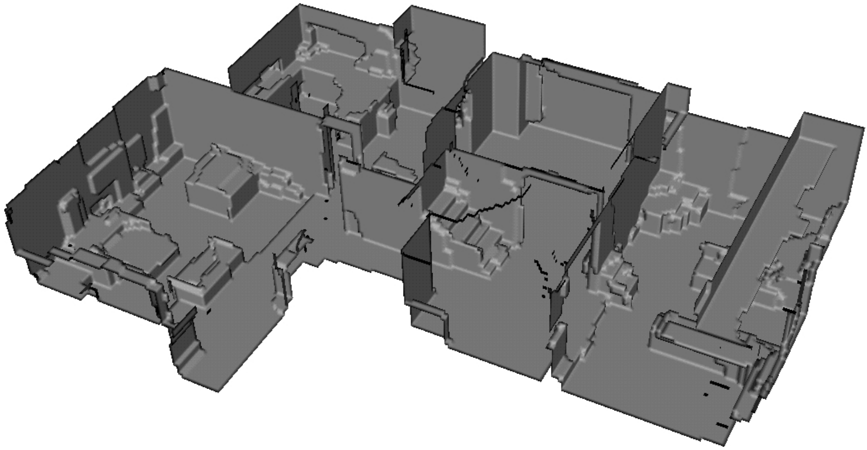
CS6670: Computer Vision

Noah Snavely

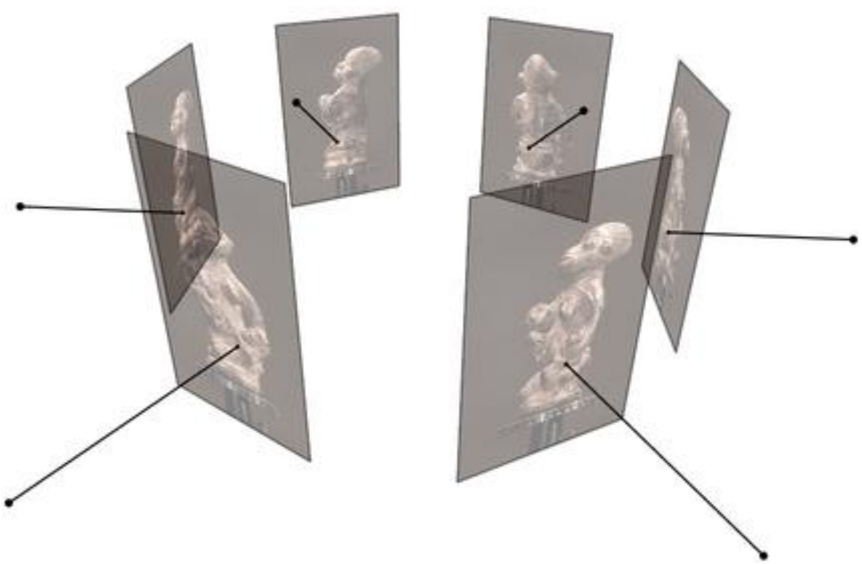
Lecture 25: Multi-view stereo, continued



Multi-view Stereo

Input: calibrated images from several viewpoints

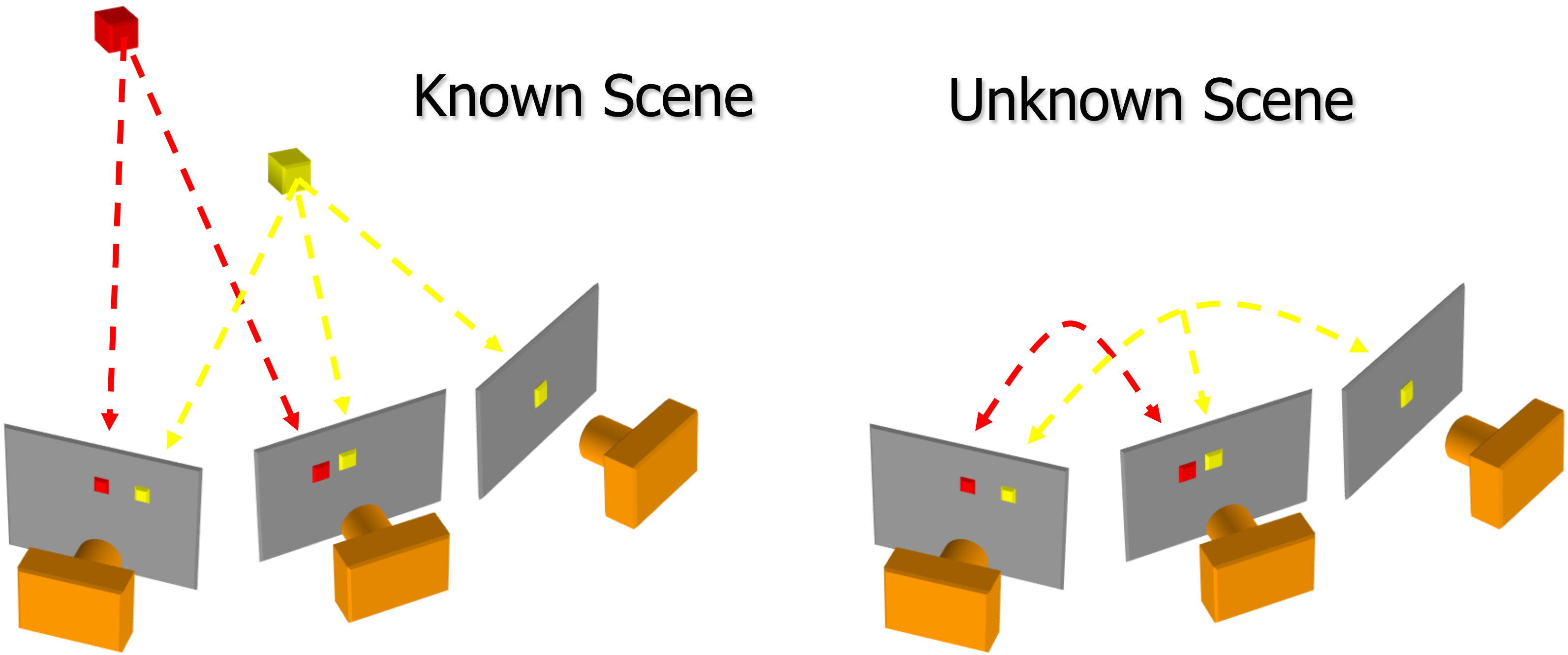
Output: 3D object model



Figures by Carlos Hernandez

The visibility problem

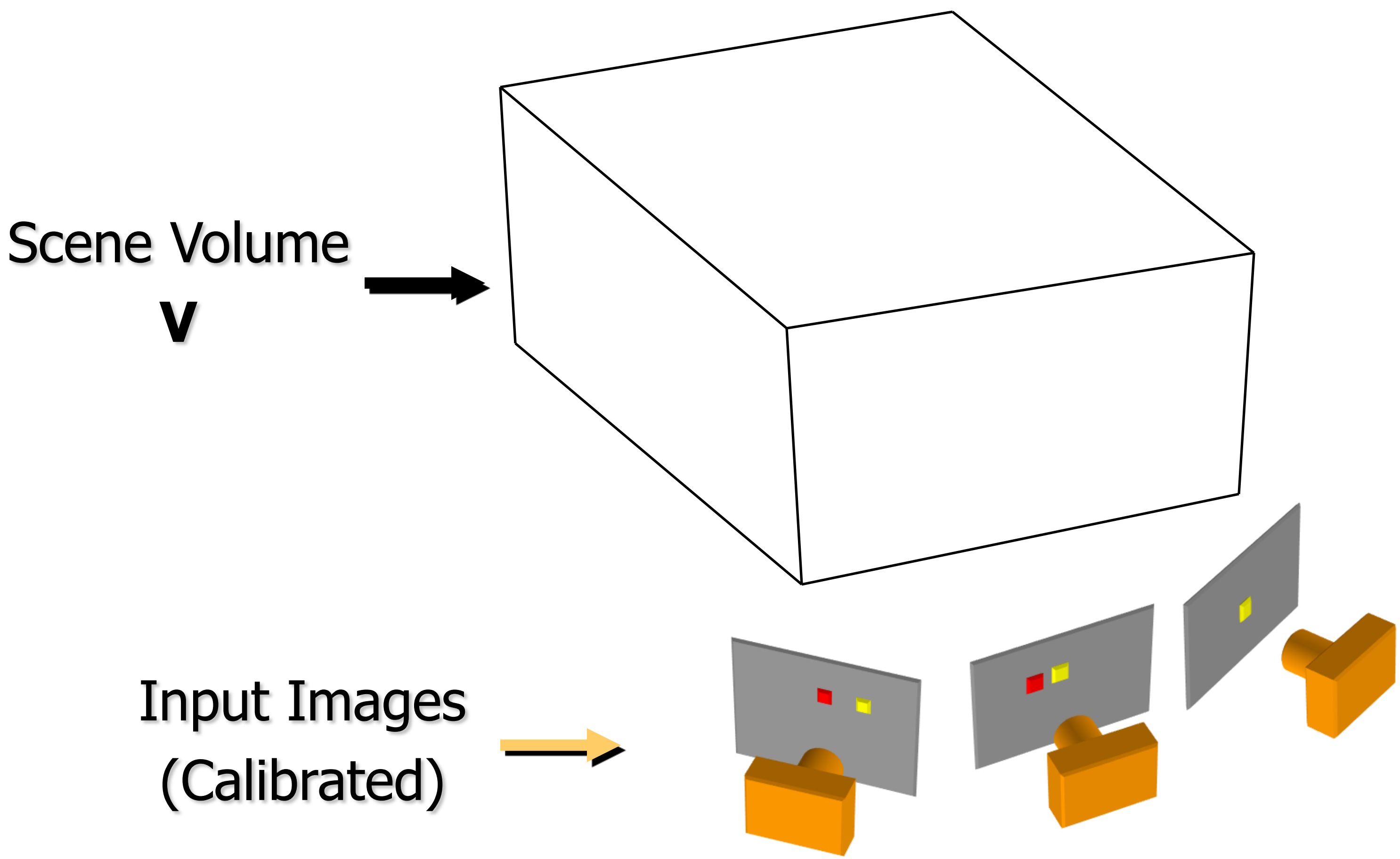
Which points are visible in which images?



Forward Visibility

Inverse Visibility

Volumetric stereo

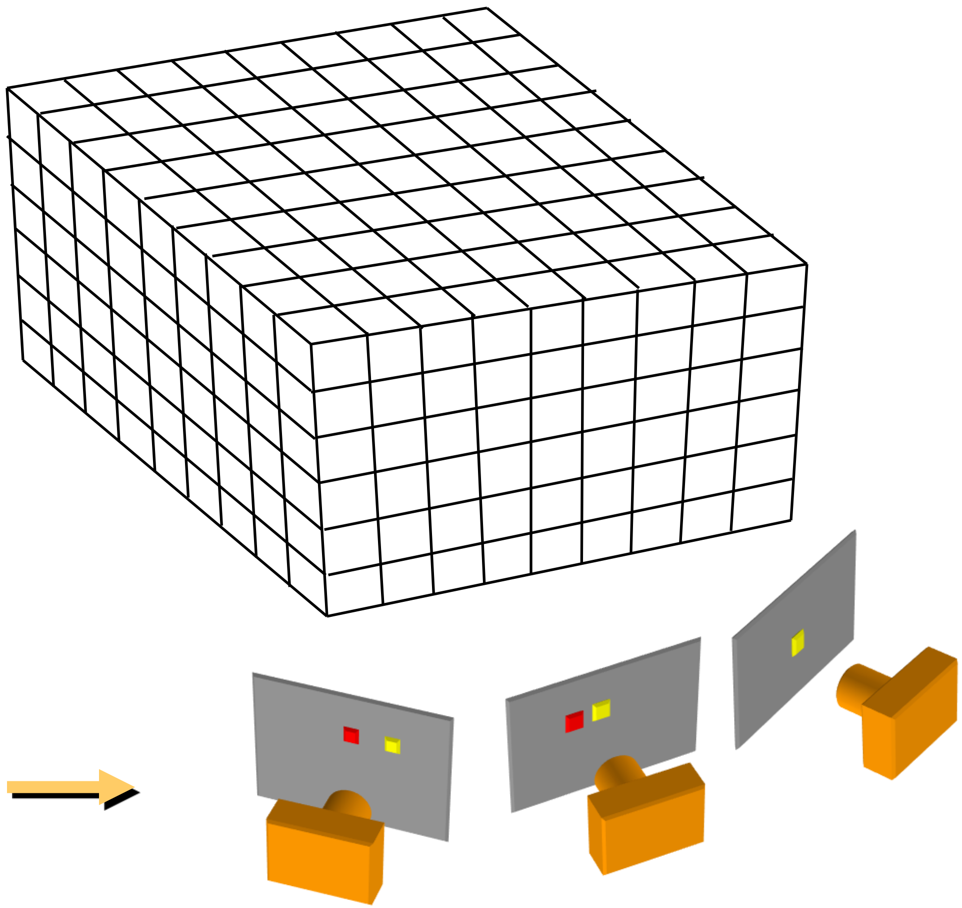


Goal: **Determine occupancy, "color" of points in V**

Discrete formulation: Voxel Coloring

Discretized Scene Volume

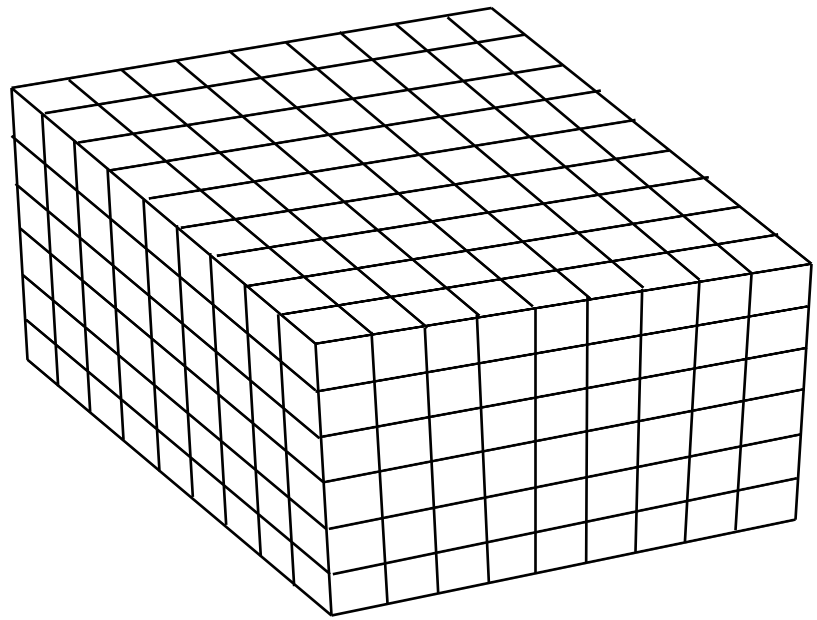
Input Images (Calibrated)

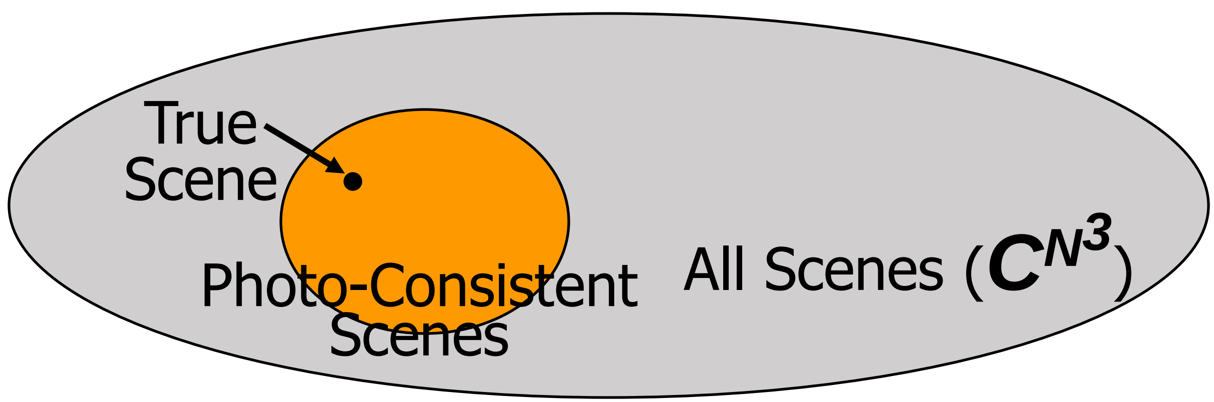


Goal: Assign RGBA values to voxels in V
photo-consistent **with images**

Complexity and computability

Discretized
Scene Volume
N^3 voxels
C colors





Issues

Theoretical Questions

- Identify class of *all* photo-consistent scenes

Practical Questions

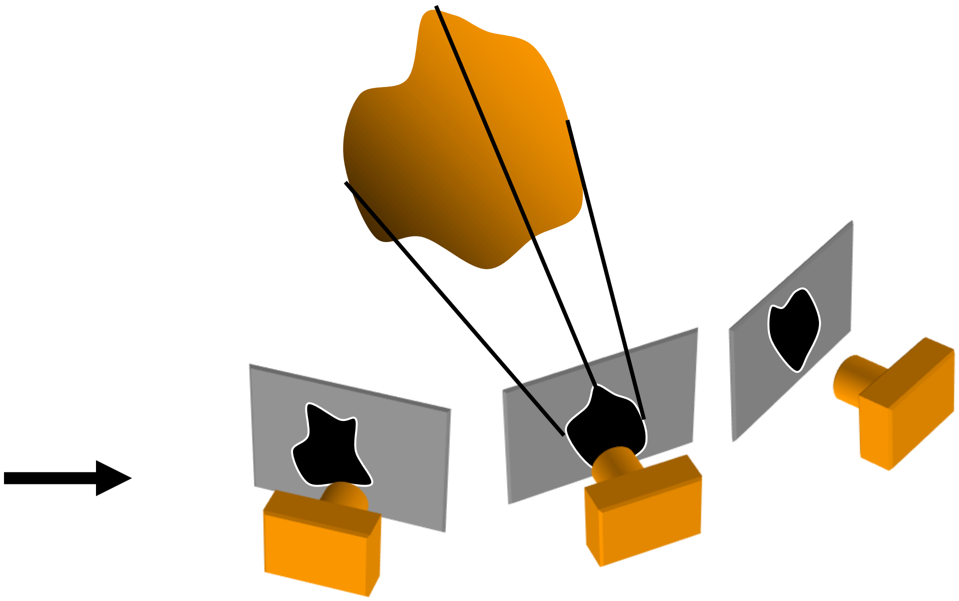
- How do we compute photo-consistent models?

Voxel coloring solutions

1. C=2 (shape from silhouettes)
   - Volume intersection [Baumgart 1974]
     > For more info: *Rapid octree construction from image sequences.* R. Szeliski, CVGIP: Image Understanding, 58(1):23-32, July 1993. (this paper is apparently not available online) or
     > W. Matusik, C. Buehler, R. Raskar, L. McMillan, and S. J. Gortler, *Image-Based Visual Hulls*, SIGGRAPH 2000 (pdf 1.6 MB)
2. C unconstrained, viewpoint constraints
   - Voxel coloring algorithm [Seitz & Dyer 97]
3. General Case
   - Space carving [Kutulakos & Seitz 98]

Reconstruction from Silhouettes (C = 2)

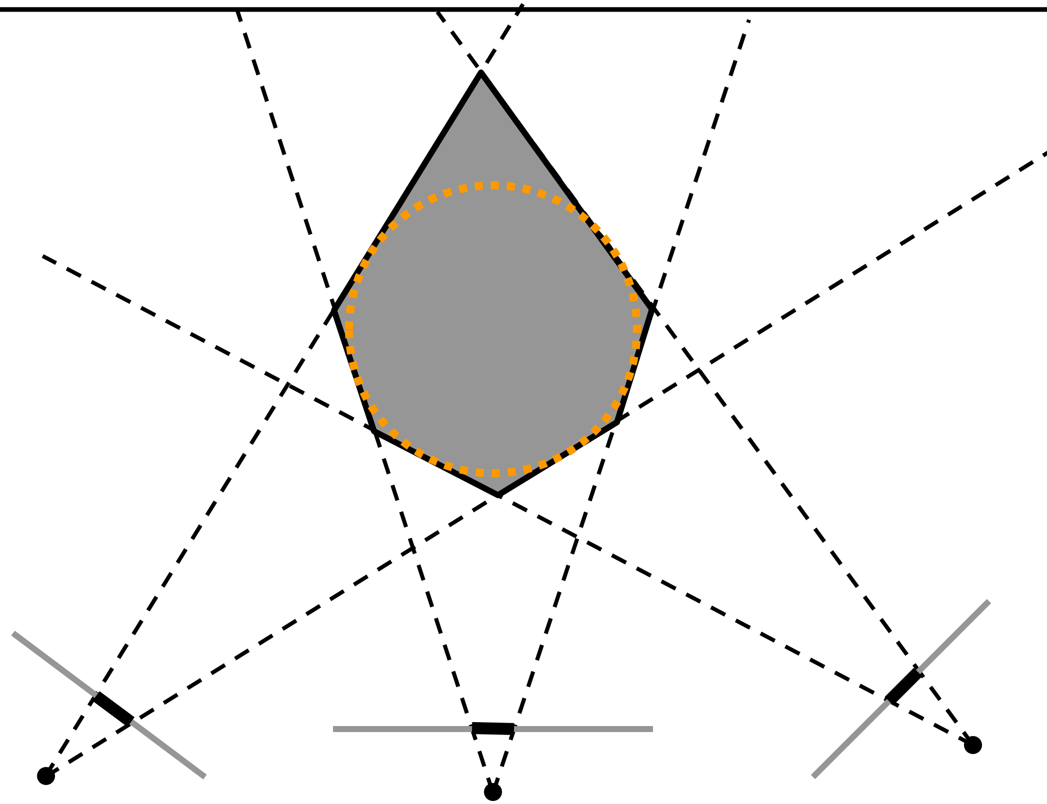
Binary Images



Approach:

- *Backproject* each silhouette
- Intersect backprojected volumes

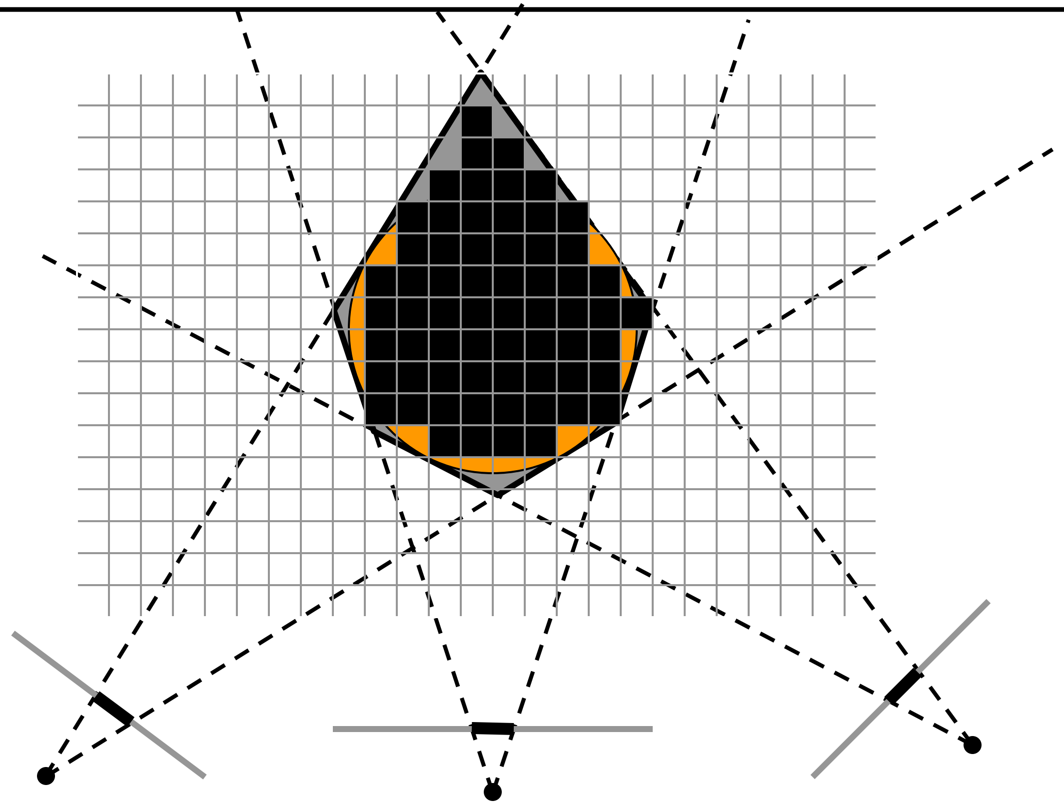
Volume intersection



Reconstruction Contains the True Scene

- But is generally not the same
- In the limit (all views) get *visual hull*
  > Complement of all lines that don’t intersect S

Voxel algorithm for volume intersection



Color voxel black if on silhouette in every image

- O(?), for M images, N^3 voxels
- Don't have to search 2^{N^3} possible scenes!

