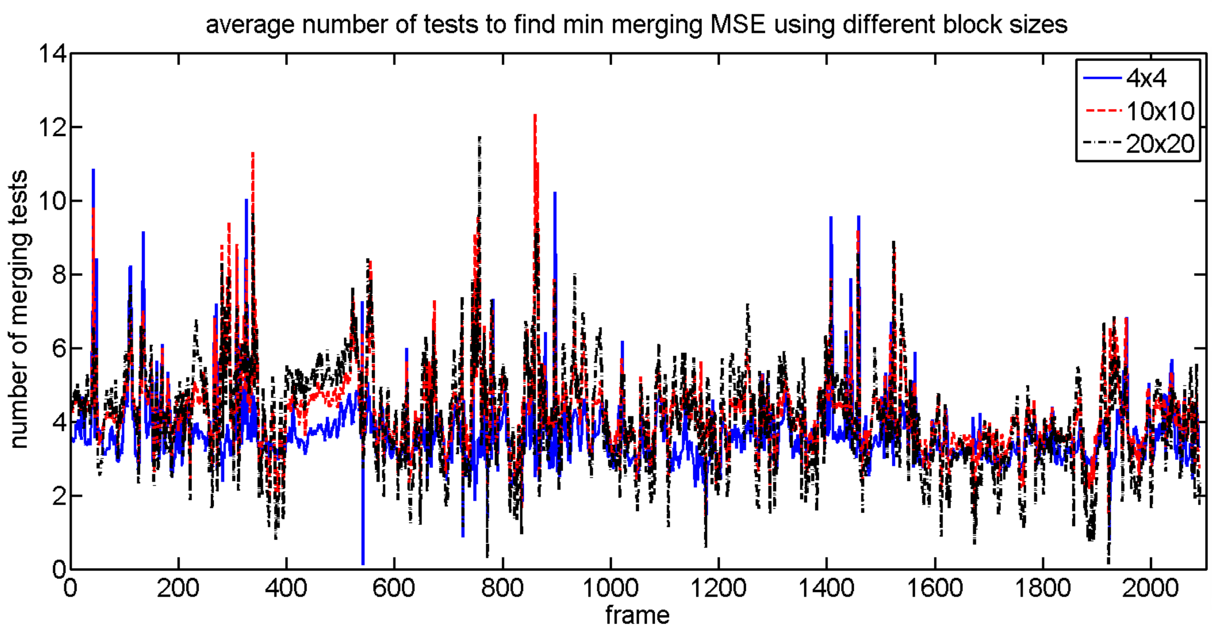
- Average Number of Merging Attempts



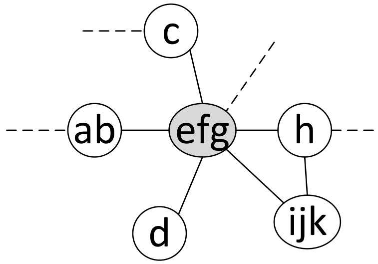


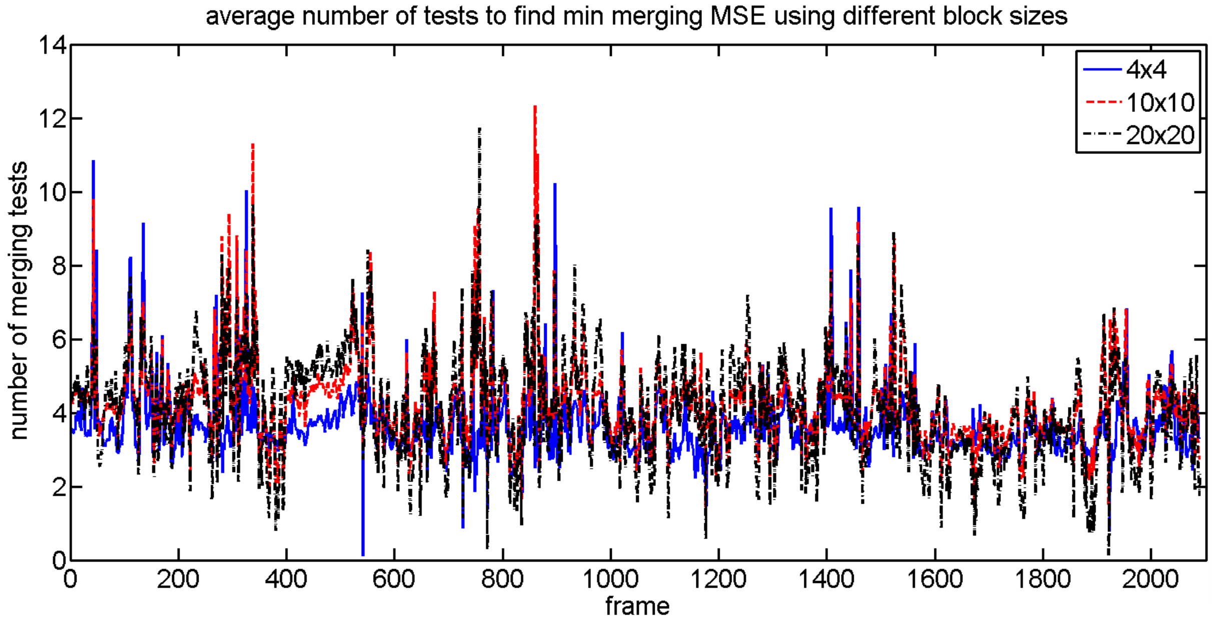
- Average Number of Merging Attempts
  - Small irrespective of initial number of nodes