Properties of Volume Intersection

Pros

- Easy to implement, fast
- Accelerated via octrees [Szeliski 1993] or interval techniques [Matusik 2000]

Cons

- No concavities
- Reconstruction is not photo-consistent
- Requires identification of silhouettes

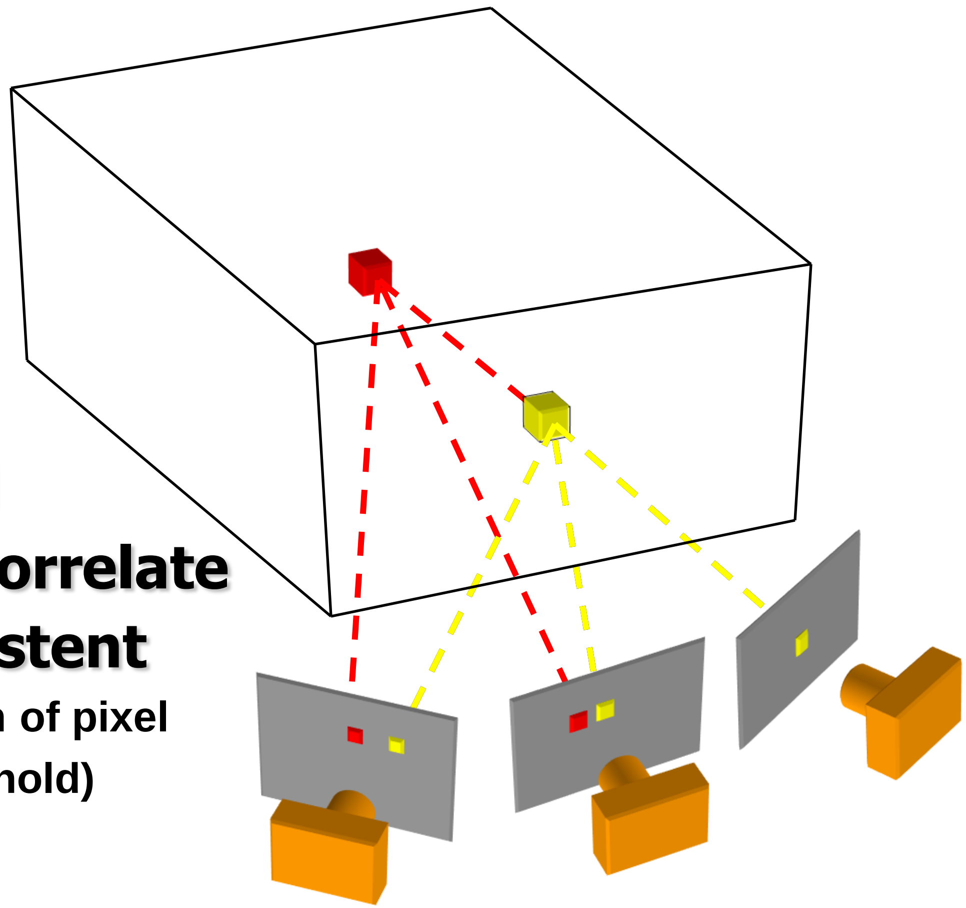
Voxel Coloring Solutions

1. C=2 (silhouettes)
   - Volume intersection [Baumgart 1974]

2. C unconstrained, viewpoint constraints
   - Voxel coloring algorithm [Seitz & Dyer 97]
     - For more info: http://www.cs.washington.edu/homes/seitz/papers/ijcv99.pdf

3. General Case
   - Space carving [Kutulakos & Seitz 98]

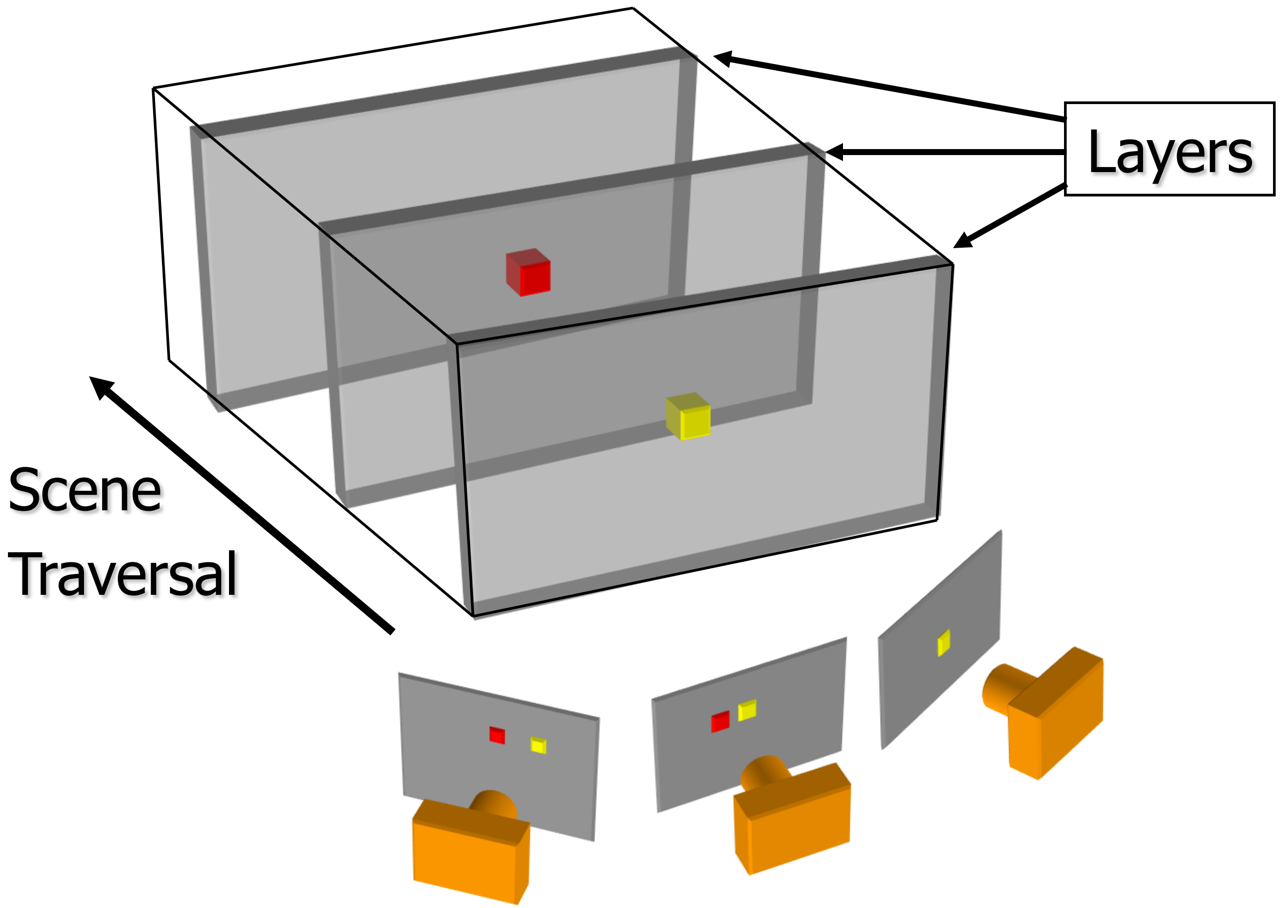
Voxel Coloring Approach



1. **Choose voxel**
2. **Project and correlate**
3. **Color if consistent**
   (standard deviation of pixel colors below threshold)

Visibility Problem: **in which images is each voxel visible?**

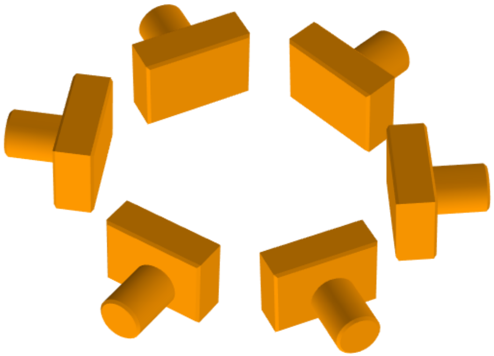
Depth Ordering: visit occluders first!



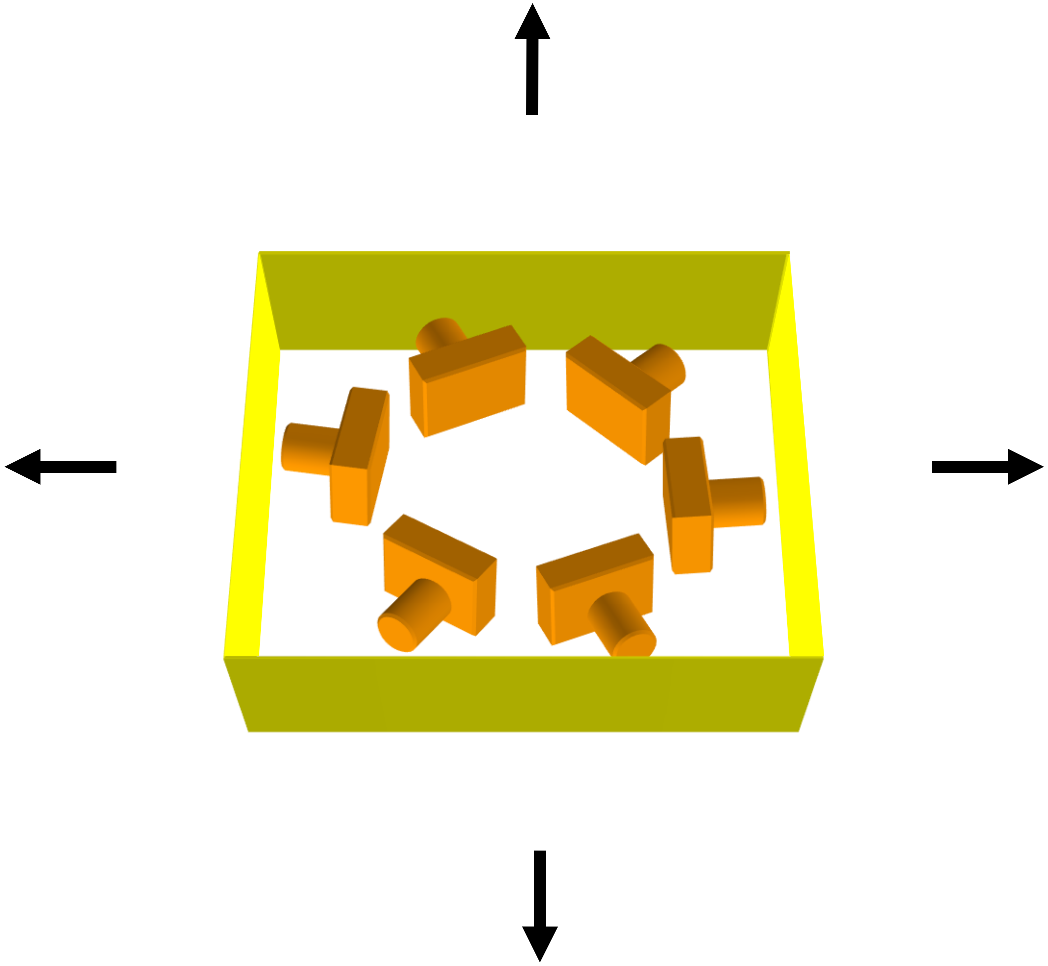
Condition: **depth order is the** ***same for all input views***

Panoramic Depth Ordering

- Cameras oriented in many different directions
- Planar depth ordering does not apply

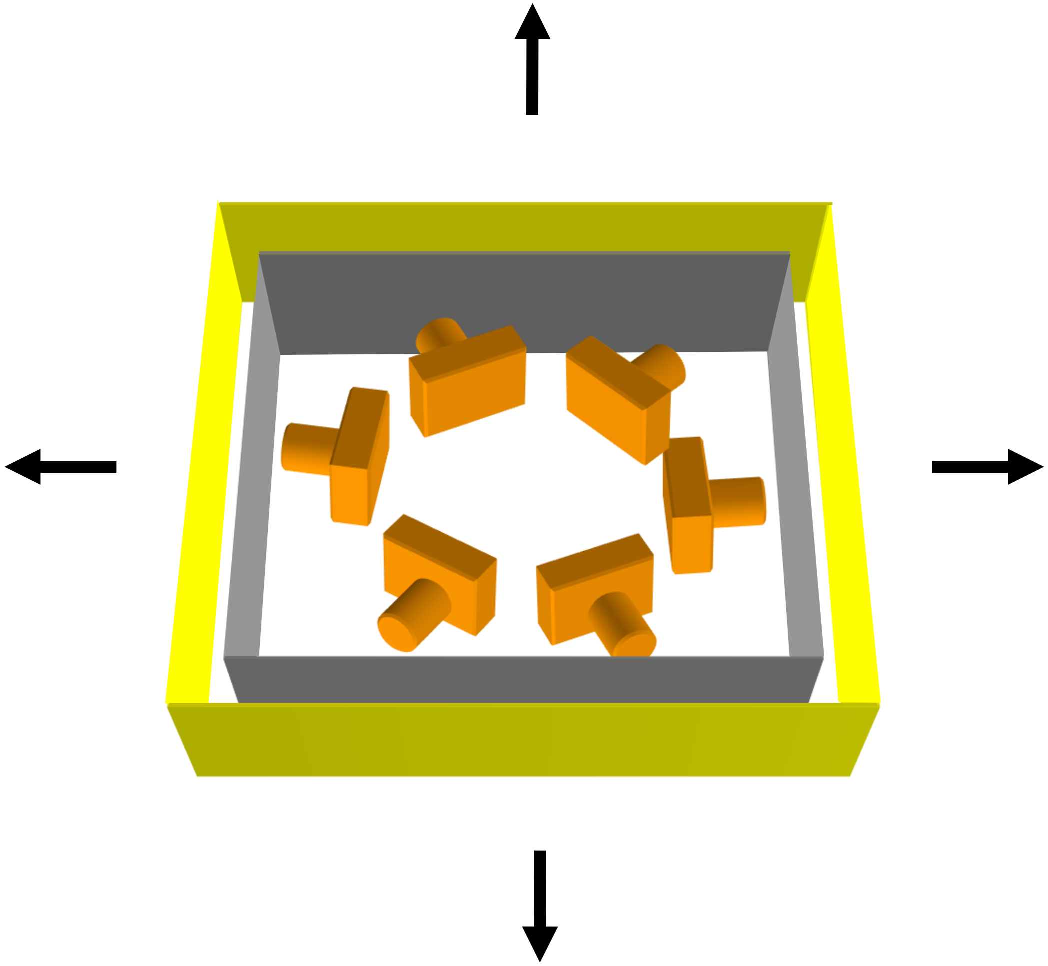


Panoramic Depth Ordering



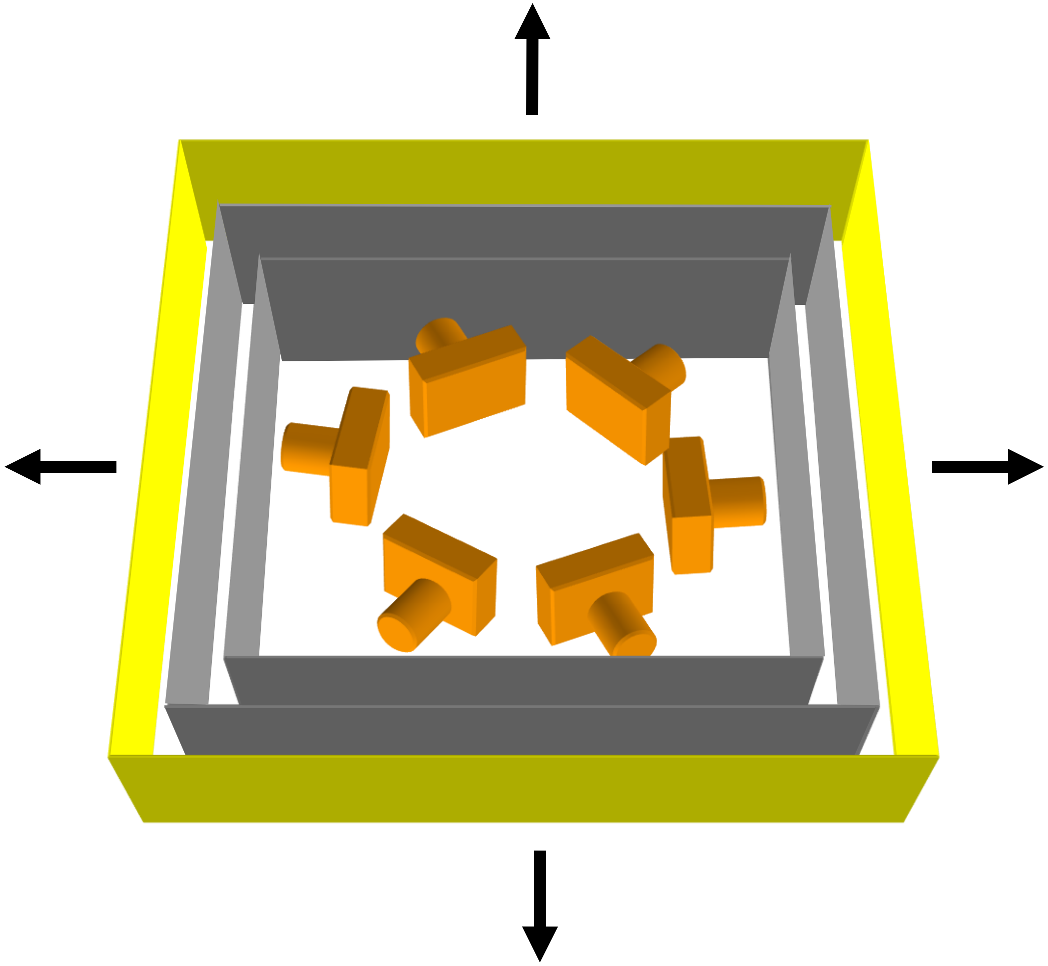
Layers radiate outwards from cameras

Panoramic Layering



Layers radiate outwards from cameras

Panoramic Layering

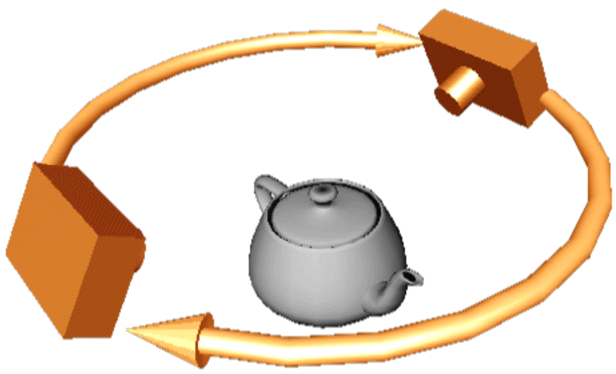


Layers radiate outwards from cameras

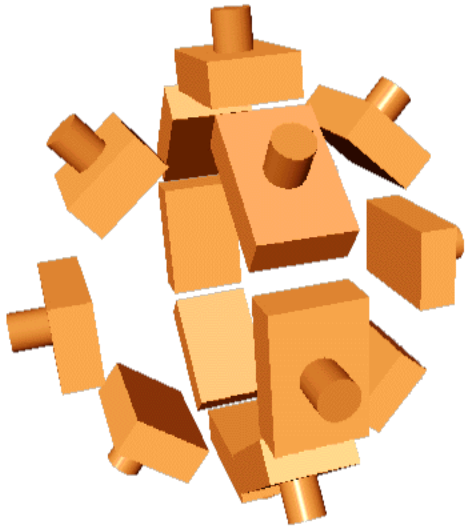
Compatible Camera Configurations

Depth-Order Constraint

- Scene outside convex hull of camera centers



Inward-Looking

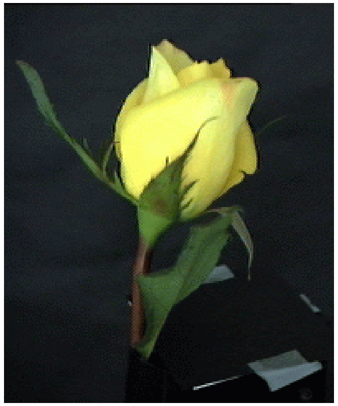
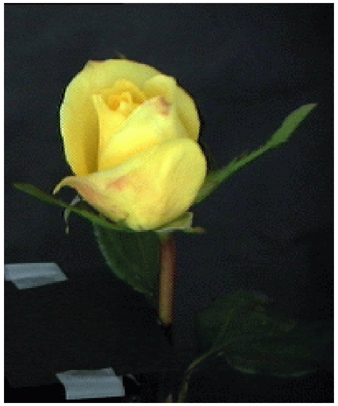