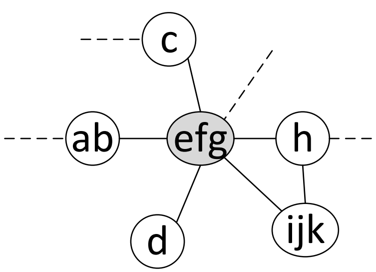


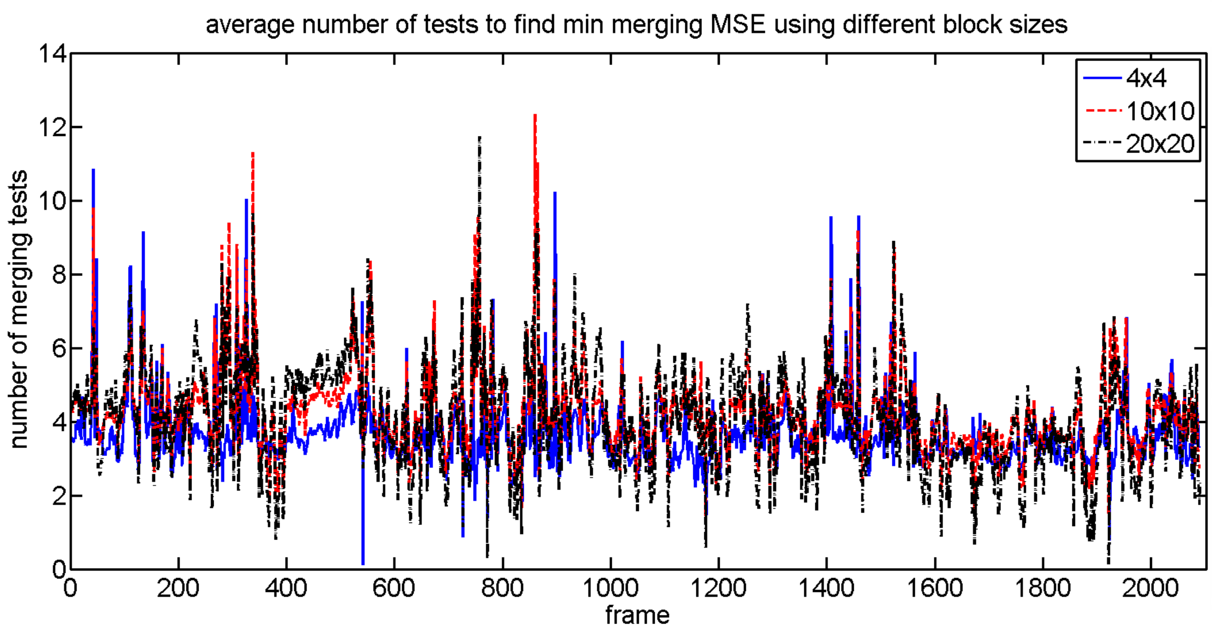
- Average Number of Merging Attempts
  - Small irrespective of initial number of nodes
  - Planar graph! Average node degree < 6

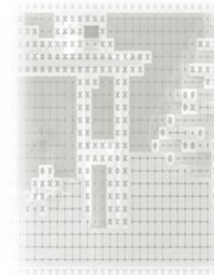


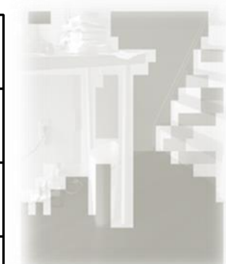


- Average Number of Merging Attempts
  - Small irrespective of initial number of nodes
  - Planar graph! Average node degree < 6
  - Merging is empirically a constant-time operation
    - O(nlogn), only arise from maintaining the min-heap

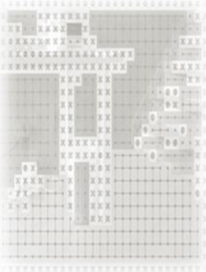




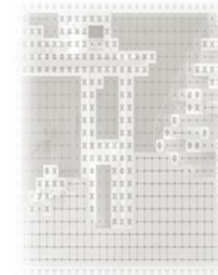




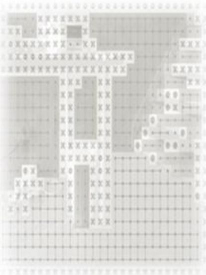
- Implementation Details



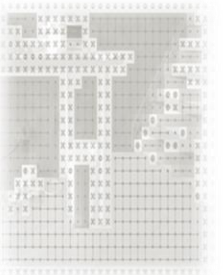

- Implementation Details
  - Disjoint set



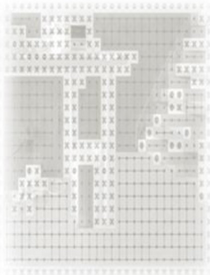

- Implementation Details
  - Disjoint set
  - Min-heap



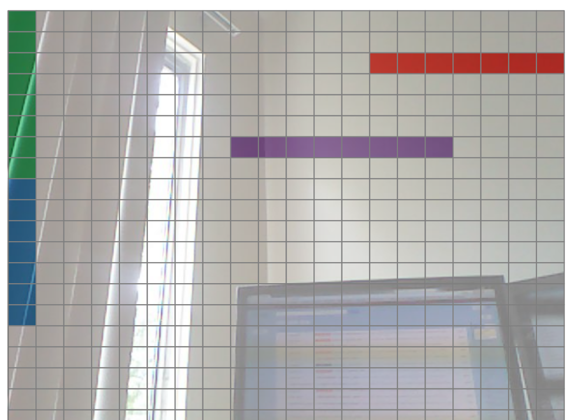
- Implementation Details
  - Disjoint set
  - Min-heap
  - Second-order statistics



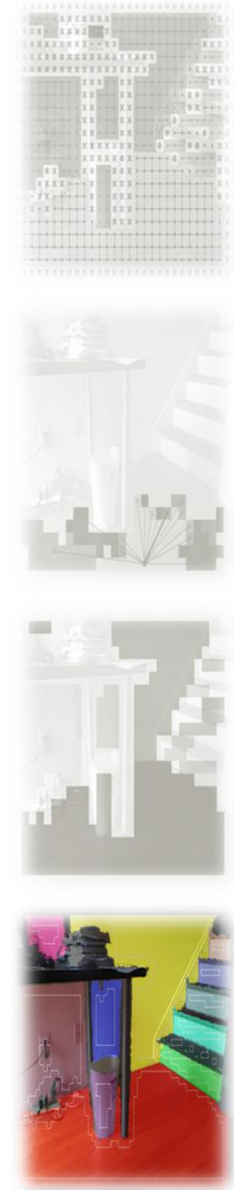

- Implementation Details
  - Disjoint set
  - Min-heap
  - Second-order statistics
  - Depth discontinuity/MSE threshold
    (Holzer et al. IROS 2012; Khoshelham & Elberink, 2012)