Outward-Looking

Calibrated Image Acquisition

Calibrated Turntable


Selected Dinosaur Images

Selected Flower Images

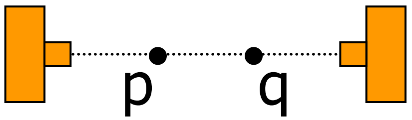
Voxel Coloring Results

Dinosaur Reconstruction
72 K voxels colored
7.6 M voxels tested
7 min. to compute
on a 250MHz SGI

Flower Reconstruction
70 K voxels colored
7.6 M voxels tested
7 min. to compute
on a 250MHz SGI

Limitations of Depth Ordering

A view-independent depth order may not exist



Need more powerful general-case algorithms

- Unconstrained camera positions
- Unconstrained scene geometry/topology

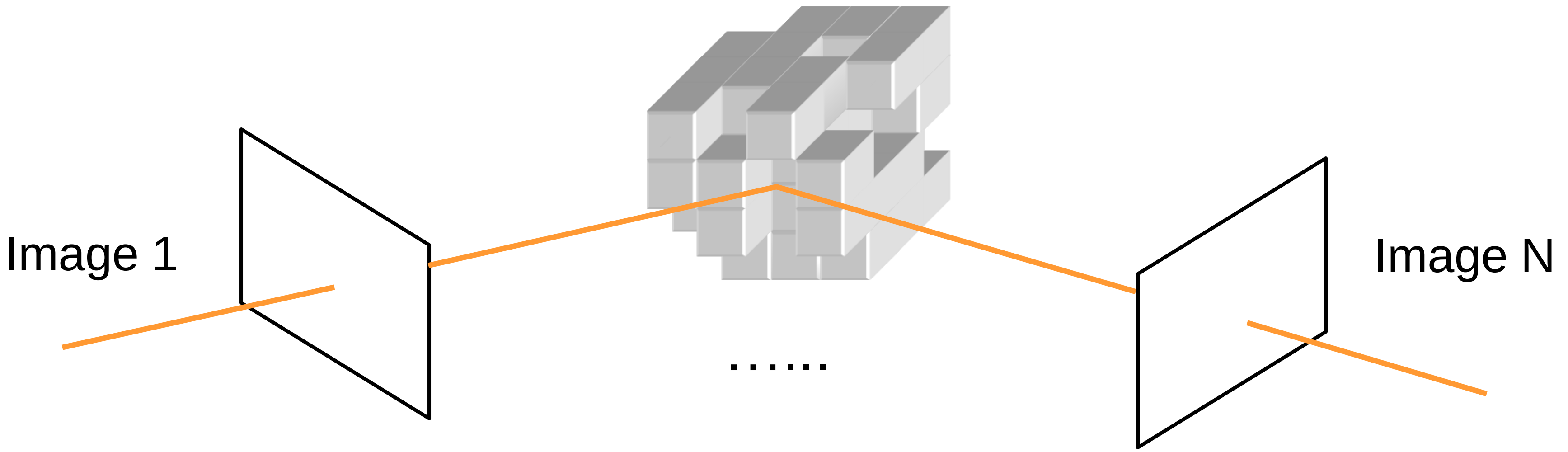
Voxel Coloring Solutions

1. C=2 (silhouettes)
   - Volume intersection [Baumgart 1974]

2. C unconstrained, viewpoint constraints
   - Voxel coloring algorithm [Seitz & Dyer 97]

3. General Case
   - Space carving [Kutulakos & Seitz 98]
     - > For more info: http://www.cs.washington.edu/homes/seitz/papers/kutu-ijcv00.pdf

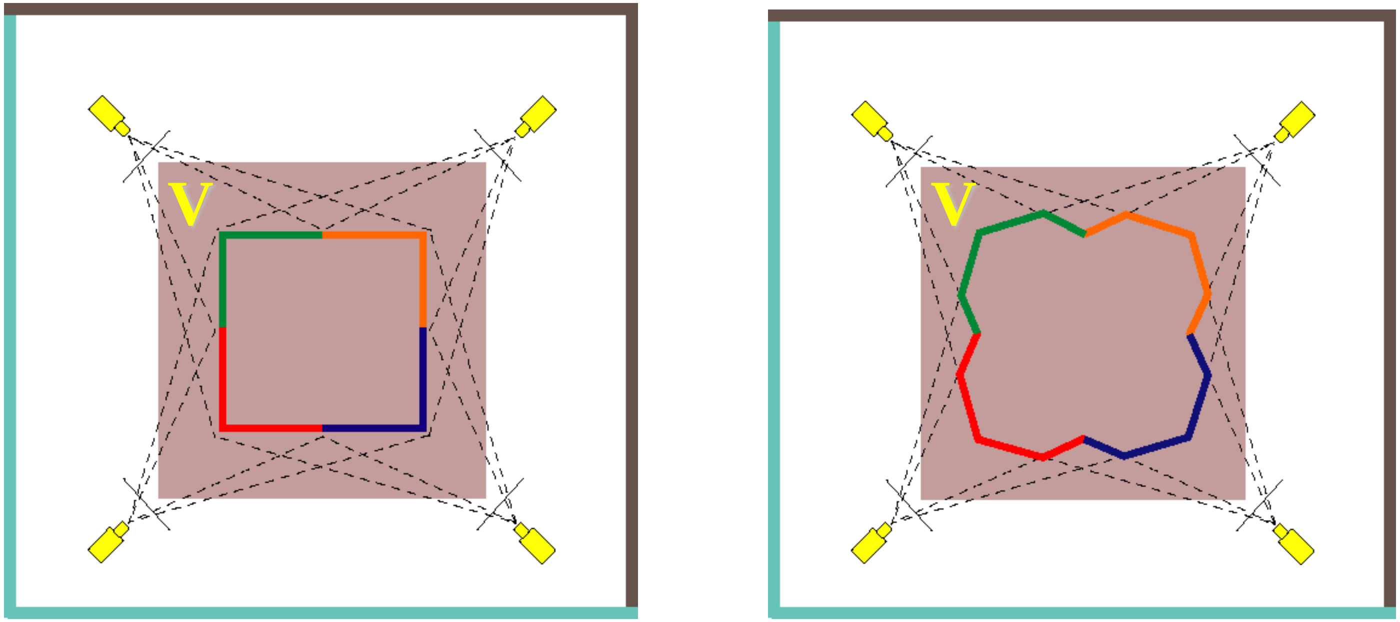
Space Carving Algorithm



Space Carving Algorithm

- Initialize to a volume V containing the true scene
- Choose a voxel on the current surface
- Project to visible input images
- Carve if not photo-consistent
- Repeat until convergence

Which shape do you get?



True Scene **Photo Hull**

The Photo Hull *is the UNION of all photo-consistent scenes in V*

- It is a photo-consistent scene reconstruction
- Tightest possible bound on the true scene

Space Carving Algorithm

Basic algorithm is unwieldy

- Complex update procedure

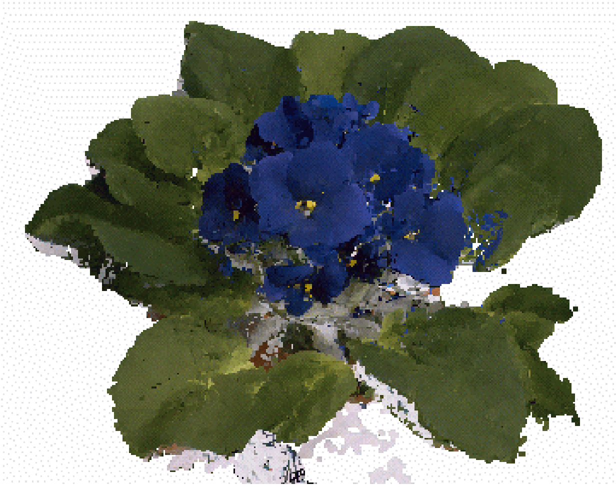
Alternative: Multi-Pass Plane Sweep

- Efficient, can use texture-mapping hardware
- Converges quickly in practice
- Easy to implement

Space Carving Results: African Violet

Input Image (1 of 45)

Reconstruction

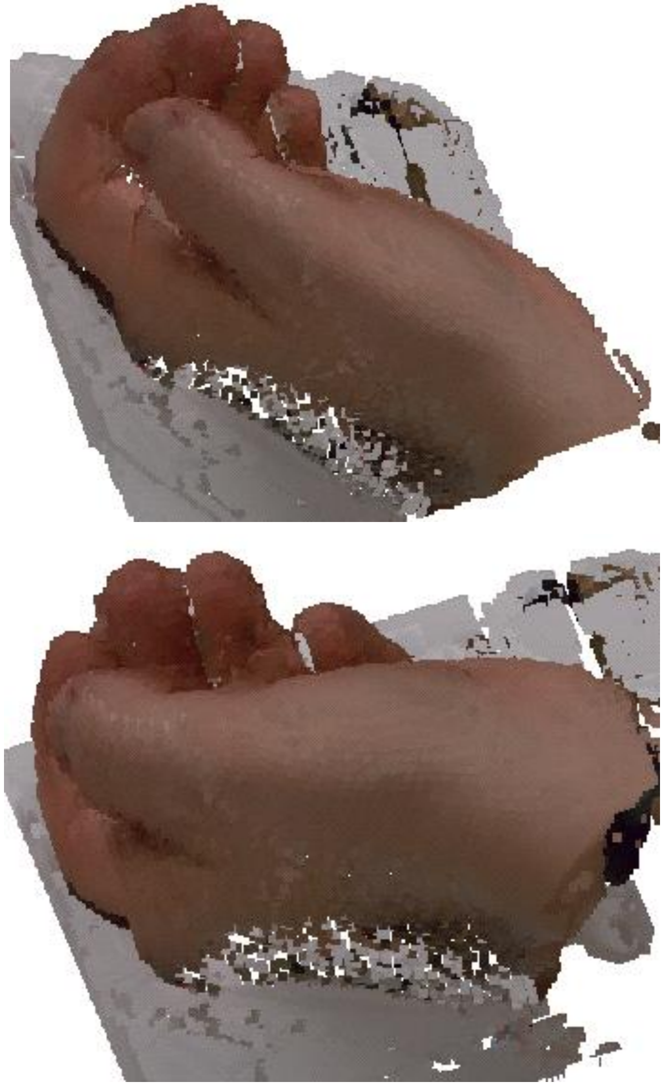


Reconstruction

Reconstruction

Space Carving Results: Hand

Input Image
(1 of 100)



Views of Reconstruction

Properties of Space Carving

Pros

- Voxel coloring version is easy to implement, fast
- Photo-consistent results
- No smoothness prior

Cons

- Bulging
- No smoothness prior

Improvements

Unconstrained camera viewpoints

- Space carving [Kutulakos & Seitz 98]

Evolving a surface

- Level sets [Faugeras & Keriven 98]
- More recent work by Pons et al.

Global optimization

- Graph cut approaches
  - [Kolmogoriv & Zabih, ECCV 2002]
  - [Vogiatzis et al., PAMI 2007]

Modeling shiny (and other reflective) surfaces

- e.g., Zickler et al., *Helmholtz Stereopsis*

Questions?

- 2-minute break

Reconstructing Building Interiors from Images

Yasutaka Furukawa   Brian Curless   Steven M. Seitz
University of Washington, Seattle, USA

Richard Szeliski
Microsoft Research, Redmond, USA

Reconstruction & Visualization of Architectural Scenes

- Manual (semi-automatic)
  - Google Earth & Virtual Earth
  - Façade [Debevec et al., 1996]
  - CityEngine [Müller et al., 2006, 2007]
- Automatic
  - Ground-level images [Cornelis et al., 2008, Pollefeys et al., 2008]
  - Aerial images [Zebedin et al., 2008]

Google Earth

Virtual Earth

Müller et al.

Zebedin et al.

Reconstruction & Visualization of Architectural Scenes

- Manual (semi-automatic)
  - Google Earth & Virtual Earth
  - Façade [Debevec et al., 1996]
  - CityEngine [Müller et al., 2006, 2007]
- Automatic
  - Ground-level images [Cornelis et al., 2008, Pollefeys et al., 2008]
  - Aerial images [Zebedin et al., 2008]

Google Earth

Virtual Earth

Müller et al.

Zebedin et al.

Reconstruction & Visualization of Architectural Scenes

- Manual (semi-automatic)
  - Google Earth & Virtual Earth
  - Façade [Debevec et al., 1996]
  - CityEngine [Müller et al., 2006, 2007]
- Automatic
  - Ground-level images [Cornelis et al., 2008, Pollefeys et al., 2008]
  - Aerial images [Zebedin et al., 2008]


Google Earth


Virtual Earth


Müller et al.


Zebedin et al.

Reconstruction & Visualization of Architectural Scenes

Little attention paid to indoor scenes

Google Earth

Virtual Earth

Müller et al.

Zebedin et al.

Our Goal

- Fully automatic system for indoors/outdoors
  - Reconstructs a simple 3D model from images
  - Provides real-time interactive visualization

What are the challenges?

Challenges - Reconstruction

- Multi-view stereo (MVS) typically produces a dense model
- We want the model to be
  - Simple for real-time interactive visualization of a large scene (e.g., a whole house)
  - Accurate for high-quality image-based rendering

Challenges - Reconstruction

- Multi-view stereo (MVS) typically produces a dense model
- We want the model to be
  - Simple for real-time interactive visualization of a large scene (e.g., a whole house)
  - Accurate for high-quality image-based rendering

Simple mode is effective for compelling visualization

Challenges – Indoor Reconstruction

Texture-poor surfaces

Challenges – Indoor Reconstruction

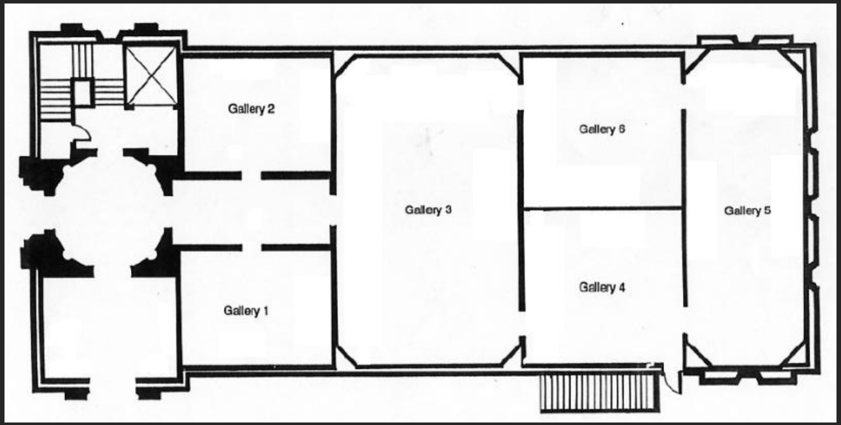
Texture-poor surfaces

Complicated visibility

Challenges – Indoor Reconstruction

Texture-poor surfaces

Complicated visibility



Prevalence of thin structures
(doors, walls, tables)

System pipeline

Images

Images

System pipeline

Structure-from-Motion

Bundler by Noah Snavely
Structure from Motion for unordered image collections
http://phototour.cs.washington.edu/bundler/

Images

System pipeline

Images

SFM

System pipeline

Multi-view Stereo

PMVS by Yasutaka Furukawa and Jean Ponce
Patch-based Multi-View Stereo Software
http://grail.cs.washington.edu/software/pmvs/

Images

SFM

System pipeline

Images

SFM

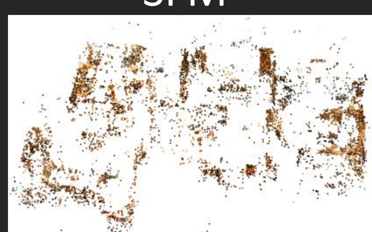
MVS

System pipeline

Manhattan-world Stereo
[Furukawa et al., CVPR 2009]

Images

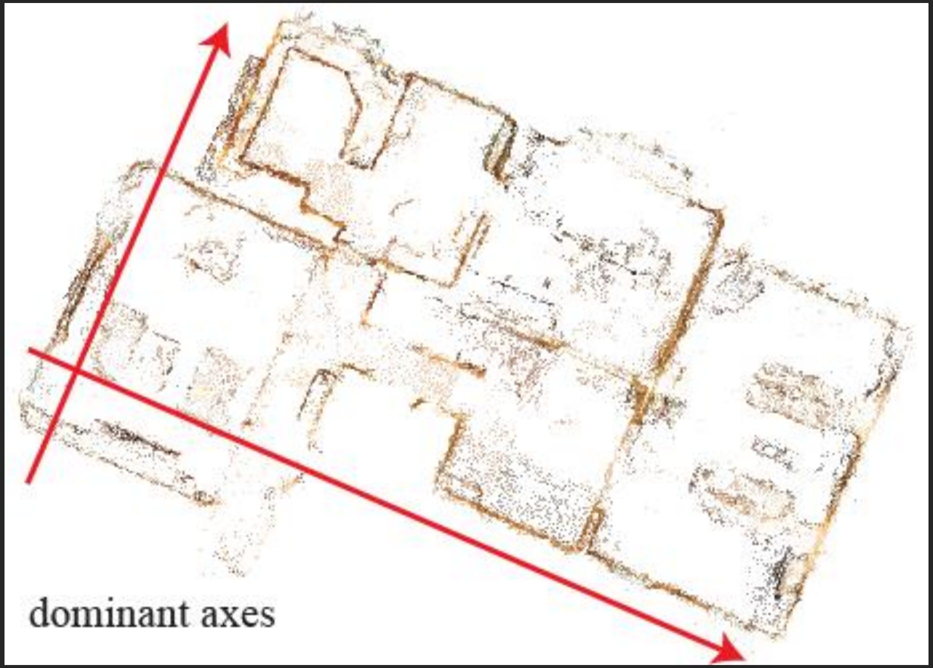
SFM



MVS

System pipeline

Manhattan-world Stereo
[Furukawa et al., CVPR 2009]



Images

SFM

MVS

System pipeline

Manhattan-world Stereo
[Furukawa et al., CVPR 2009]

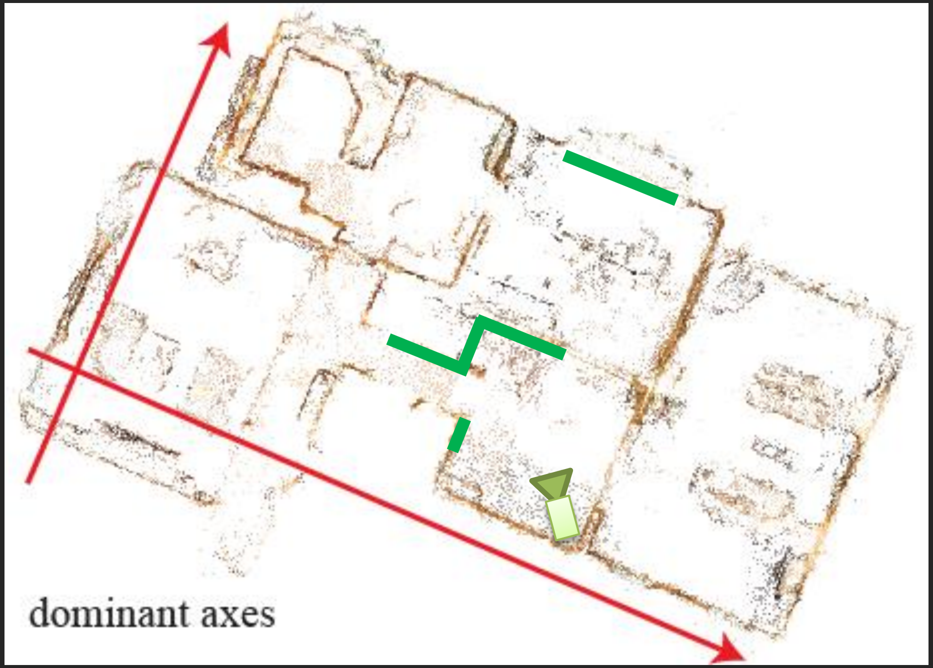
Images

SFM

MVS

System pipeline

Manhattan-world Stereo
[Furukawa et al., CVPR 2009]



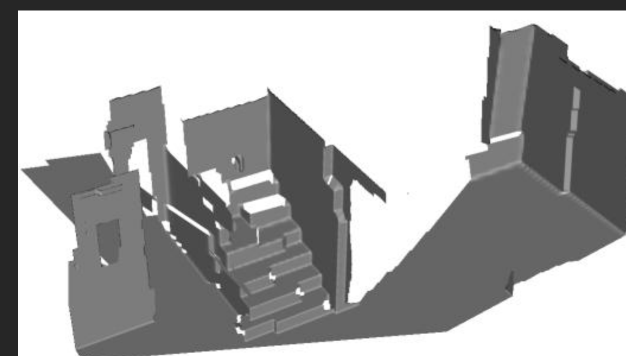
Images

SFM

MVS

System pipeline

Manhattan-world Stereo
[Furukawa et al., CVPR 2009]



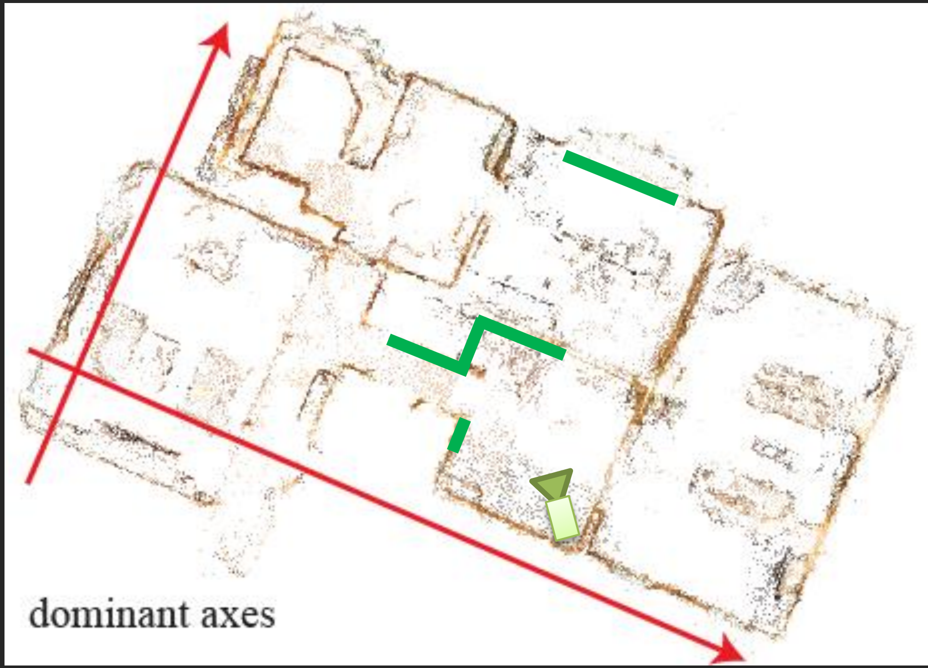
Images

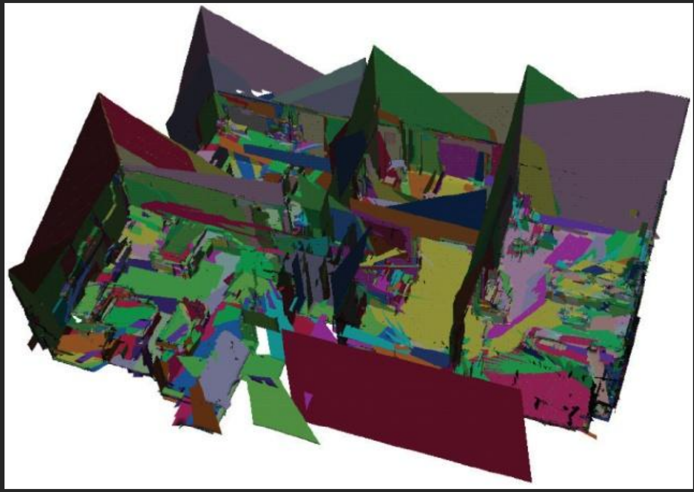
SFM

MVS

System pipeline

Manhattan-world Stereo
[Furukawa et al., CVPR 2009]