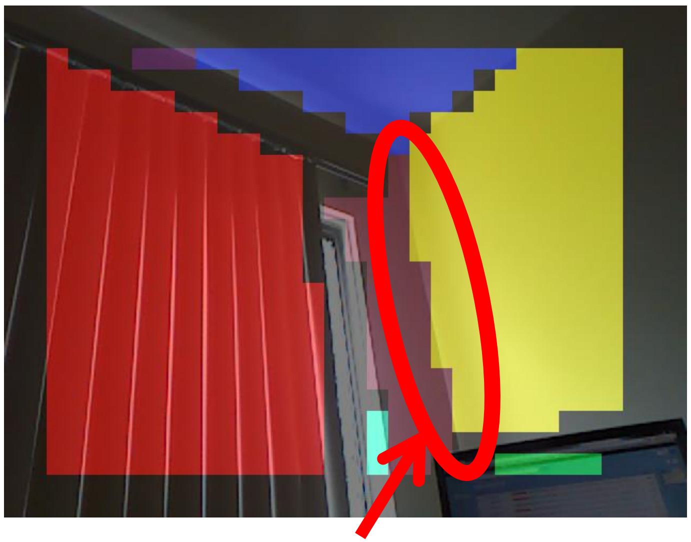

- Implementation Details
  - Disjoint set
  - Min-heap
  - Second-order statistics
  - Depth discontinuity/MSE threshold
    (Holzer et al. IROS 2012; Khoshelham & Elberink, 2012)
  - Avoid strip-like initial node shape