Images

SFM

MVS

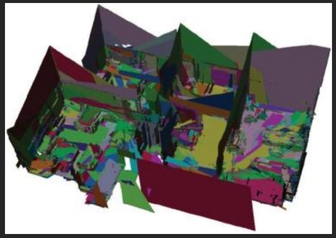
System pipeline

Images

SFM

MVS

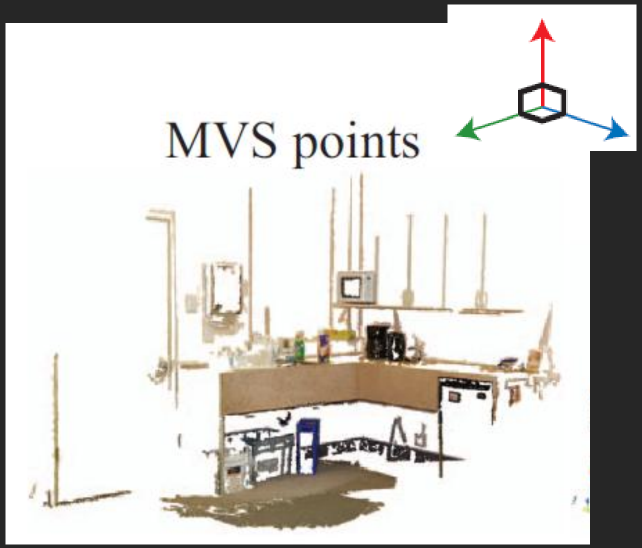
MWS



Manhattan-World Stereo (MWS)

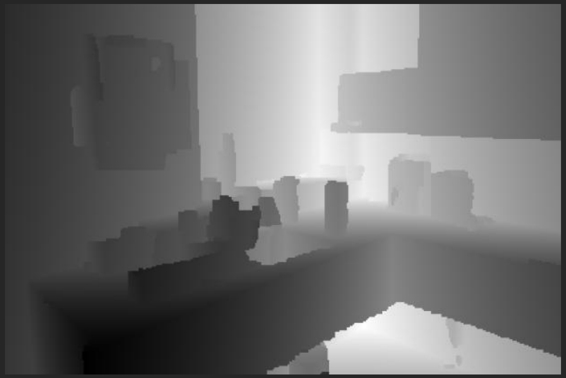
0. Assume that surfaces are oriented along three mutually orthogonal directions
1. Axis-align model by detecting dominant orientations
2. Generate hypothesis planes for each direction
3. Label each pixel in each image with a plane (graph cuts)

Finding hypothesis planes

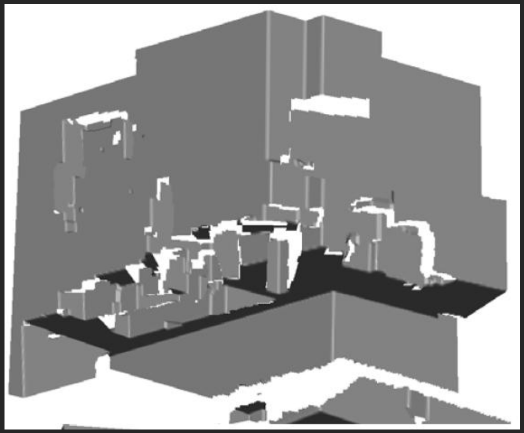


Example

Input image



Computed depth map



Mesh model

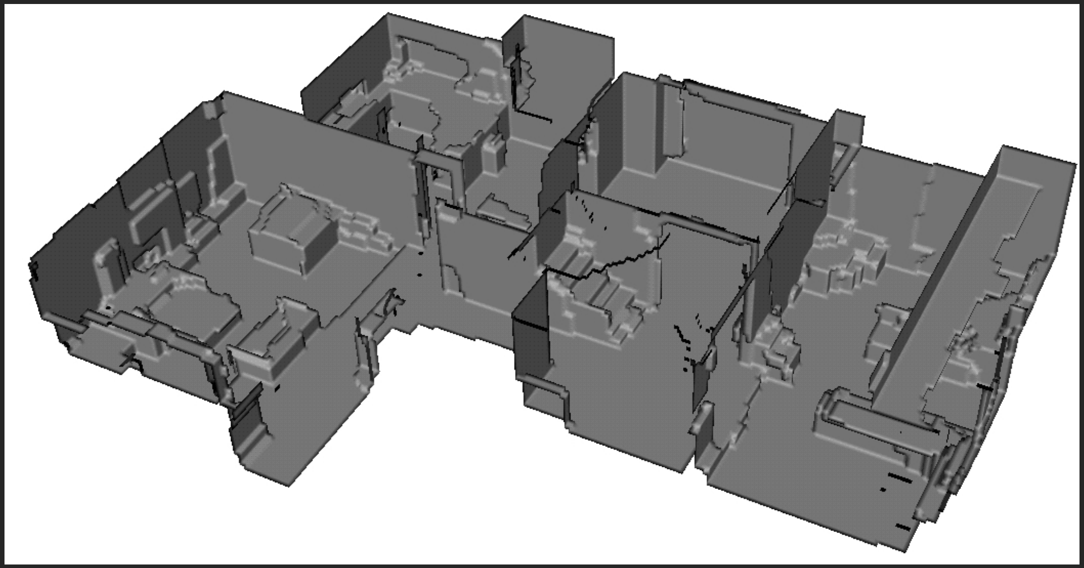
Texture-mapped model

Manhattan-World Stereo (MWS)

Video

System pipeline

Axis-aligned depth map merging
(our contribution)

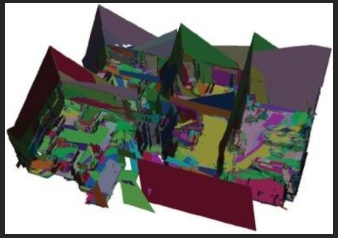


Images

SFM

MVS

MWS



System pipeline

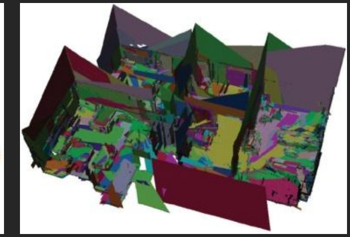
Rendering: simple view-dependent texture mapping

Images

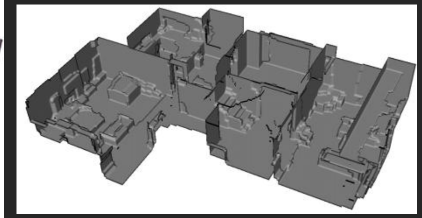
SFM

MVS

MWS



Merging

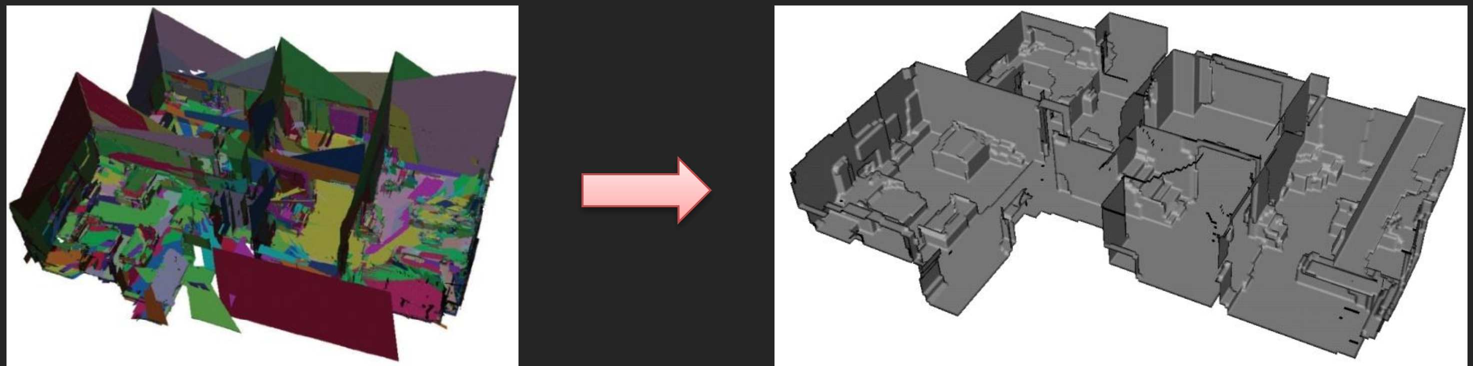


Outline

- System pipeline (system contribution)
- Algorithmic details (technical contribution)
- Experimental results
- Conclusion and future work

Axis-aligned Depth-map Merging

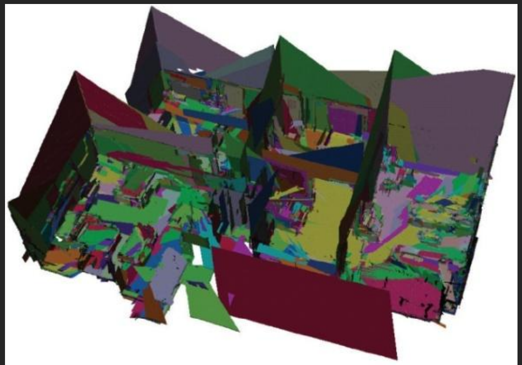
- Basic framework is similar to volumetric MRF
  [Vogiatzis 2005, Sinha 2007, Zach 2007, Hernández 2007]

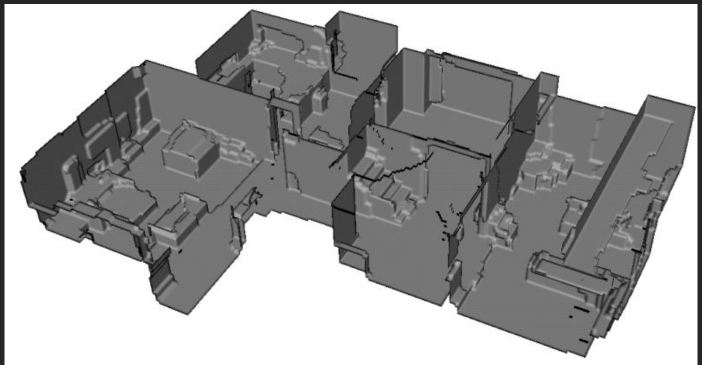


Axis-aligned Depth-map Merging

- Basic framework is similar to volumetric MRF
  [Vogiatzis 2005, Sinha 2007, Zach 2007, Hernández 2007]

Assign label $l(\boldsymbol{v})$ (***empty*** or ***occupied***)
to each voxel $\boldsymbol{v}$

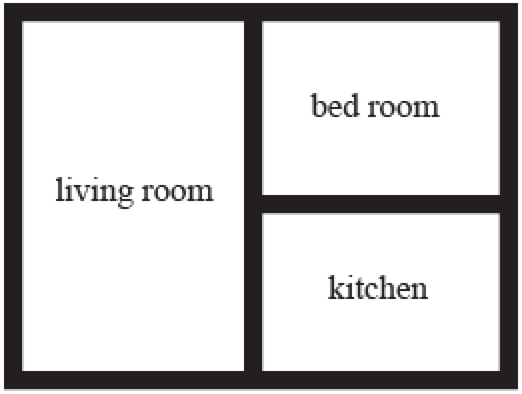


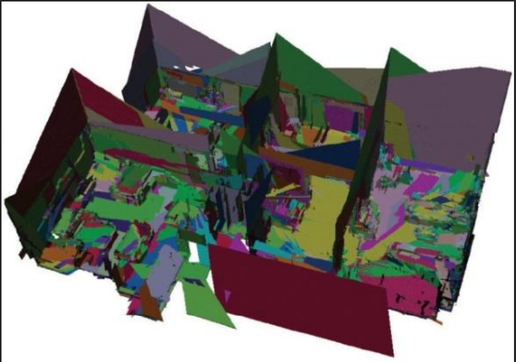


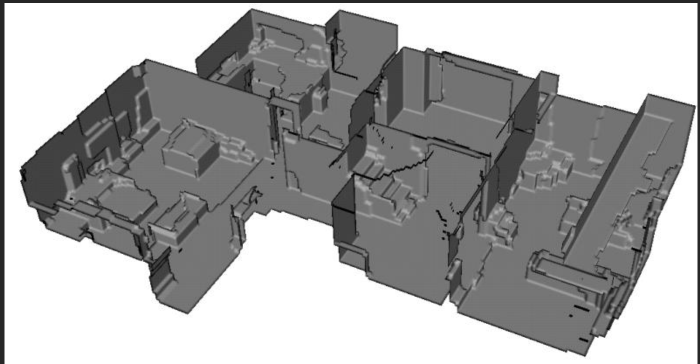
Axis-aligned Depth-map Merging

- Basic framework is similar to volumetric MRF [Vogiatzis 2005, Sinha 2007, Zach 2007, Hernández 2007]

Assign label $l(\boldsymbol{v})$ (***empty*** or ***occupied***) to each voxel $\boldsymbol{v}$



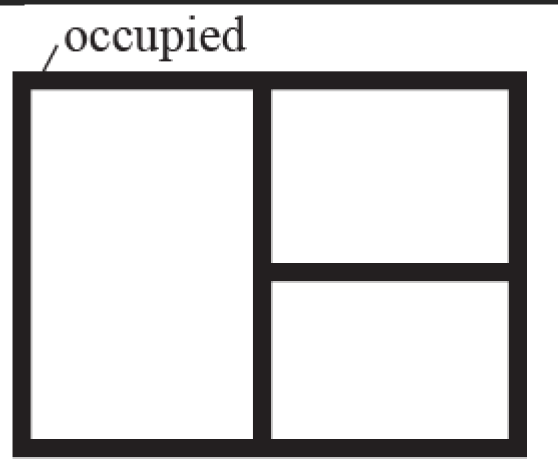


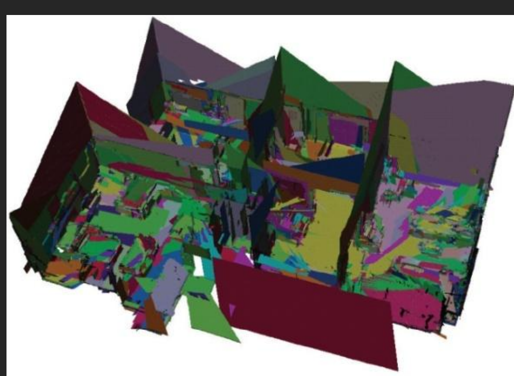


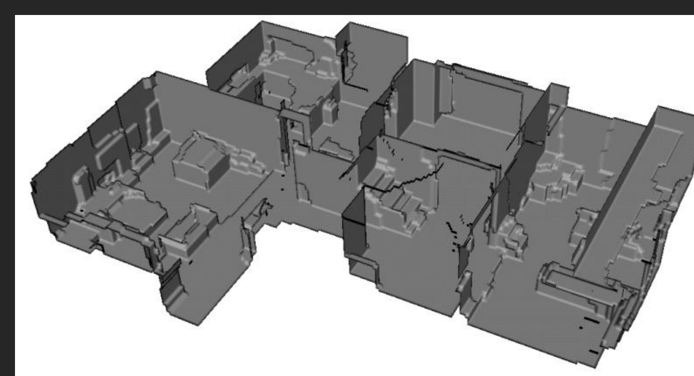
Axis-aligned Depth-map Merging

- Basic framework is similar to volumetric MRF [Vogiatzis 2005, Sinha 2007, Zach 2007, Hernández 2007]

Assign label $l(\boldsymbol{v})$ (***empty*** or ***occupied***) to each voxel $\boldsymbol{v}$



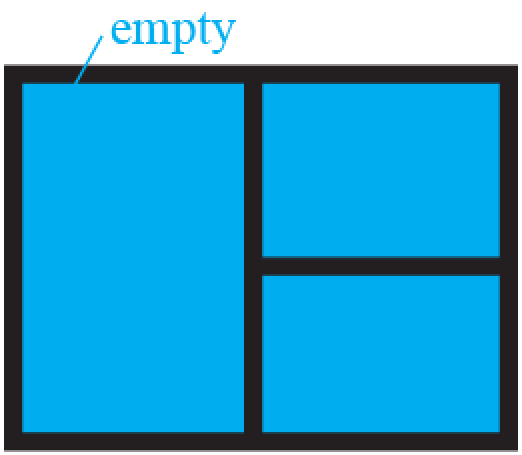


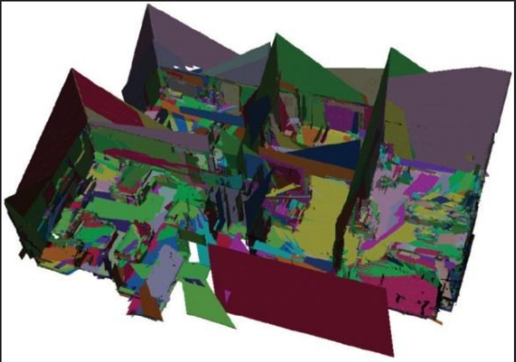


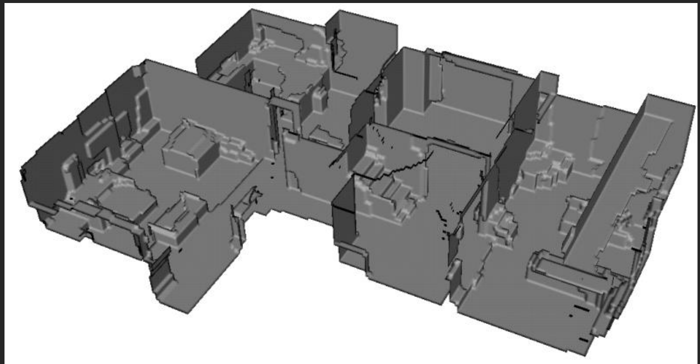
Axis-aligned Depth-map Merging

- Basic framework is similar to volumetric MRF
  [Vogiatzis 2005, Sinha 2007, Zach 2007, Hernández 2007]

Assign label $l(v)$ (***empty*** or ***occupied***) to each voxel v







Axis-aligned Depth-map Merging

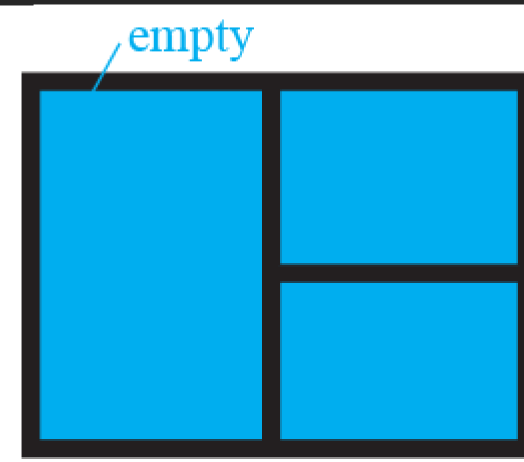
- Basic framework is similar to volumetric MRF [Vogiatzis 2005, Sinha 2007, Zach 2007, Hernández 2007]

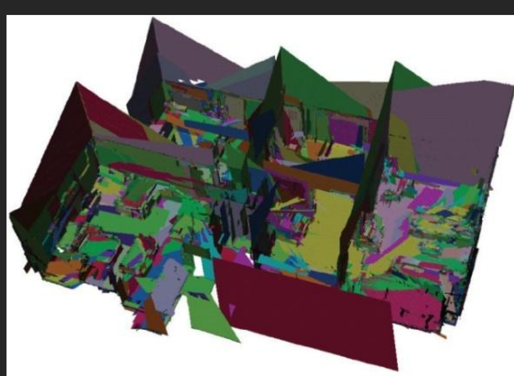
Assign label $l(v)$ (***empty*** or ***occupied***) to each voxel $\boldsymbol{v}$ to minimize

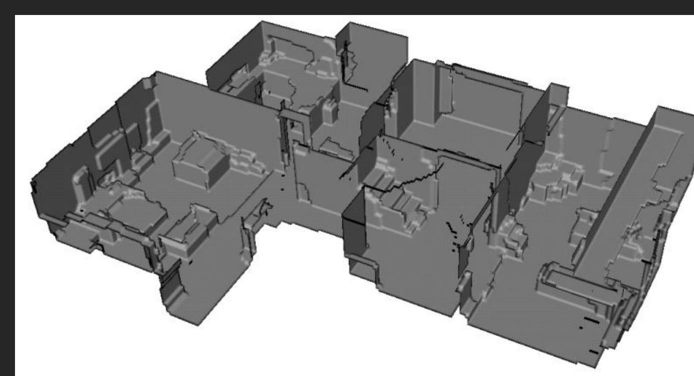
$$\boldsymbol{E} = \sum F_1(l(\boldsymbol{v})) + \sum\sum F_2(l(\boldsymbol{u}), l(\boldsymbol{v}))$$