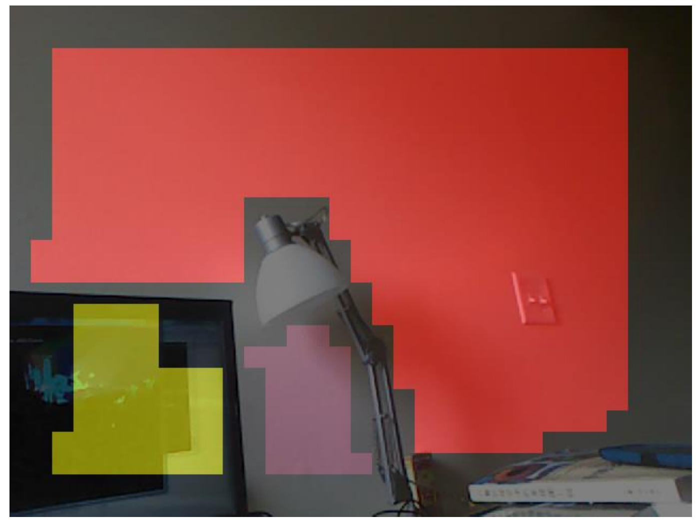
- Segmentation Refinement

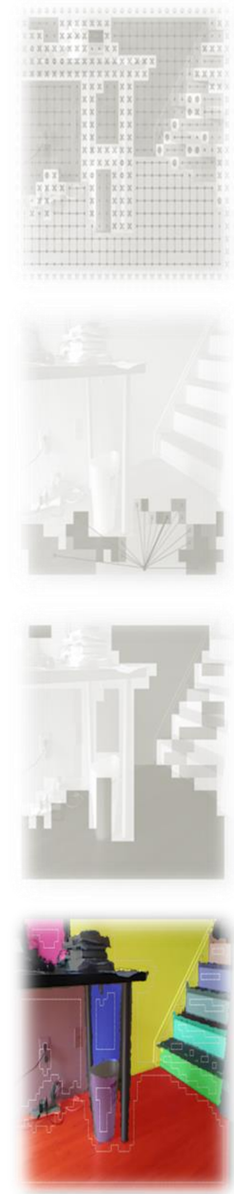


- Segmentation Refinement
  - Artifacts

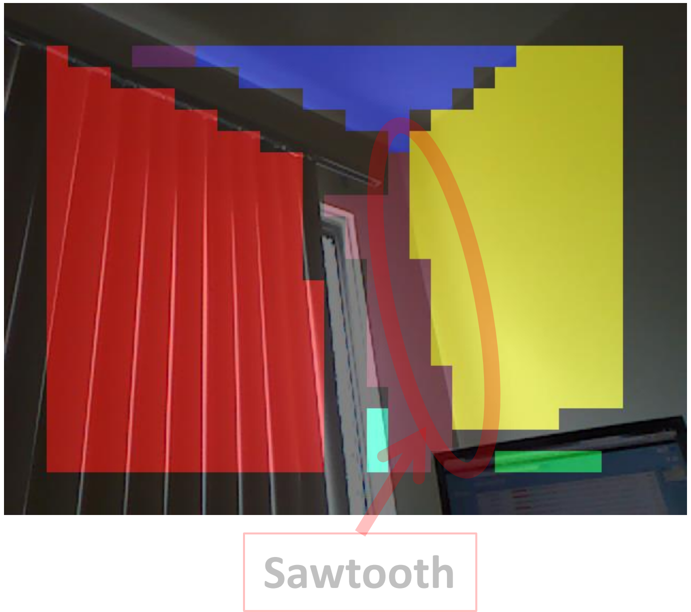


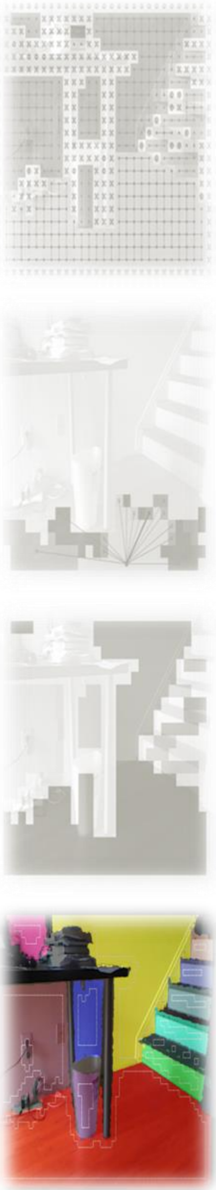
Sawtooth



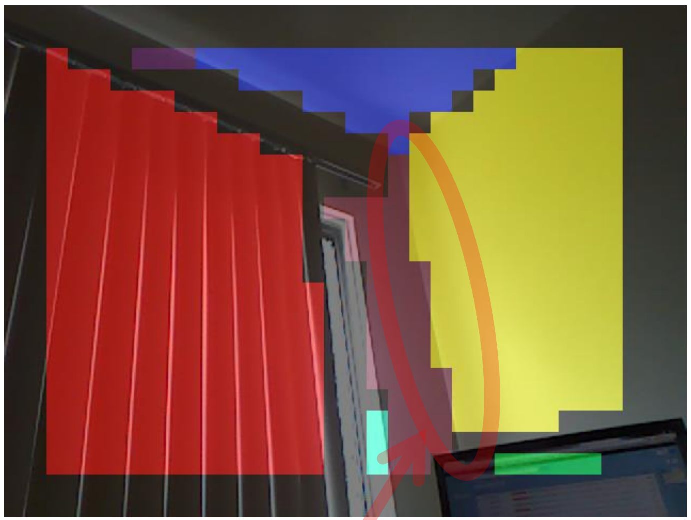


- Segmentation Refinement
  - Artifacts



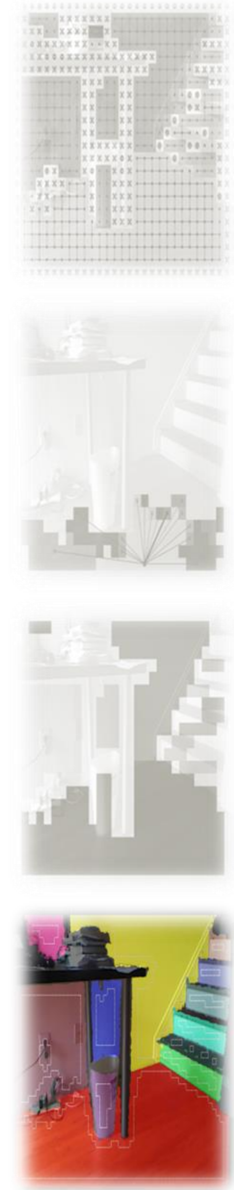


- Segmentation Refinement
  - Artifacts

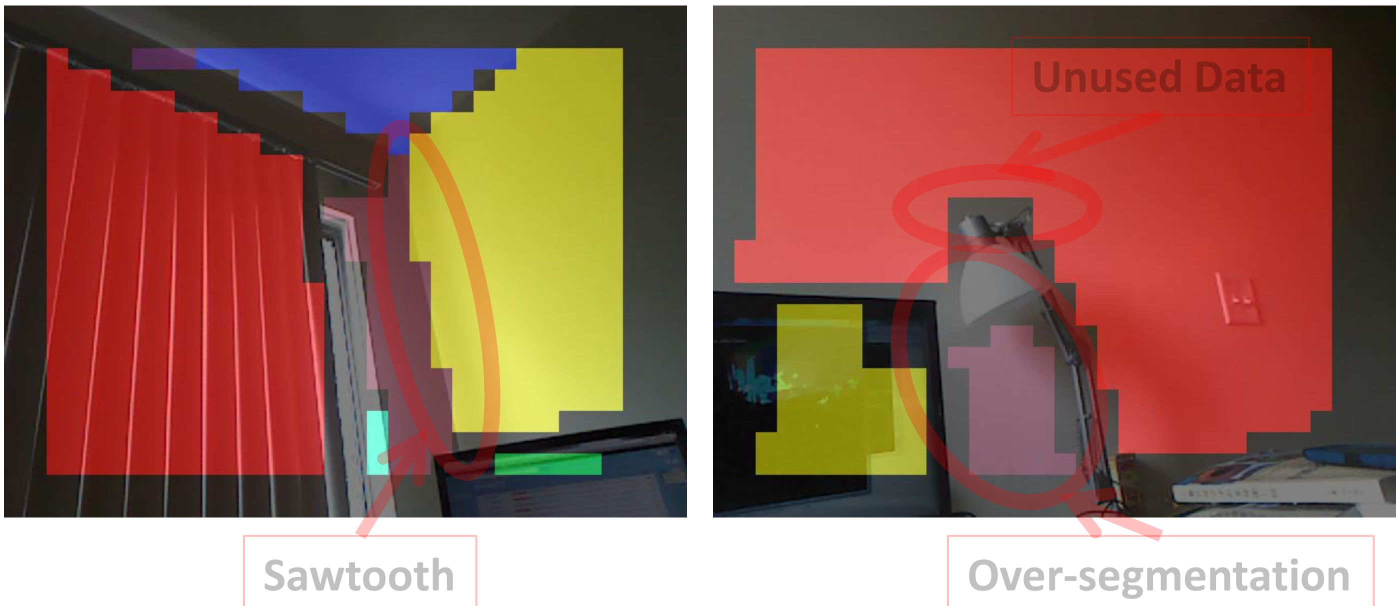


Sawtooth

Over-segmentation

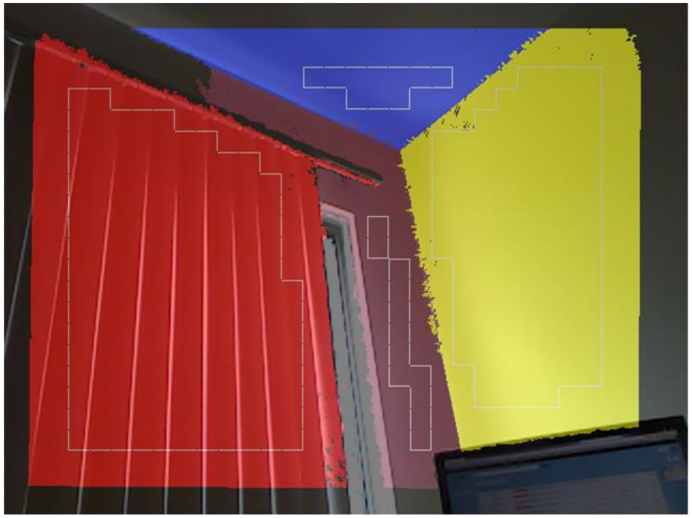