F_1 : unary penalty

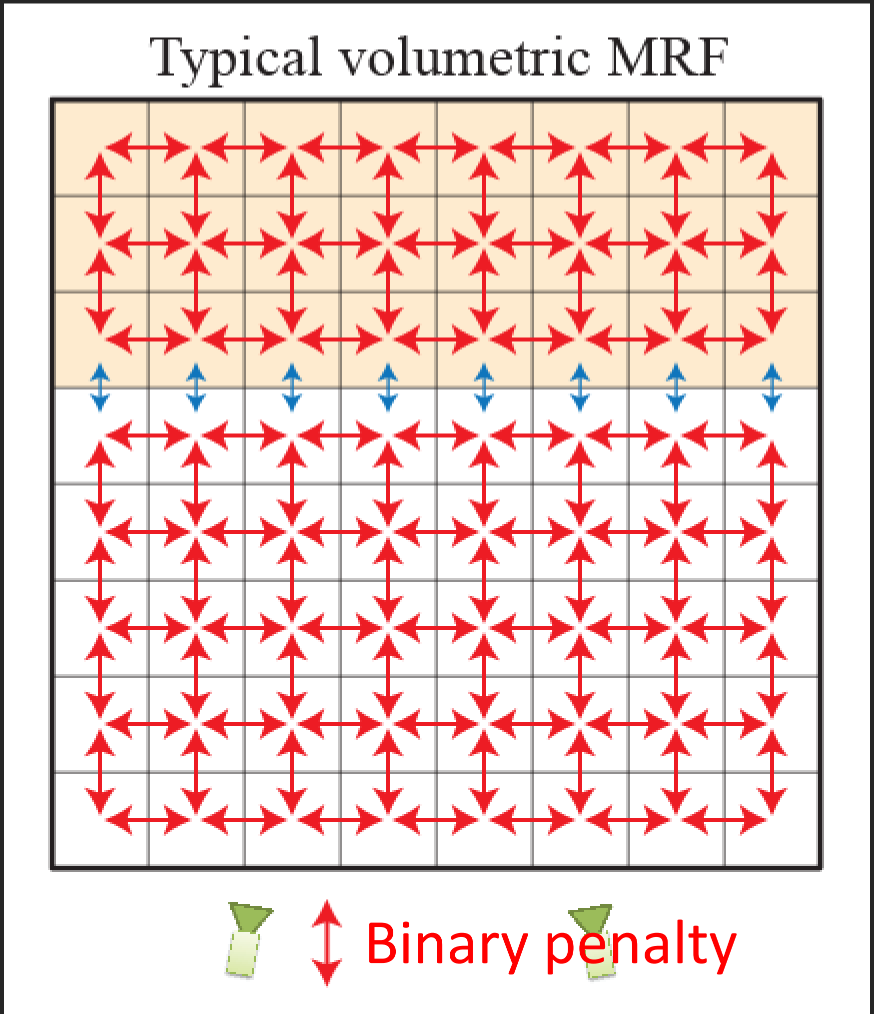
F_2 : binary penalty for neighboring voxels



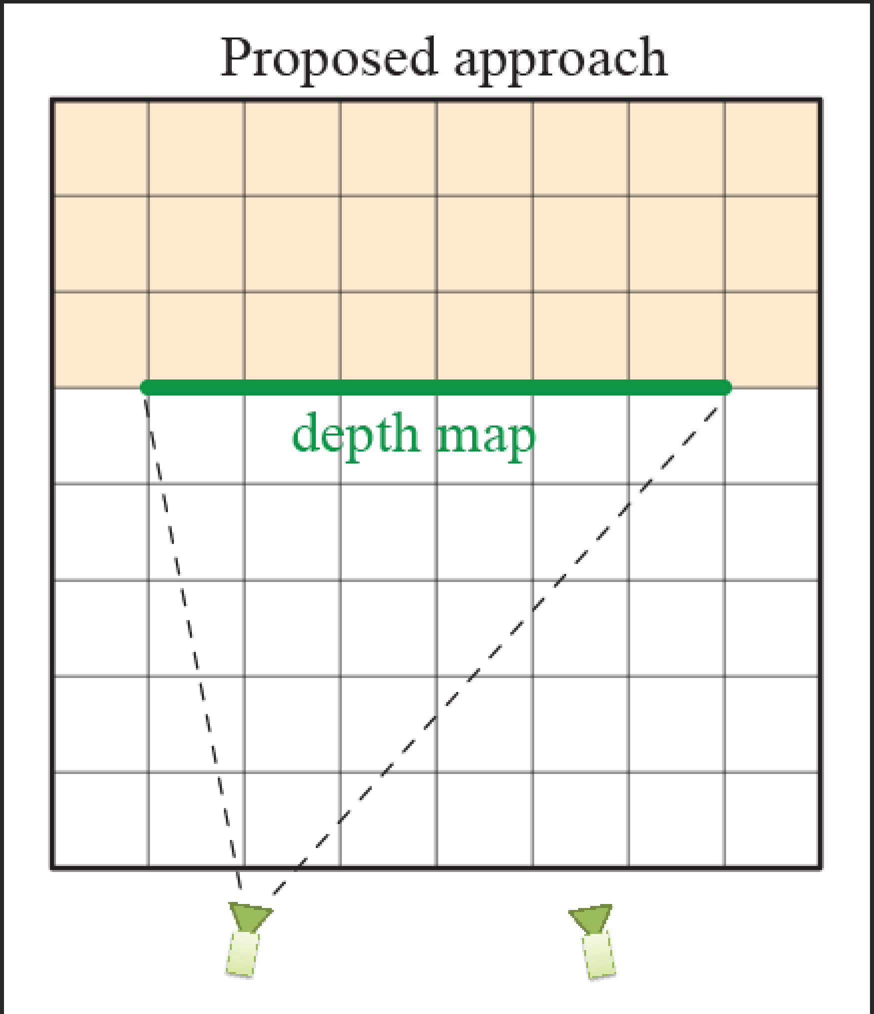




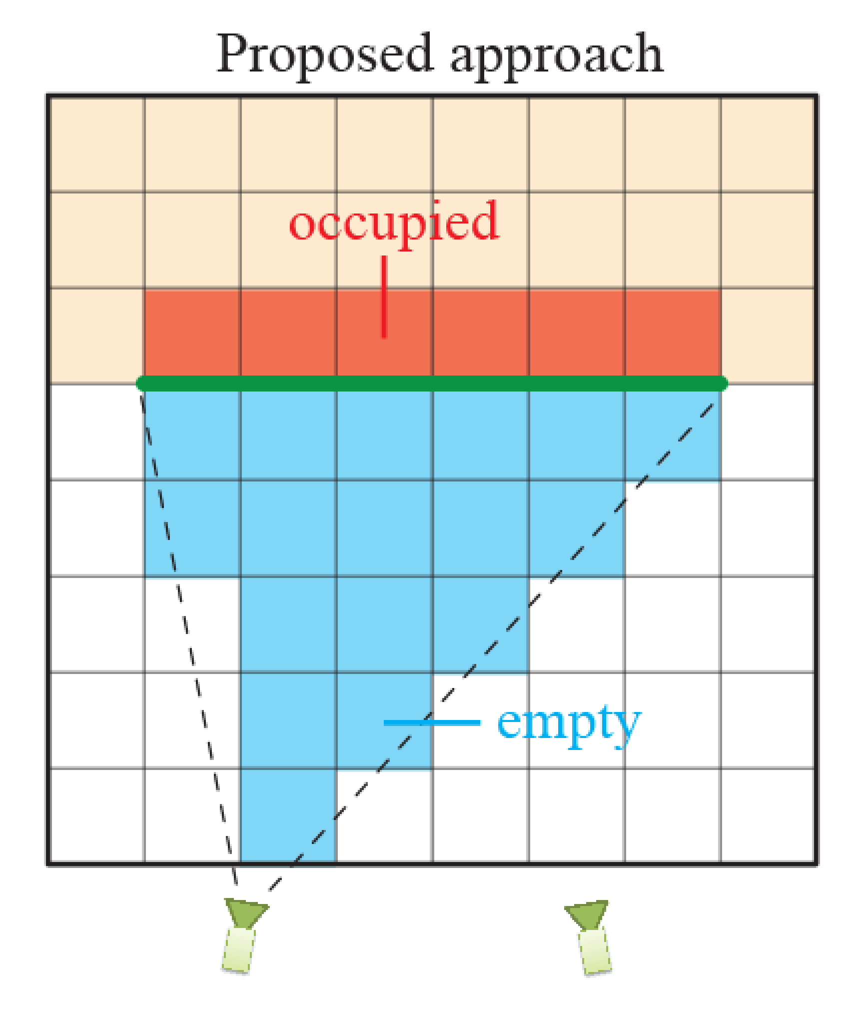
Key Feature 1 - Penalty terms



Key Feature 1 - Penalty terms

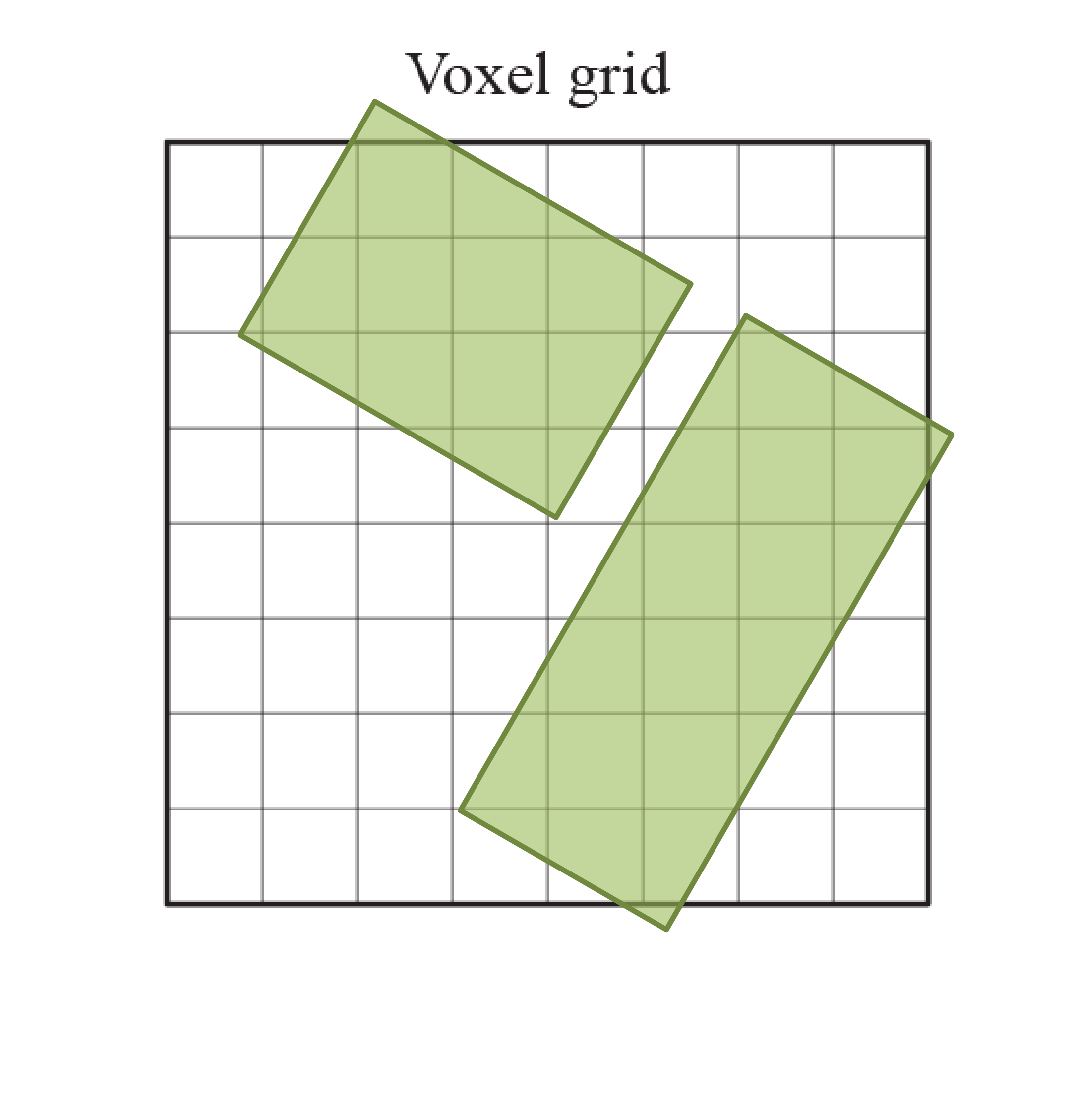


Key Feature 1 - Penalty terms



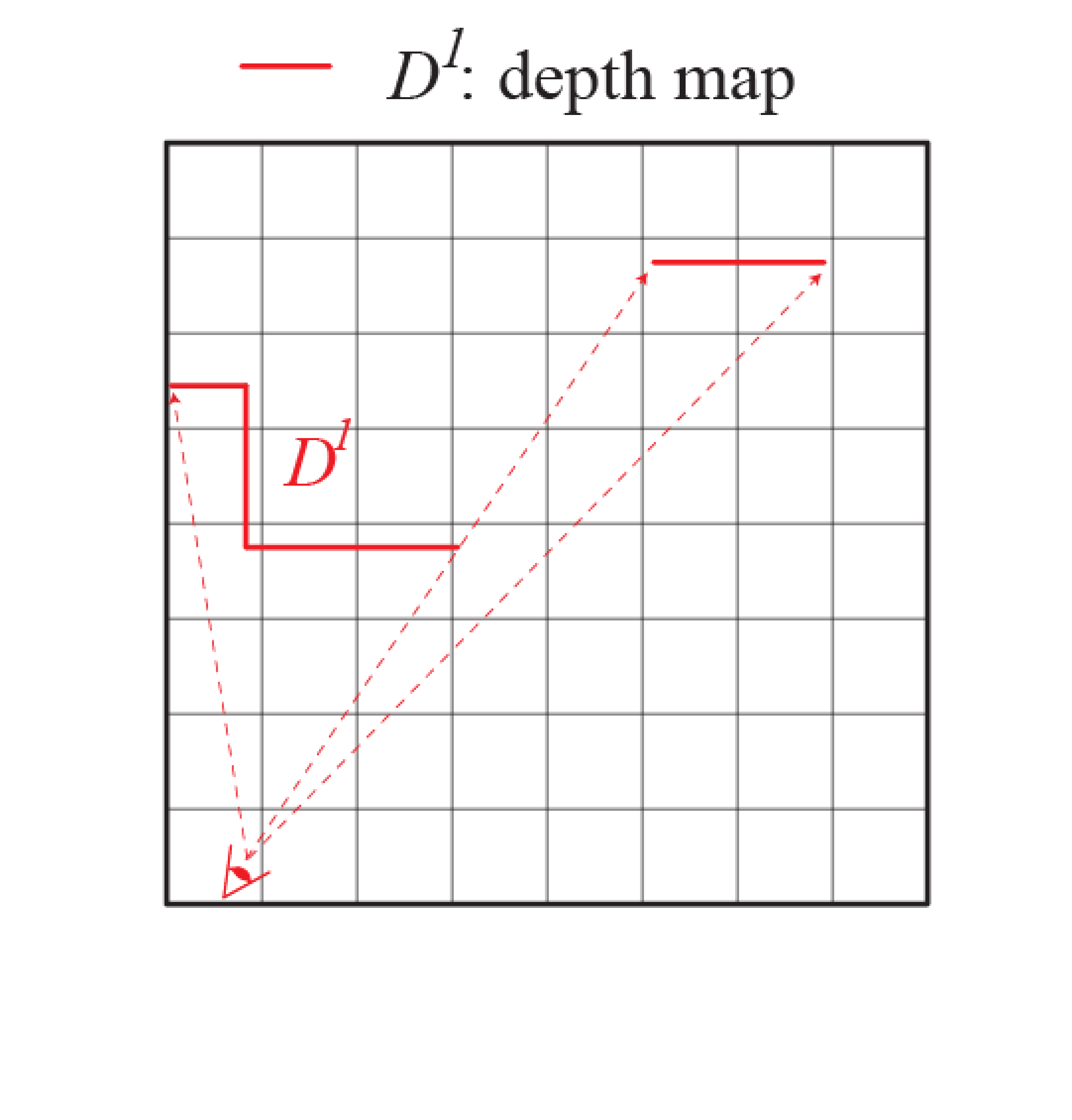
Axis-aligned Depth-map Merging

- Align voxel grid with the dominant axes



Axis-aligned Depth-map Merging

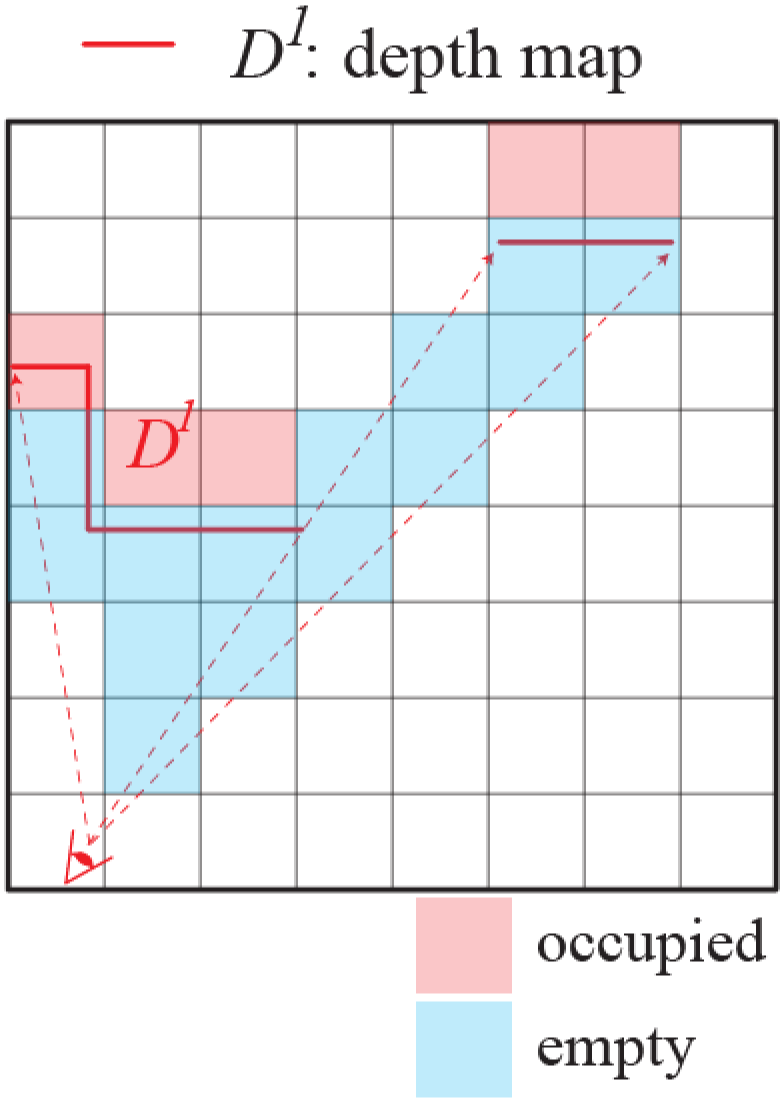
- Align voxel grid with the dominant axes
- Data term (unary)



Axis-aligned Depth-map Merging

- Align voxel grid with the dominant axes
- Data term (unary)

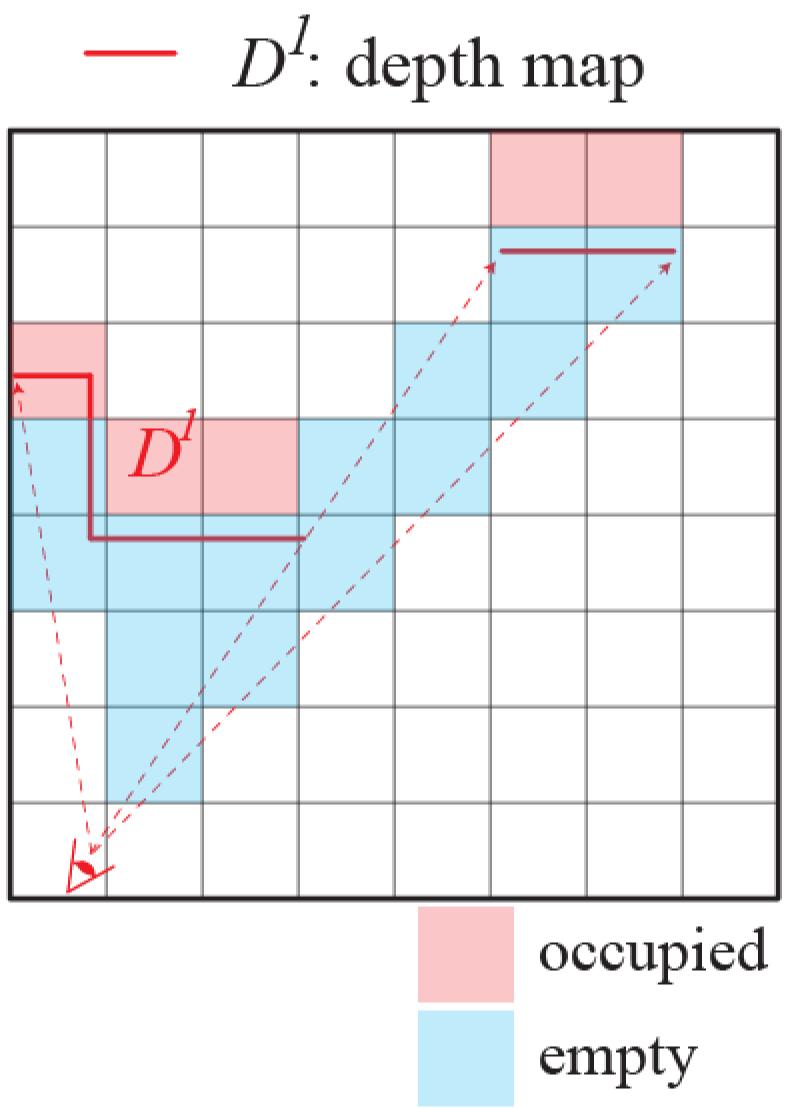
For each ***occupied*** (red) voxel ***v***
  // Decrease penalty for ***occupied***
  $F_1(l(v) = occupied) \mathrel{-}= 1;$



Axis-aligned Depth-map Merging

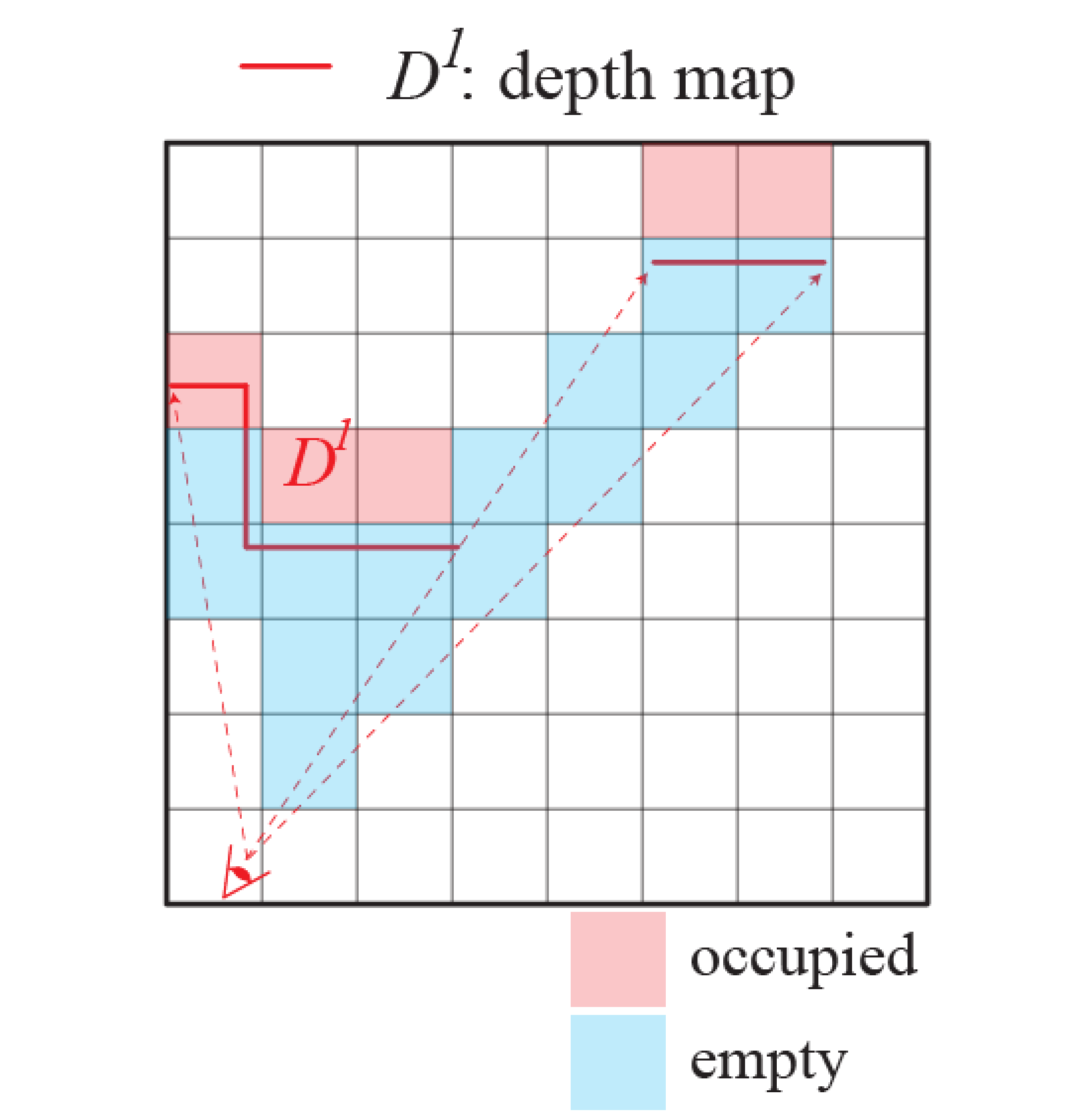
- Align voxel grid with the dominant axes
- Data term (unary)