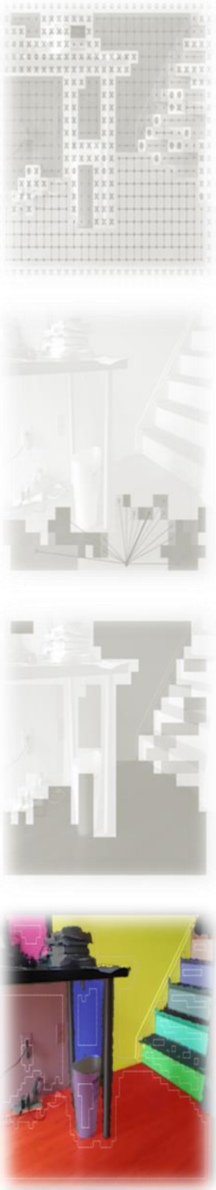
- Segmentation Refinement
  - Artifacts



  - Pixel-wise region-grow refinement
    - Only check boundary blocks and points

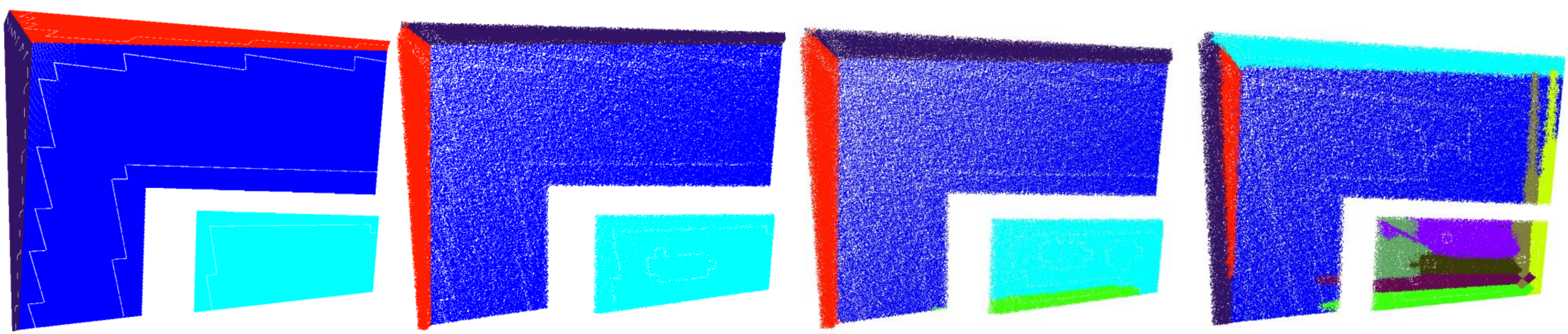
- Segmentation Refinement
  - Artifacts





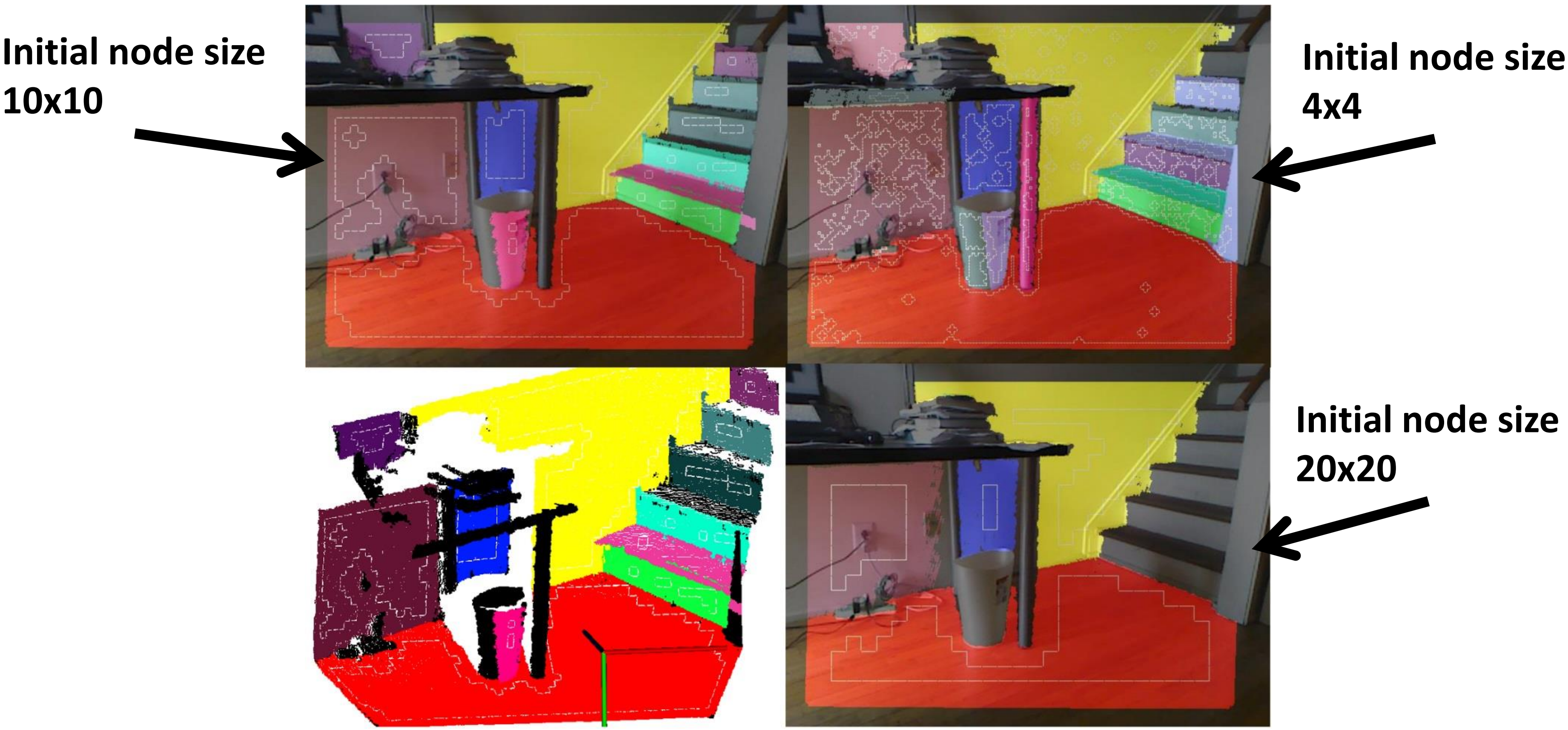
  - Pixel-wise region-grow refinement
    - Only check boundary blocks and points

- Simulated Data
  - Robustness to uniformly distributed depth noise (Georgiev et al., IROS 2011)
  - Noise magnitude E = 0, 10, …, 200mm
  - Ground truth depth ranges from 1396mm to 3704mm

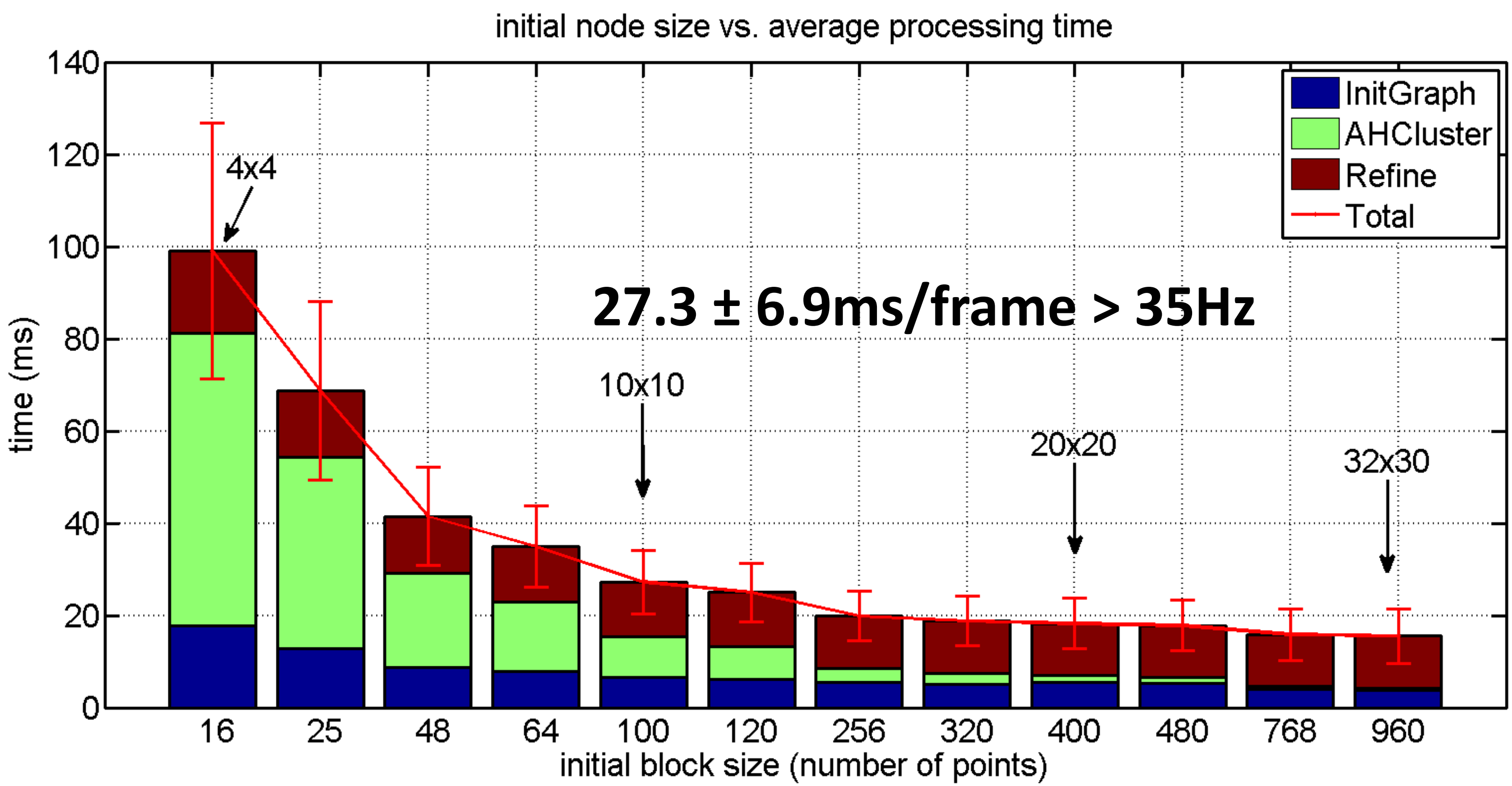


E=0 E=100 E=150 E=200

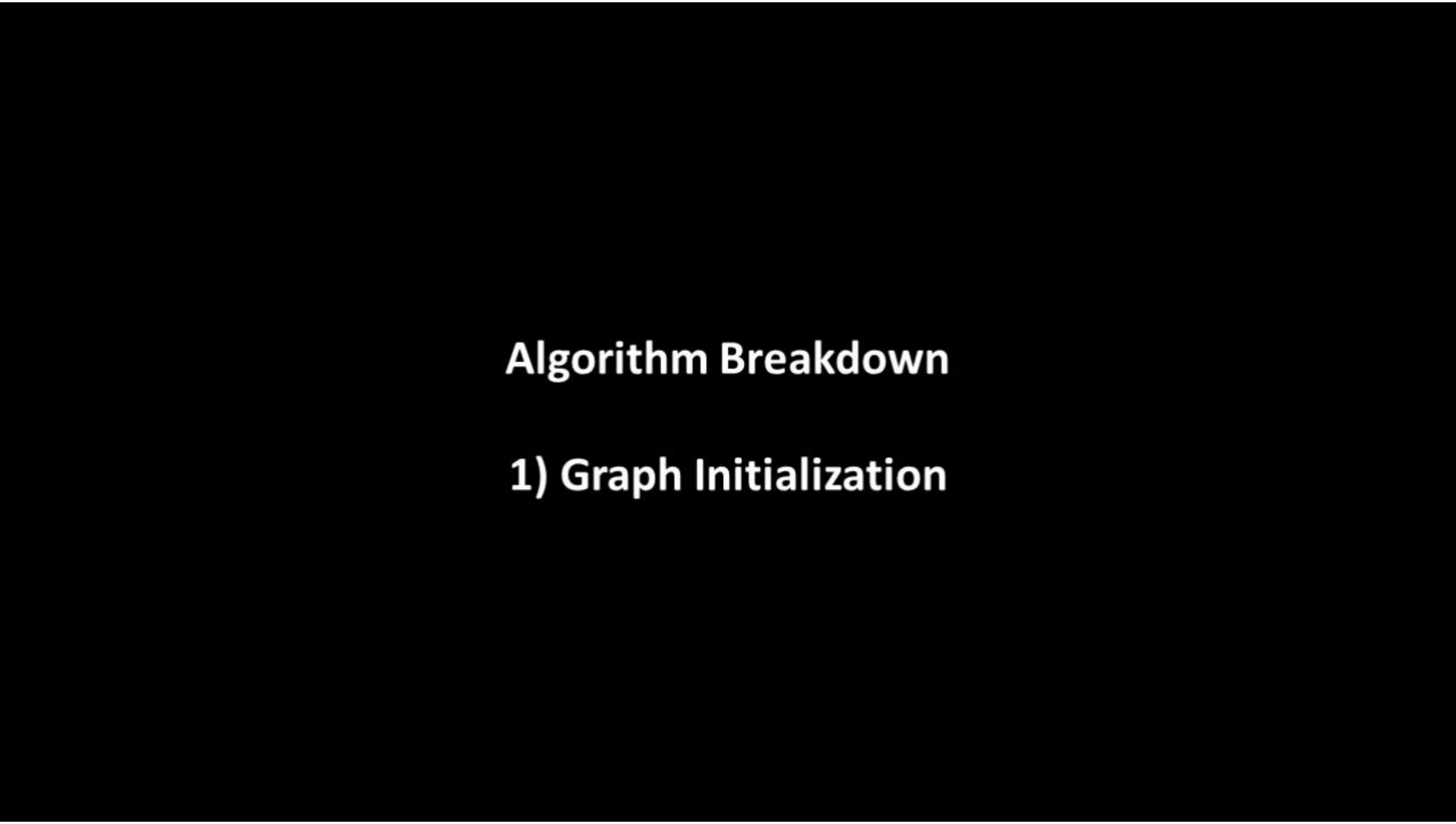
- Real-World Kinect Data
  - 2102 frames of an indoor scene
  - 640 × 480 pixel/frame