For each ***occupied*** (red) voxel $\boldsymbol{v}$
    // Decrease penalty for ***occupied***
    $F_1(l(\boldsymbol{v}) = \textbf{\textit{occupied}}) \mathrel{-}= 1;$
For each ***empty*** (blue) voxel $\boldsymbol{v}$
    // Decrease penalty for ***empty***
    $F_1(l(\boldsymbol{v}) = \textbf{\textit{empty}}) \mathrel{-}= 1;$



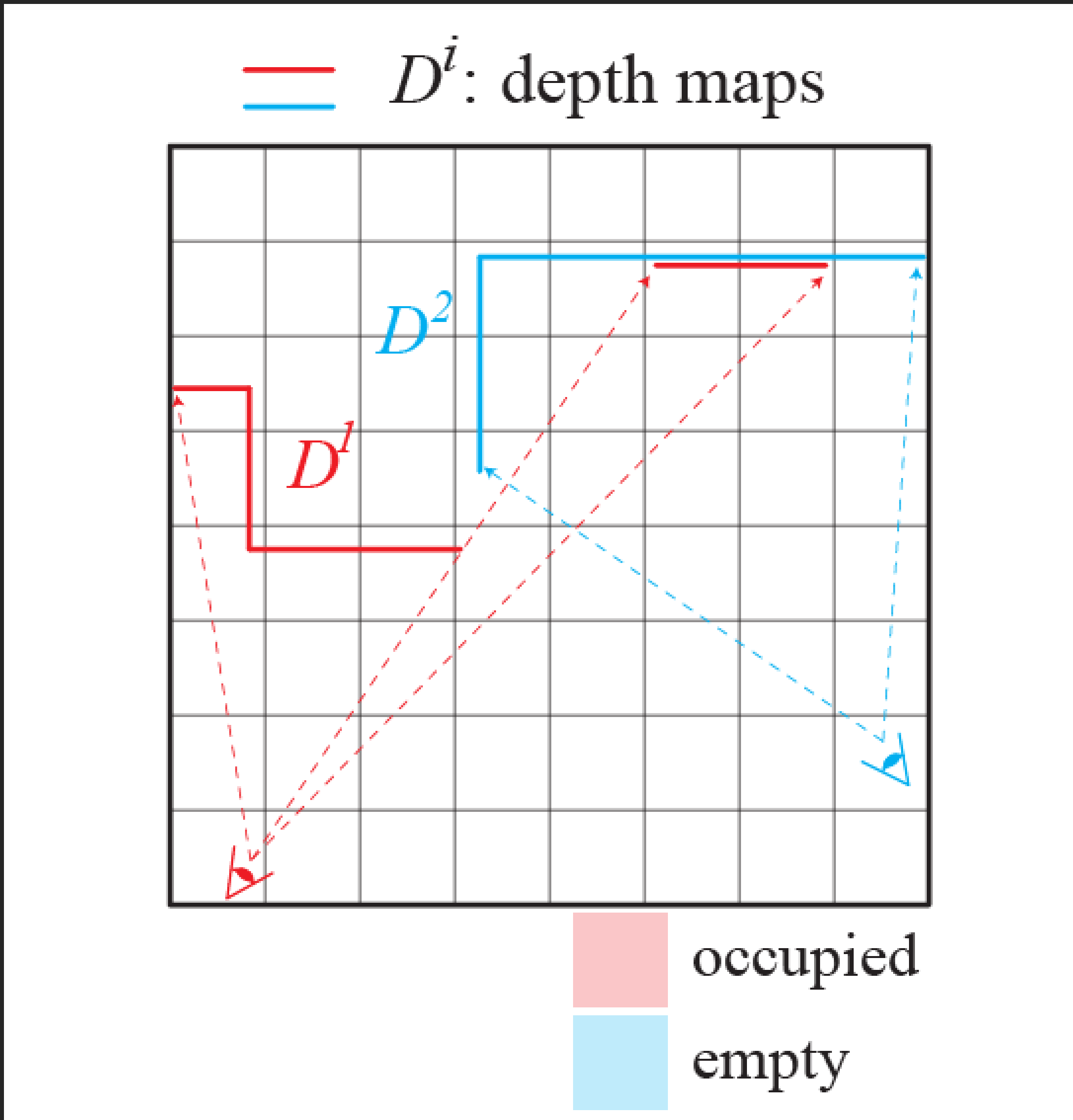
Axis-aligned Depth-map Merging

- Align voxel grid with the dominant axes
- Data term (unary)
- Smoothness (binary)



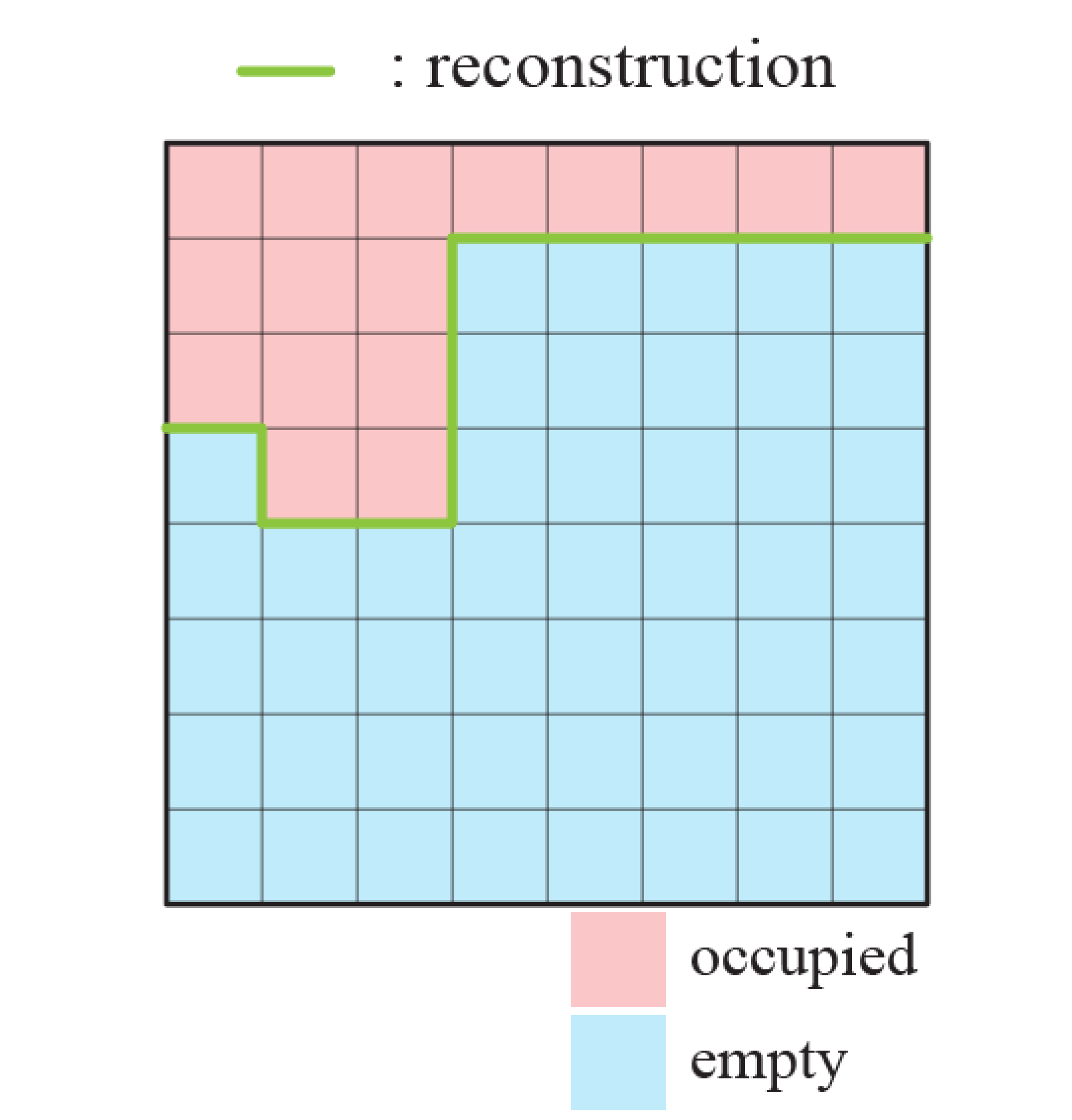
Axis-aligned Depth-map Merging

- Align voxel grid with the dominant axes
- Data term (unary)
- Smoothness (binary)



Axis-aligned Depth-map Merging

- Align voxel grid with the dominant axes
- Data term (unary)
- Smoothness (binary)
- Graph-cuts



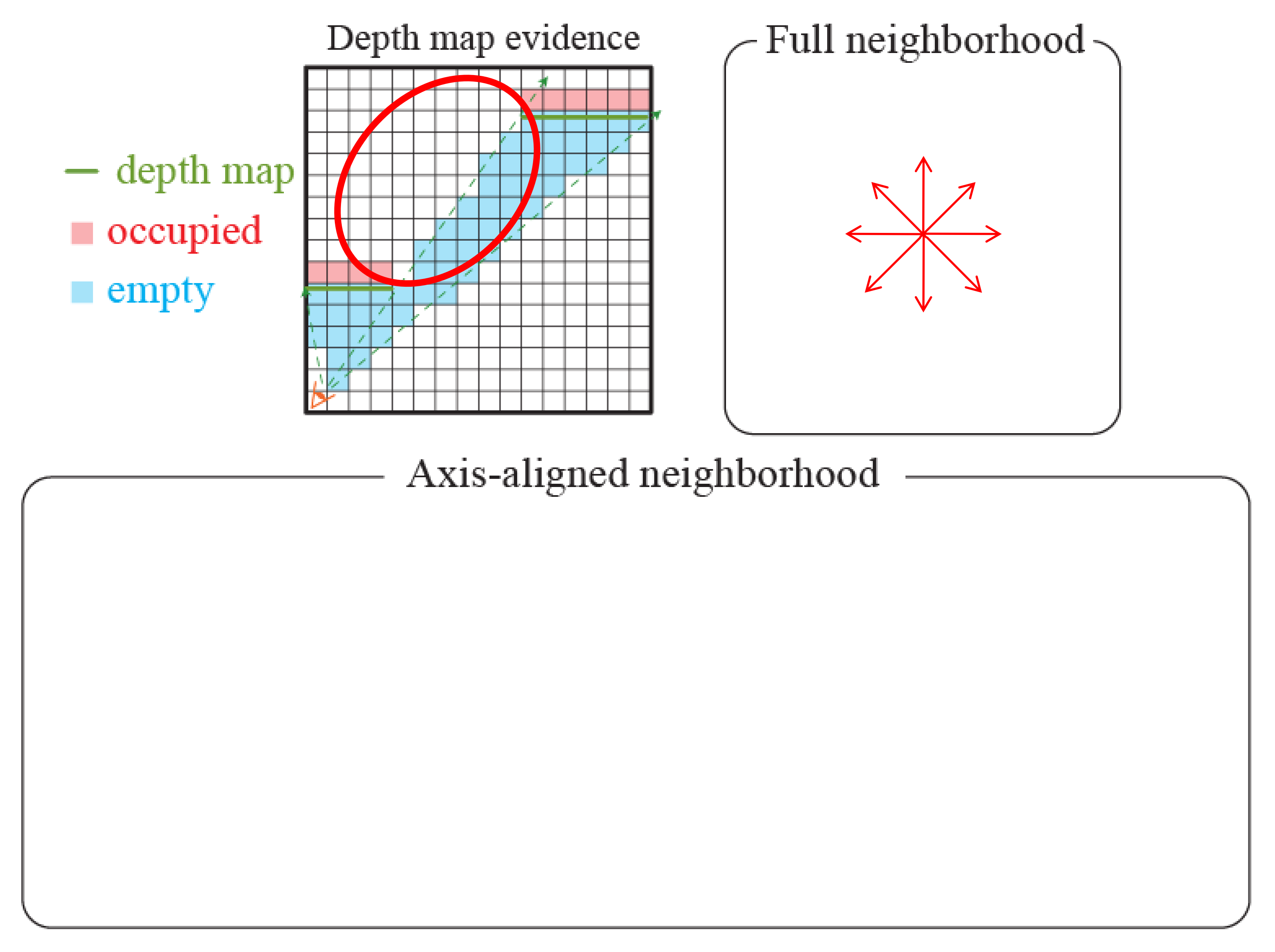
Key Feature 2 – Regularization

- For large scenes, data info are not complete

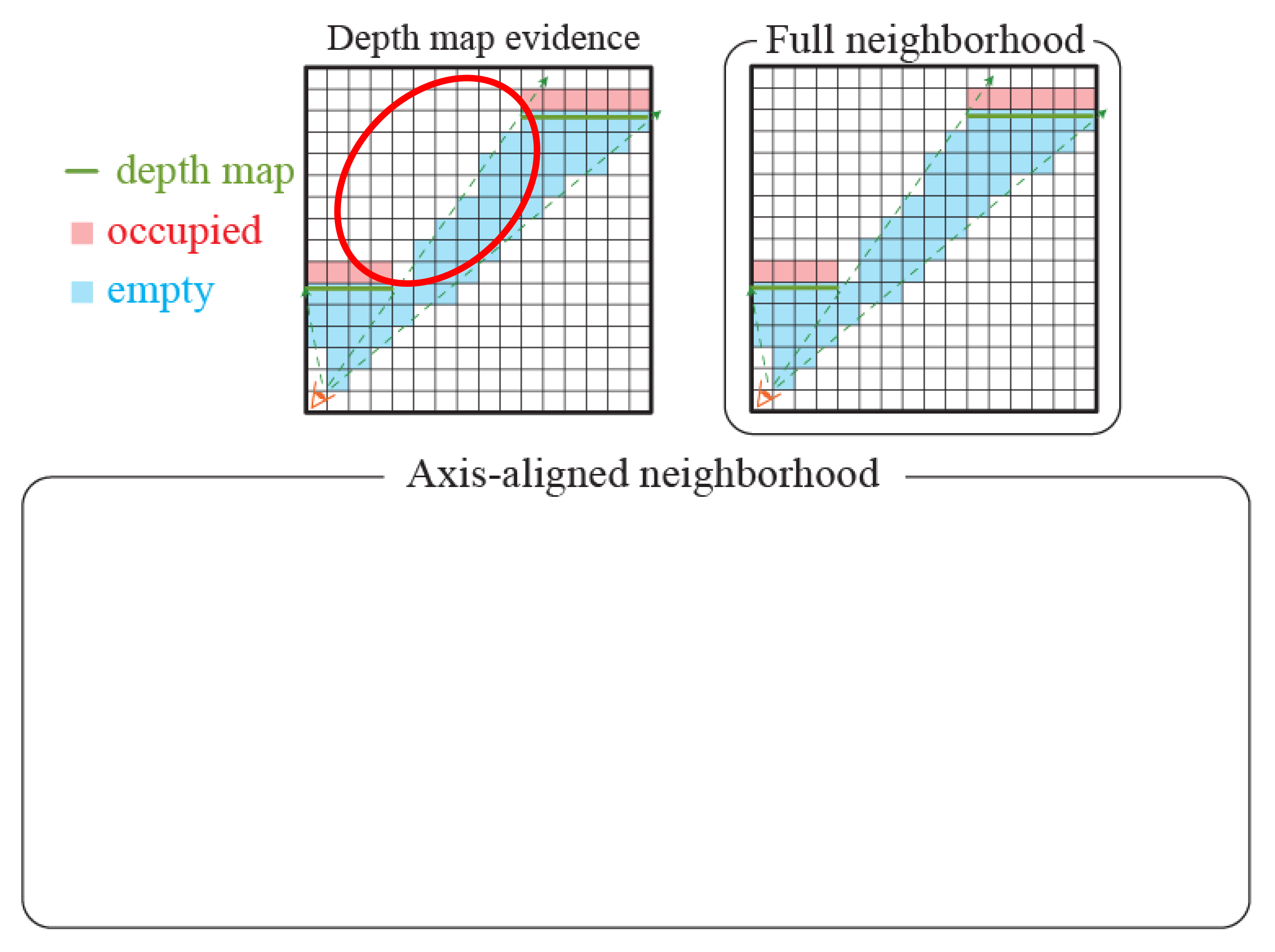
Key Feature 2 – Regularization

- For large scenes, data info are not complete
- Typical volumetric MRFs bias to general minimal surface [Boykov and Kolmogorov, 2003]
- We bias to piece-wise planar axis-aligned for architectural scenes