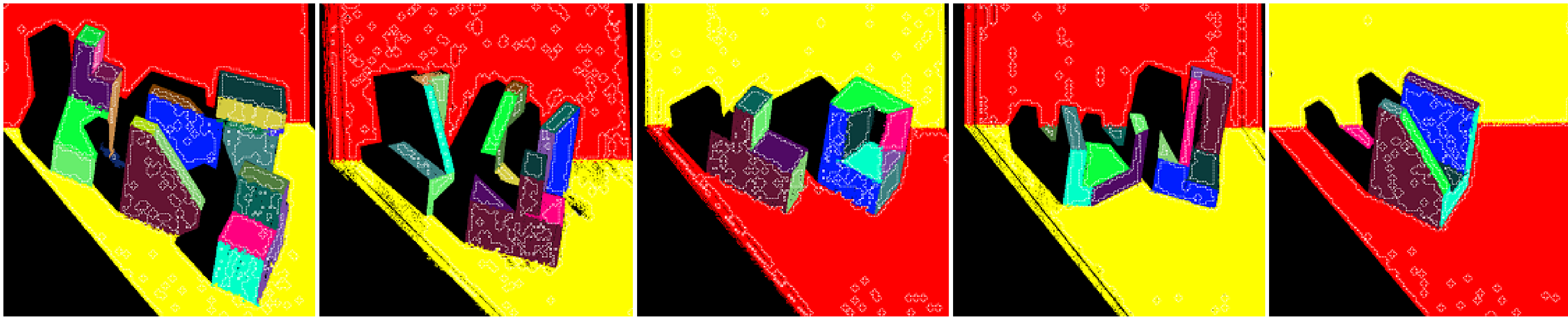
- Real-World Kinect Data
  - 2102 frames of an indoor scene
  - 640 × 480 pixel/frame



- Real-World Kinect Data

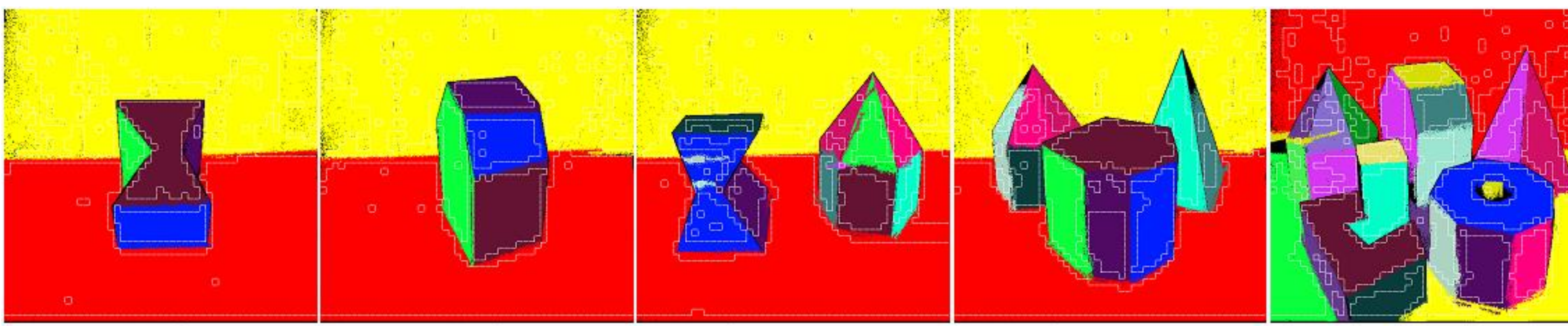


- SegComp Datasets (Hoover et al. PAMI 1996)
  - ABW-TEST



(a) 21/27, 1 under, 4 misses, 3 noises | (b) 18/19, 1 miss | (c) 14/17, 3 misses | (d) 15/17, 2 misses | (e) 10/10

  - PERCEPTRON-TEST



(a) 6/6 | (b) 6/6 | (c) 10/11, 1 over, 1 noise | (d) 12/13, 1 miss, 1 noise | (e) 20/30, 1 over, 1 under, 7 misses, 3 noises

- # SegComp Benchmark (Gotardo et al. CVPR 2003; Oehler et al. ICIRA 2011; Holz & Behnke IAS 2012)