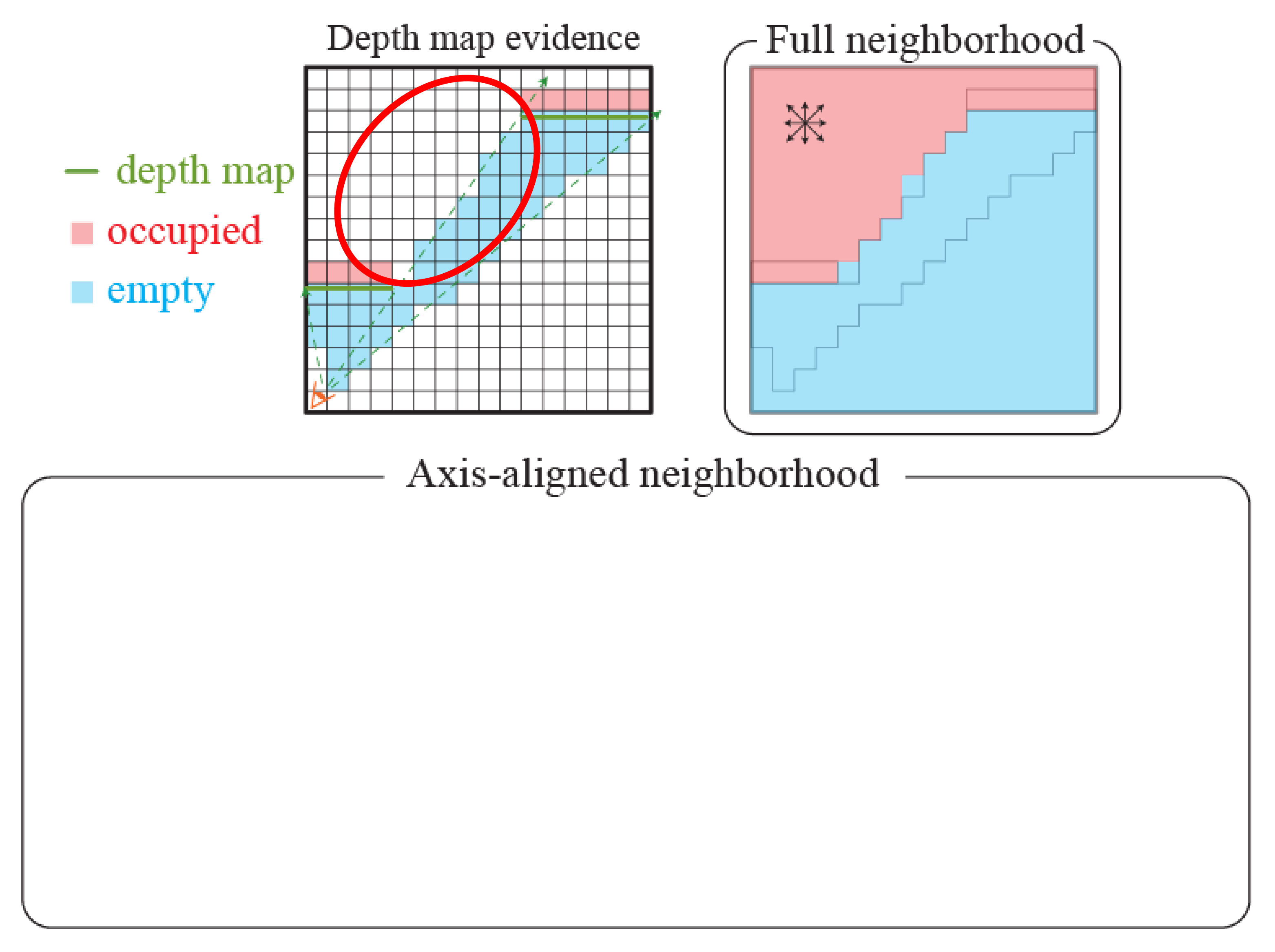
Key Feature 2 – Regularization



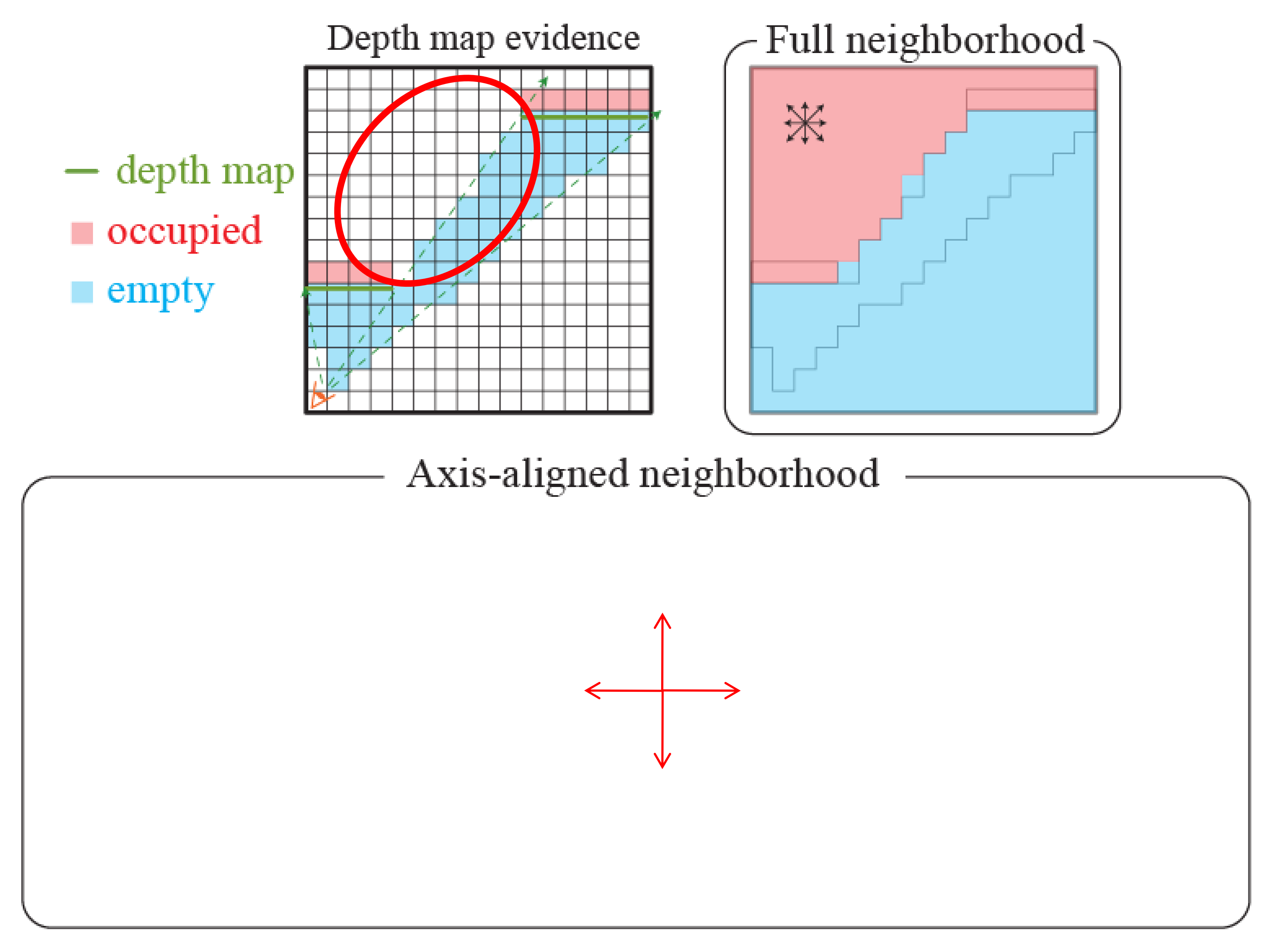
Key Feature 2 – Regularization



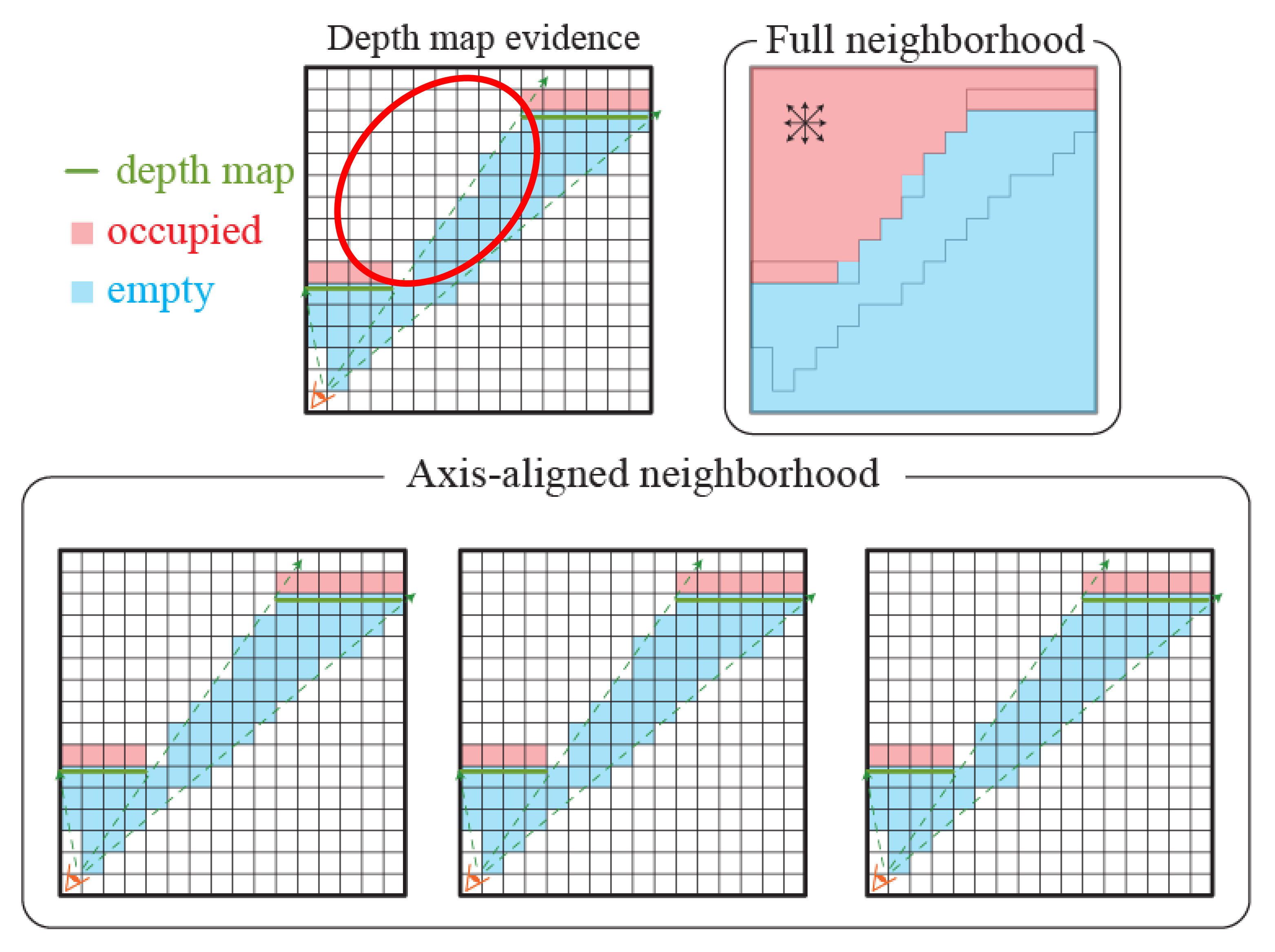
Key Feature 2 – Regularization



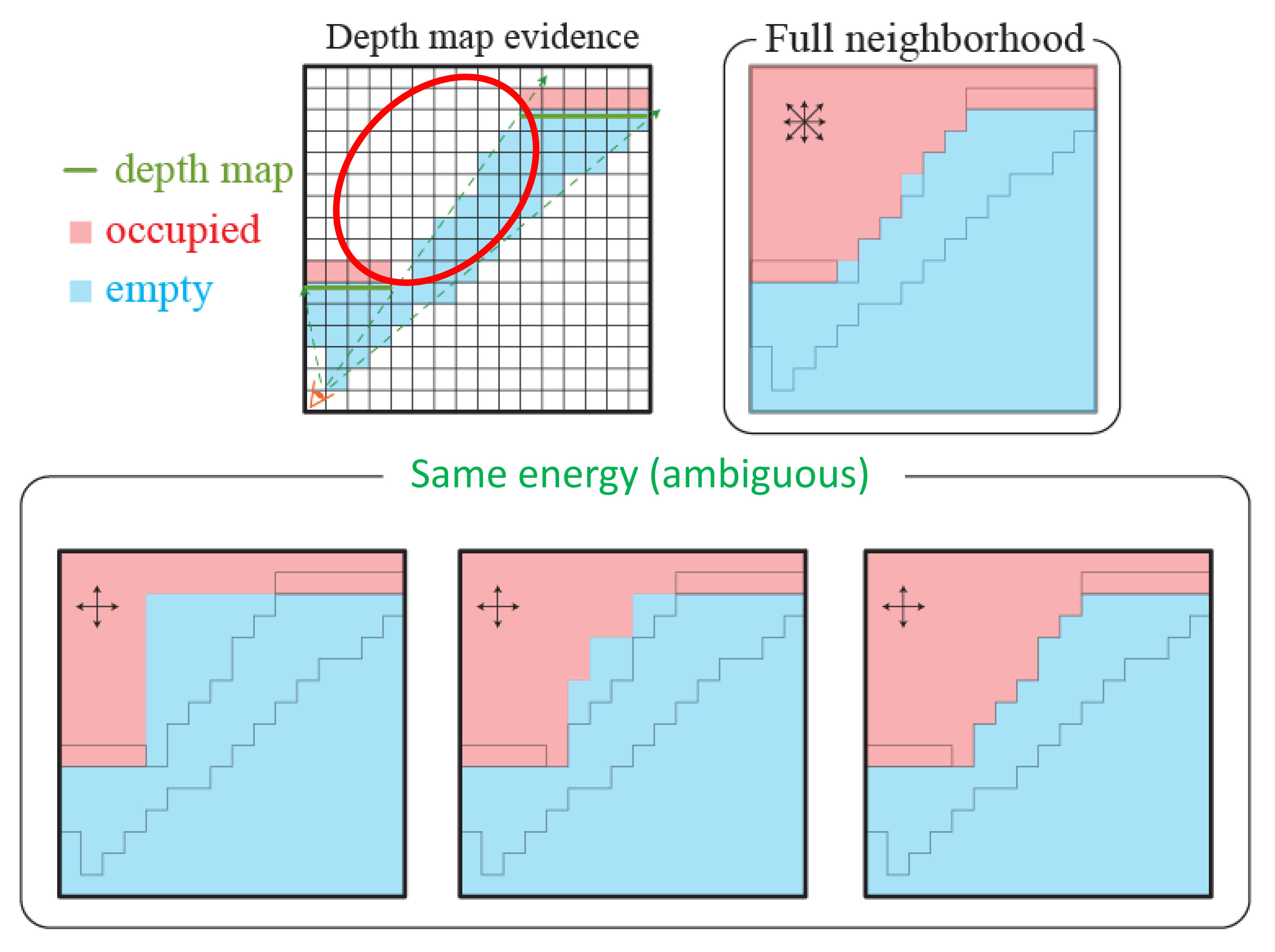
Key Feature 2 – Regularization



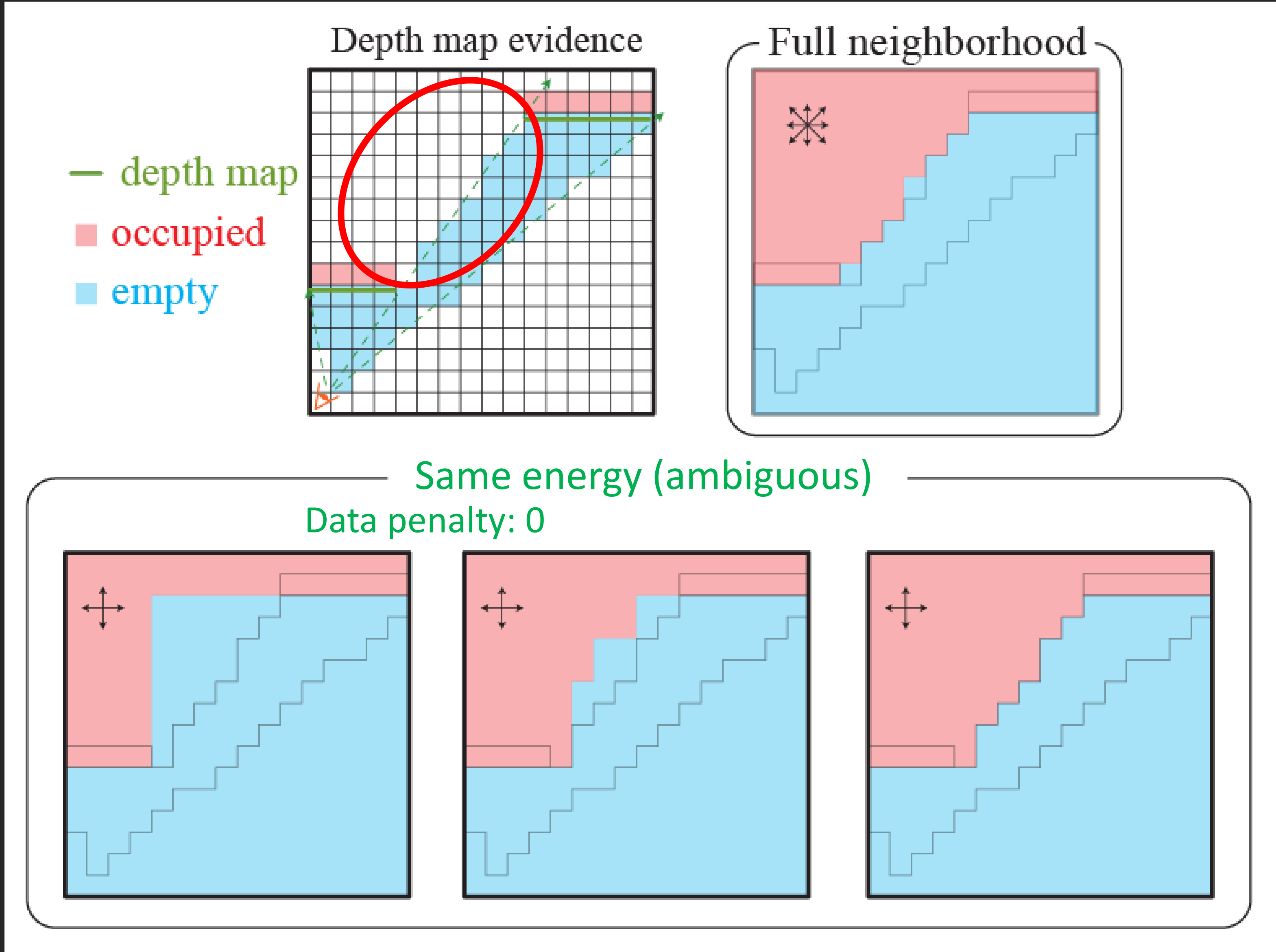
Key Feature 2 – Regularization



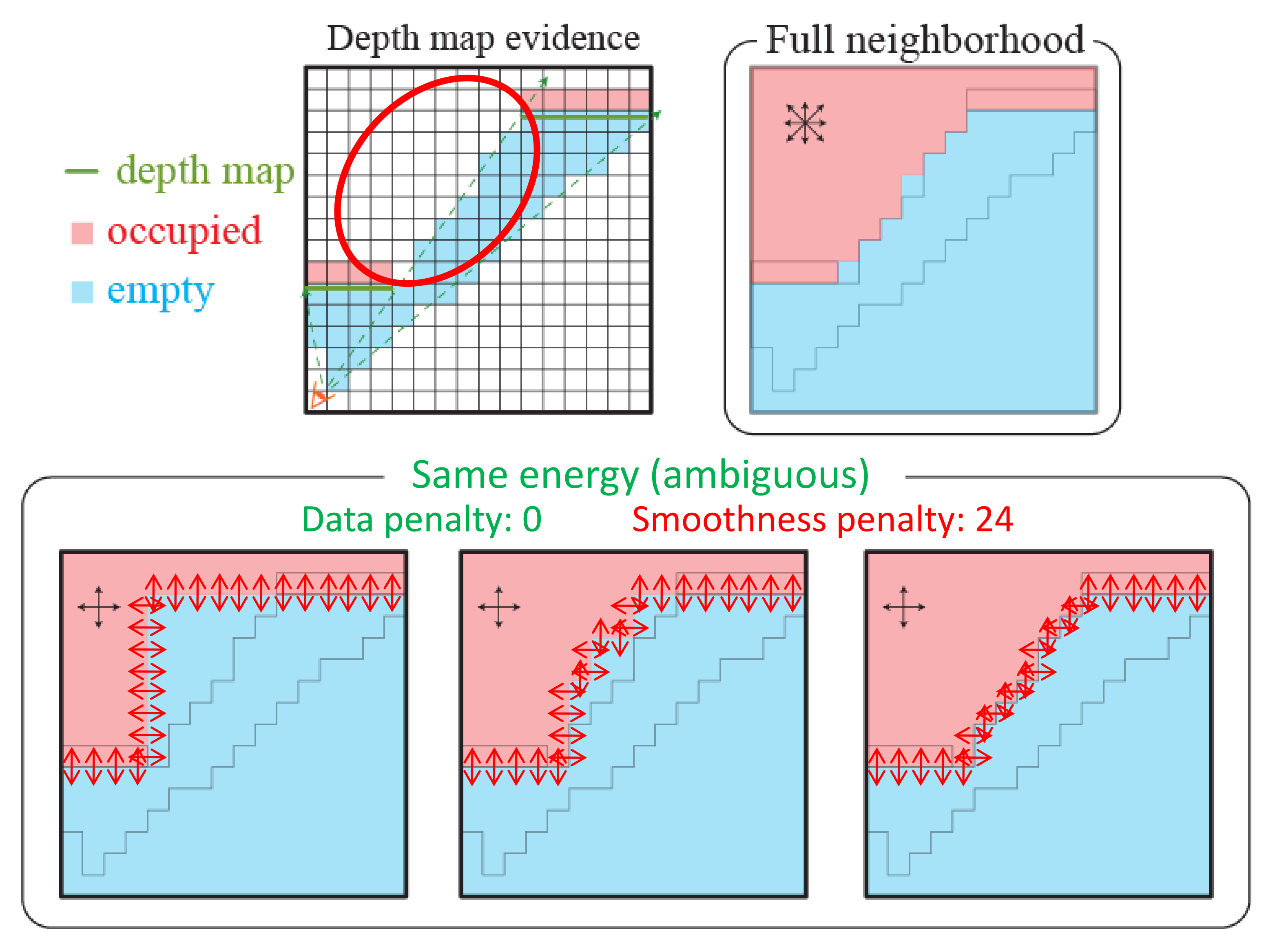
Key Feature 2 – Regularization



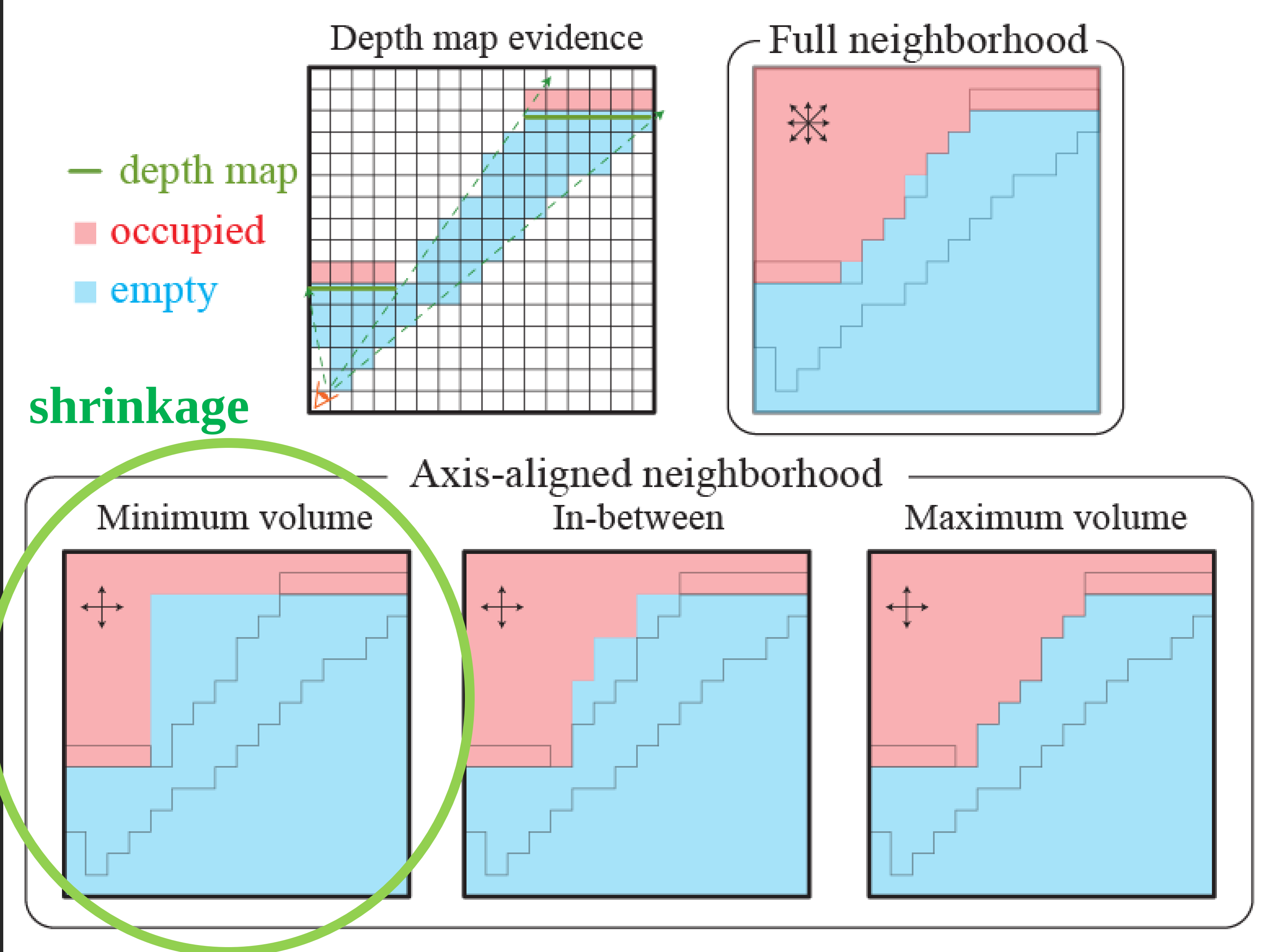
Key Feature 2 – Regularization



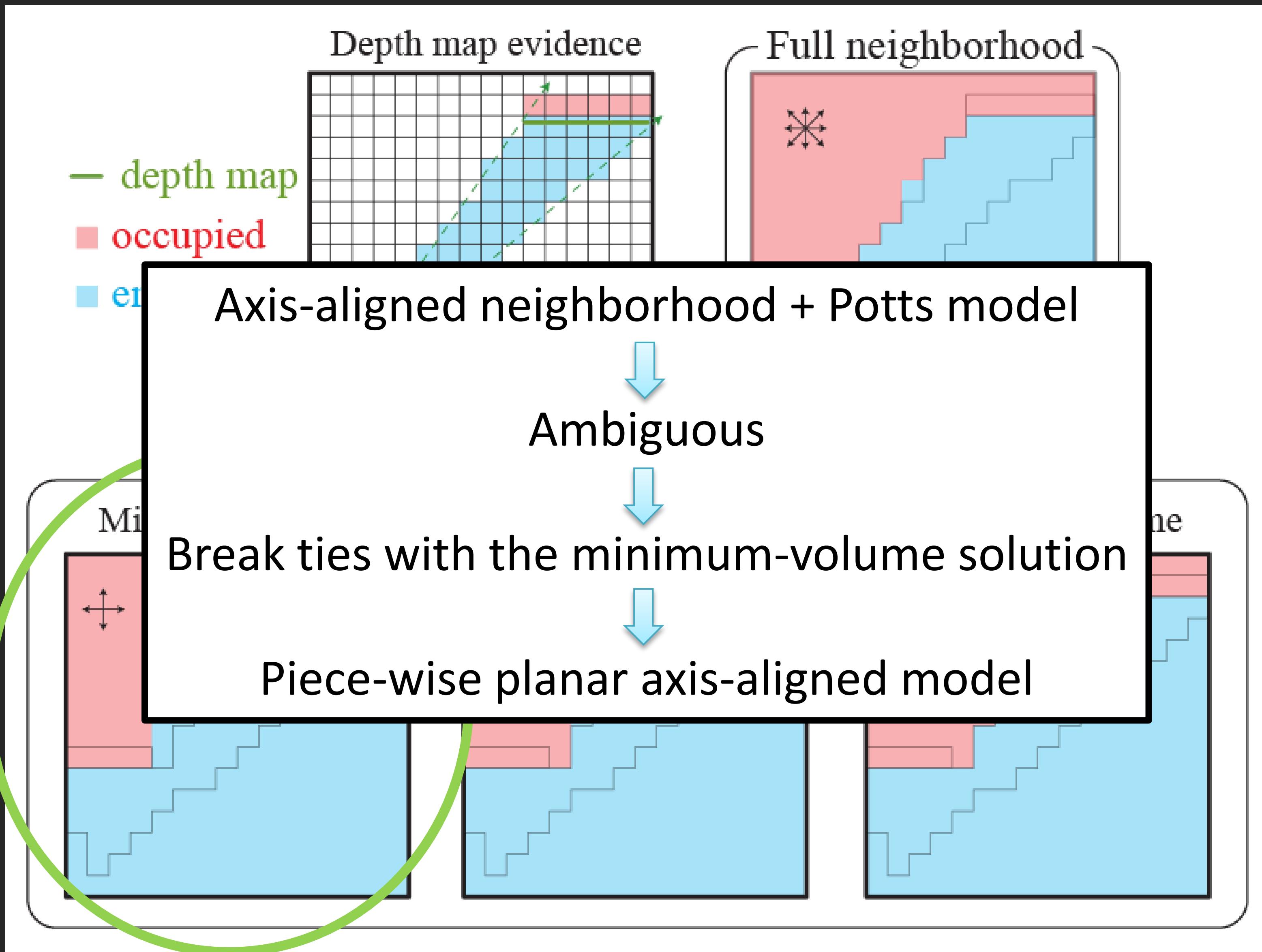
Key Feature 2 – Regularization



Key Feature 2 – Regularization



Key Feature 2 – Regularization



Axis-aligned neighborhood + Potts model

↓

Ambiguous

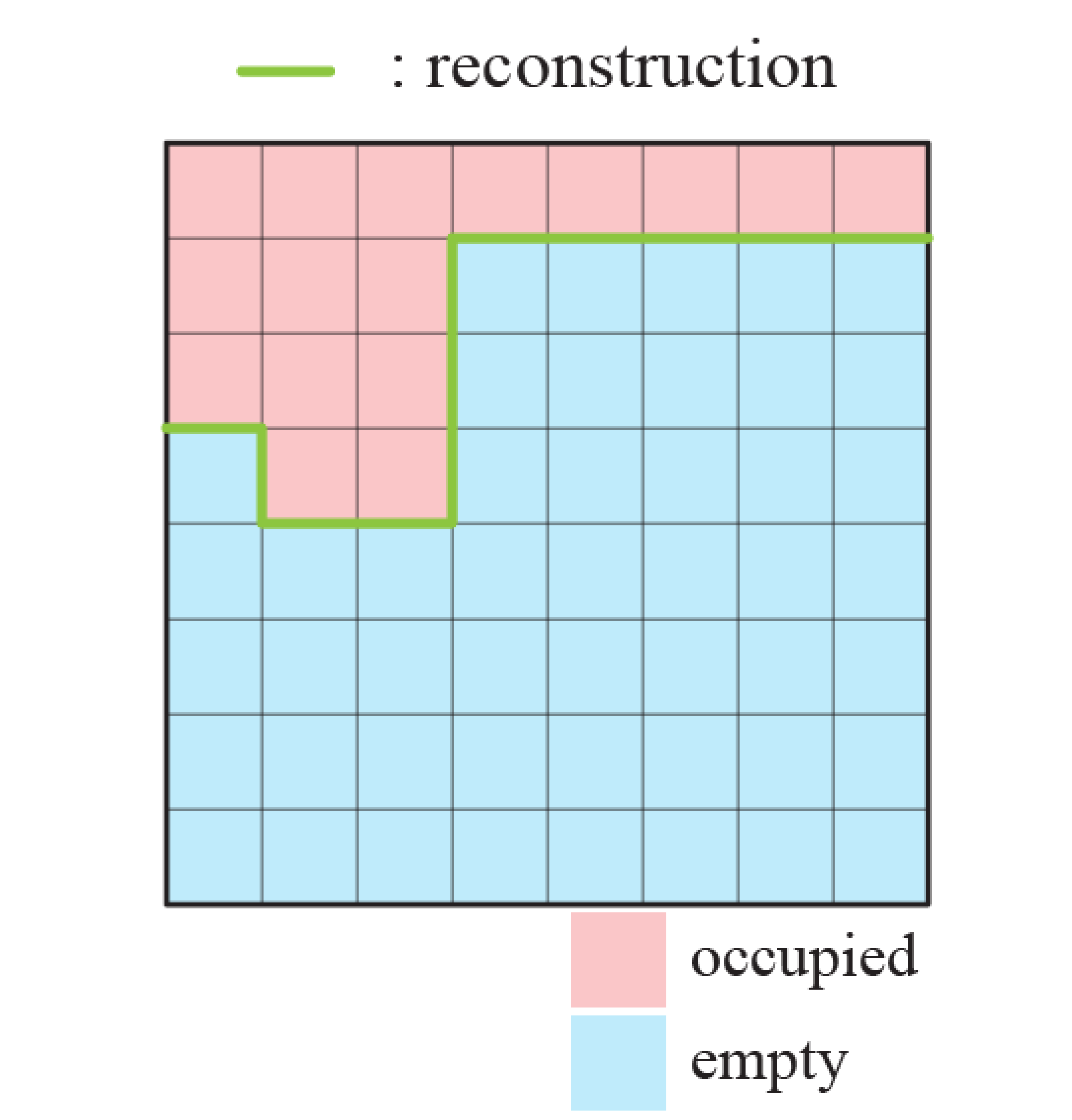
↓

Break ties with the minimum-volume solution

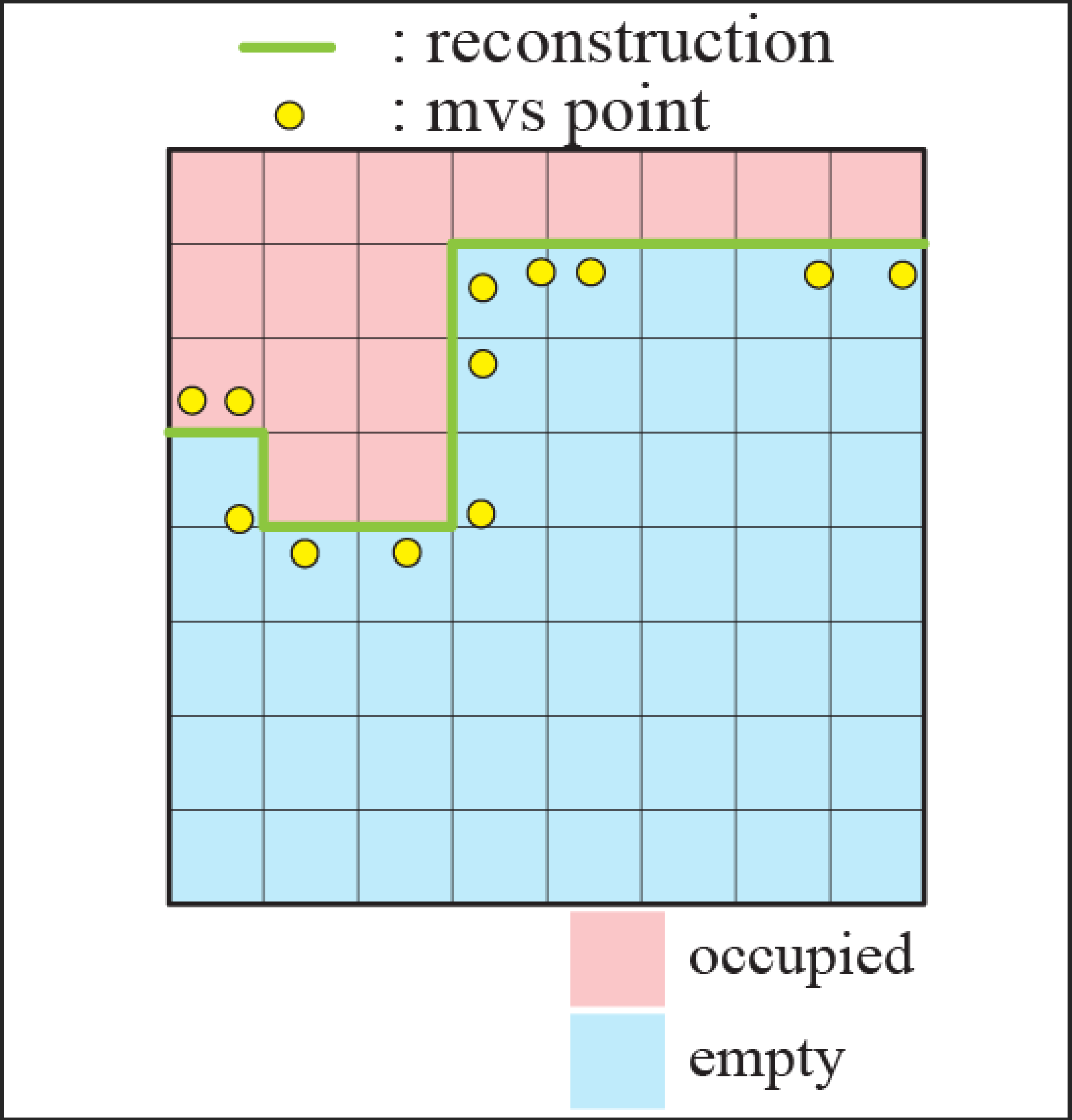
↓

Piece-wise planar axis-aligned model