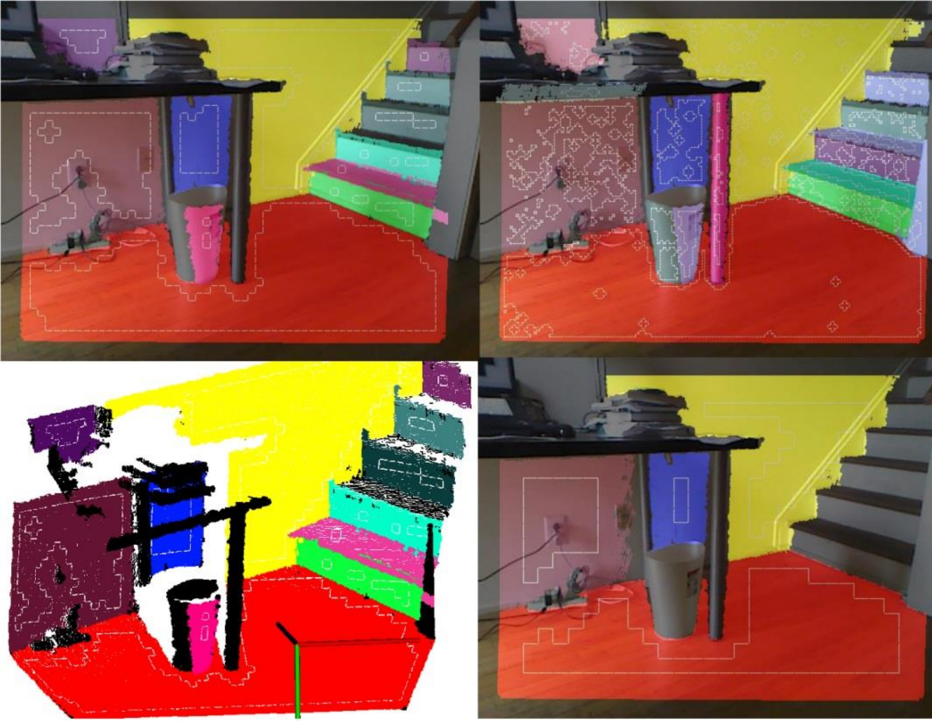
Approach	Regions in ground truth	Correctly detected	Orientation deviation (°)	Over-segmented	Under-segmented	Missed (not detected)	Noise (non-existent)
SegComp ABW data set (30 test images) by Hoover et al. [26], assuming 80% pixel overlap as in [27]							
USF [27]	15.2	12.7 (83.5%)	1.6	0.2	0.1	2.1	1.2
WSU [27]	15.2	9.7 (63.8%)	1.6	0.5	0.2	4.5	2.2
UB [27]	15.2	12.8 (84.2%)	1.3	0.5	0.1	1.7	2.1
UE [27]	15.2	13.4 (88.1%)	1.6	0.4	0.2	1.1	0.8
OU [27]	15.2	9.8 (64.4%)	–	0.2	0.4	4.4	3.2
PPU [27]	15.2	6.8 (44.7%)	–	0.1	2.1	3.4	2.0
UA [27]	15.2	4.9 (32.2%)	–	0.3	2.2	3.6	3.2
UFPR [27]	15.2	13.0 (85.5%)	1.5	0.5	0.1	1.6	1.4
Oehler et al. [2]	15.2	11.1 (73.0%)	1.4	0.2	0.7	2.2	0.8
Holz et al. [8]	15.2	12.2 (80.1%)	1.9	1.8	0.1	0.9	1.3
Ours	**15.2**	**12.8 (84.2%)**	**1.7**	**0.1**	**0.0**	**2.4**	**0.7**
SegComp PERCEPTRON data set (30 test images) by Hoover et al. [26], assuming 80% pixel overlap as in [27]							
USF [27]	14.6	8.9 (60.9%)	2.7	0.4	0.0	5.3	3.6
WSU [27]	14.6	5.9 (40.4%)	3.3	0.5	0.6	6.7	4.8
UB [27]	14.6	9.6 (65.7%)	3.1	0.6	0.1	4.2	2.8
UE [27]	14.6	10.0 (68.4%)	2.6	0.2	0.3	3.8	2.1
UFPR [27]	14.6	11.0 (75.3%)	2.5	0.3	0.1	3.0	2.5
Oehler et al. [2]	14.6	7.4 (50.1%)	5.2	0.3	0.4	6.2	3.9
Holz et al. [8]	14.6	11.0 (75.3%)	2.6	0.4	0.2	2.7	0.3
Ours	**14.6**	**8.9 (60.9%)**	**2.4**	**0.2**	**0.2**	**5.1**	**2.1**

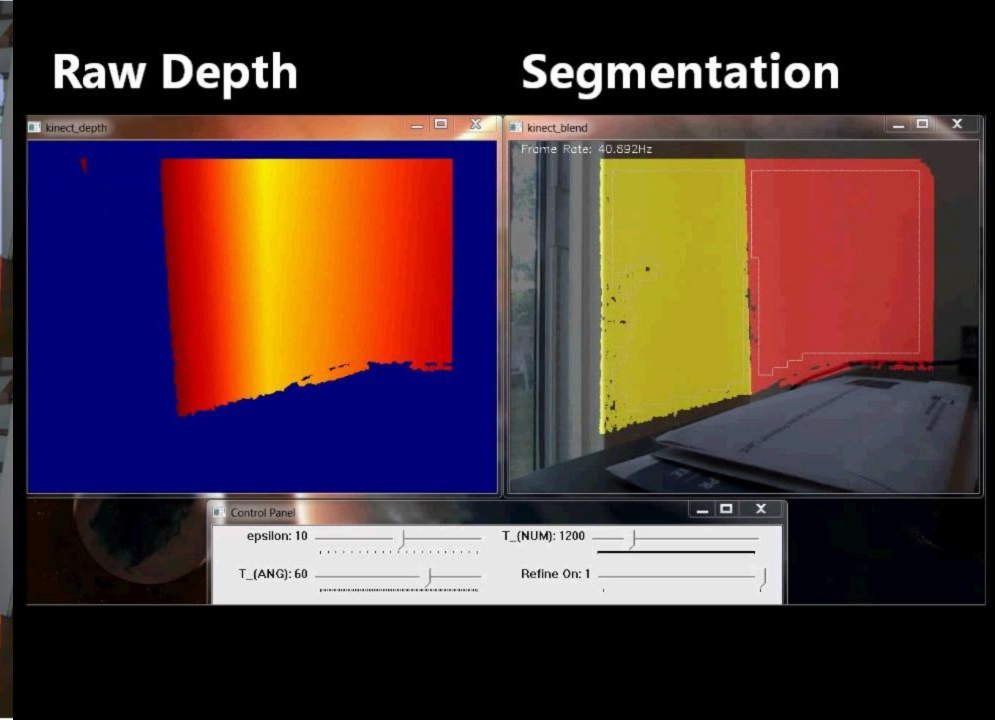
- We presented an efficient plane extraction algorithm based on agglomerative clustering for organized point clouds.

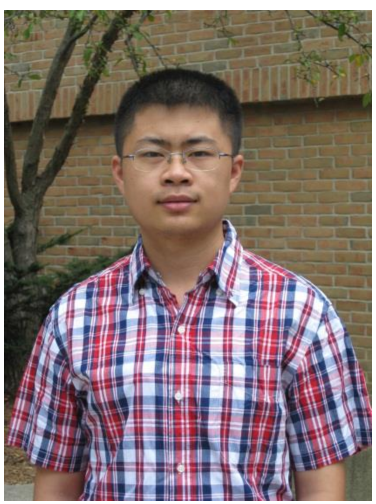
- We presented an efficient plane extraction algorithm based on agglomerative clustering for organized point clouds.
- We analyzed the complexity of the clustering algorithm and shown that it is log-linear in the number of initial nodes.

- We presented an efficient plane extraction algorithm based on agglomerative clustering for organized point clouds.
- We analyzed the complexity of the clustering algorithm and shown that it is log-linear in the number of initial nodes.
- We demonstrated real-time performance with the accuracy comparable to state-of-the-art algorithms.

- Acknowledgement
  - MERL
    - Jay Thornton
    - Srikumar Ramalingam
  - Rackham Graduate School, University of Michigan
  - NSF Student Travel Grant







Chen Feng (in Chinese: 冯晨)

PhD Candidate

Department of Civil and Environmental Engineering

University of Michigan, Ann Arbor

E-mail: cforrest@umich.edu

Web: http://www.umich.edu/~cforrest/