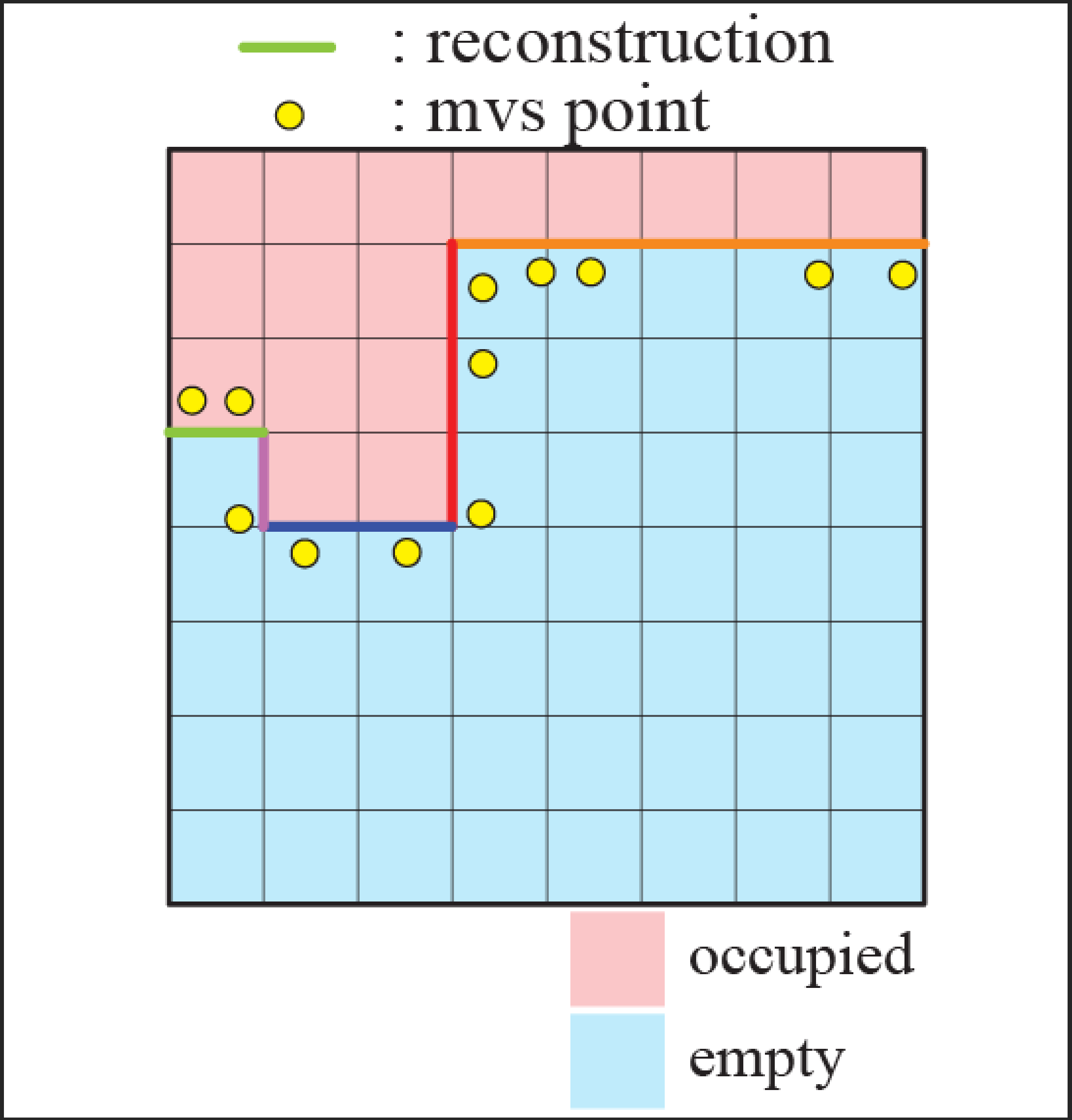
Key Feature 3 – Sub-voxel accuracy



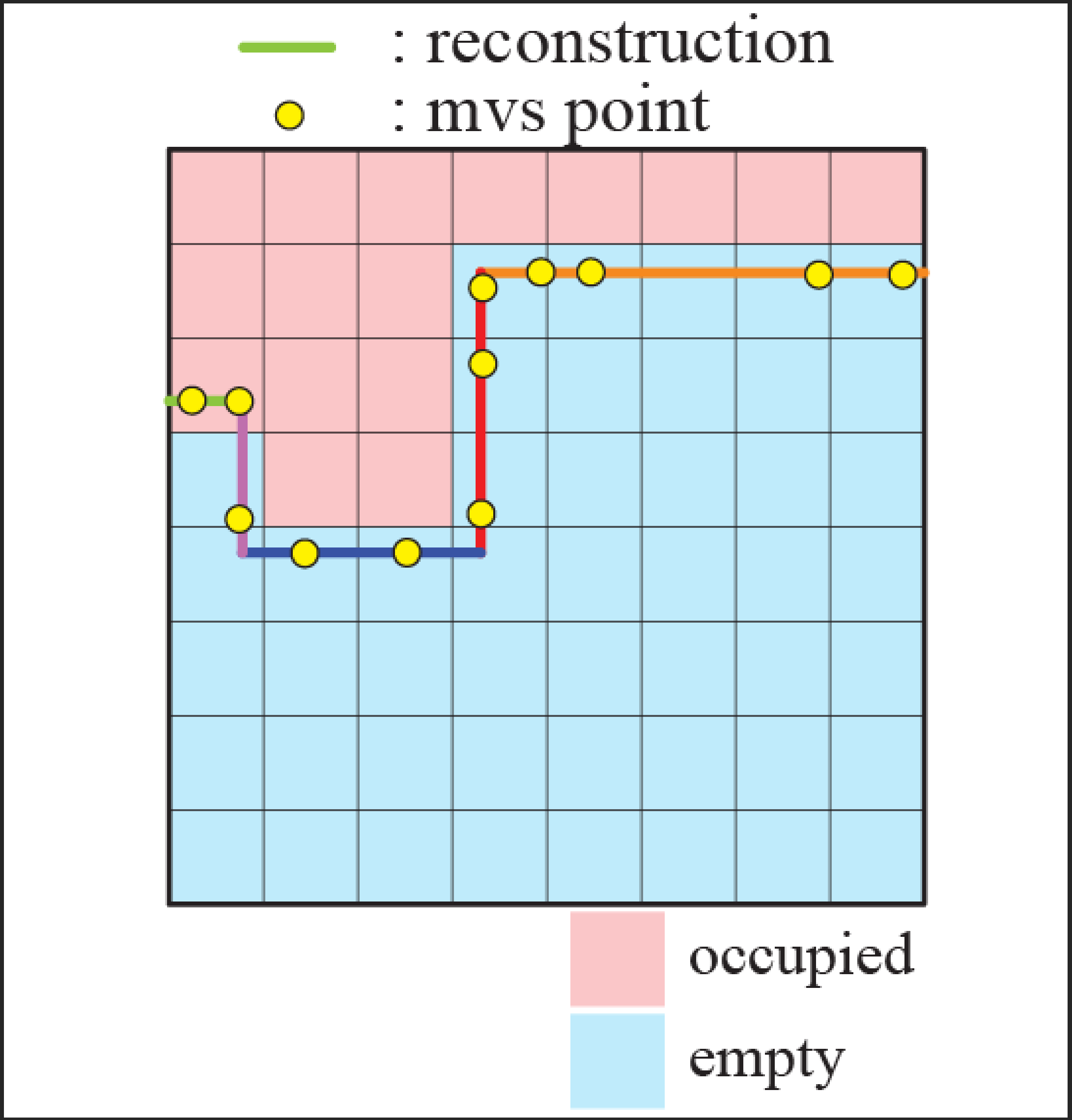
Key Feature 3 – Sub-voxel accuracy



Key Feature 3 – Sub-voxel accuracy



Key Feature 3 – Sub-voxel accuracy



Summary of Depth-map Merging

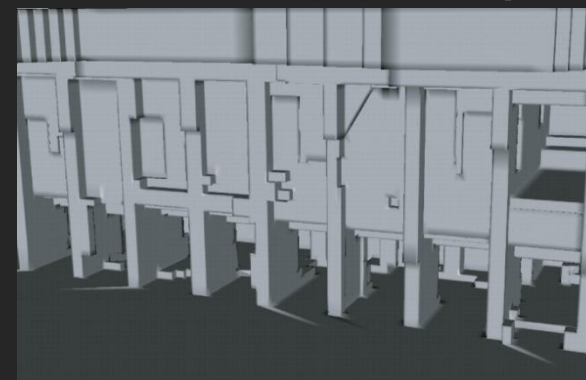
- For a simple and axis-aligned model
  - Explicit regularization in binary
  - Axis-aligned neighborhood system & minimum-volume solution
- For an accurate model
  - Sub-voxel refinement

Outline

- System pipeline (system contribution)
- Algorithmic details (technical contribution)
- Experimental results
- Conclusion and future work

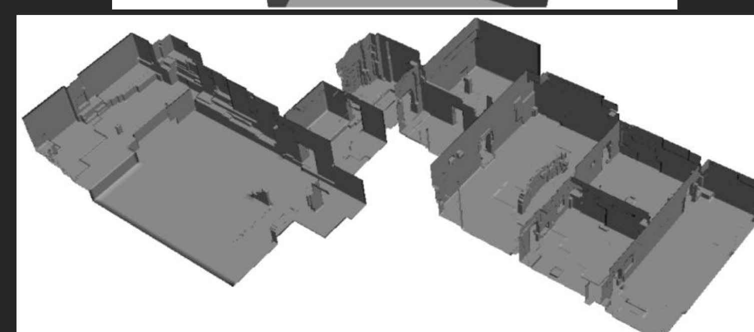
Kitchen - 22 images
1364 triangles

hall - 97 images
3344 triangles

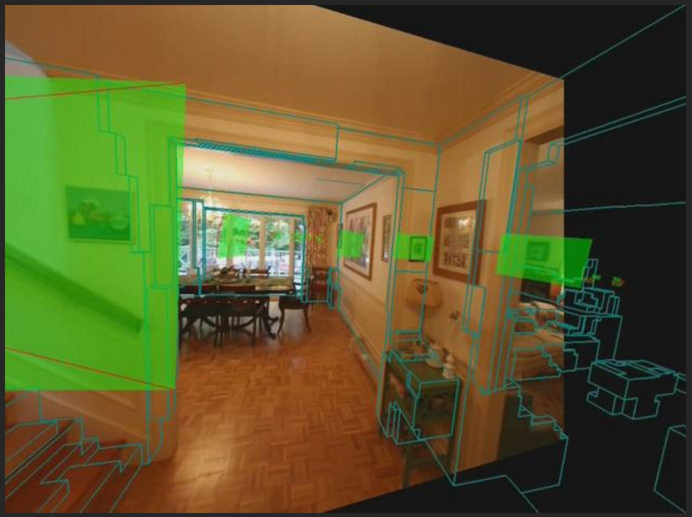


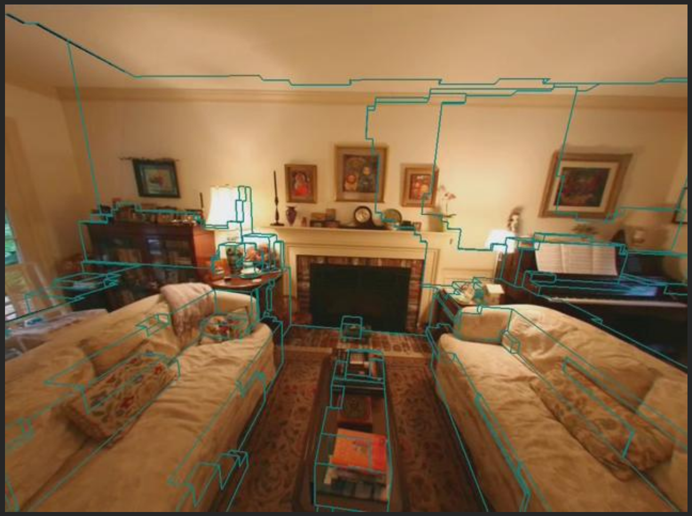
house - 148 images
8196 triangles

gallery - 492 images
8302 triangles



Demo





Questions?