Automatic 3-D Model Acquisition from Range Images

Michael K. Reed and Peter K. Allen

Computer Science Department

Columbia University

Introduction

- Objective:
  - given an arbitrary object or scene, construct a representative CAD model
- Method:
  - rangefinding sensor
  - solid modeling
  - view planning

Motivation: Applications

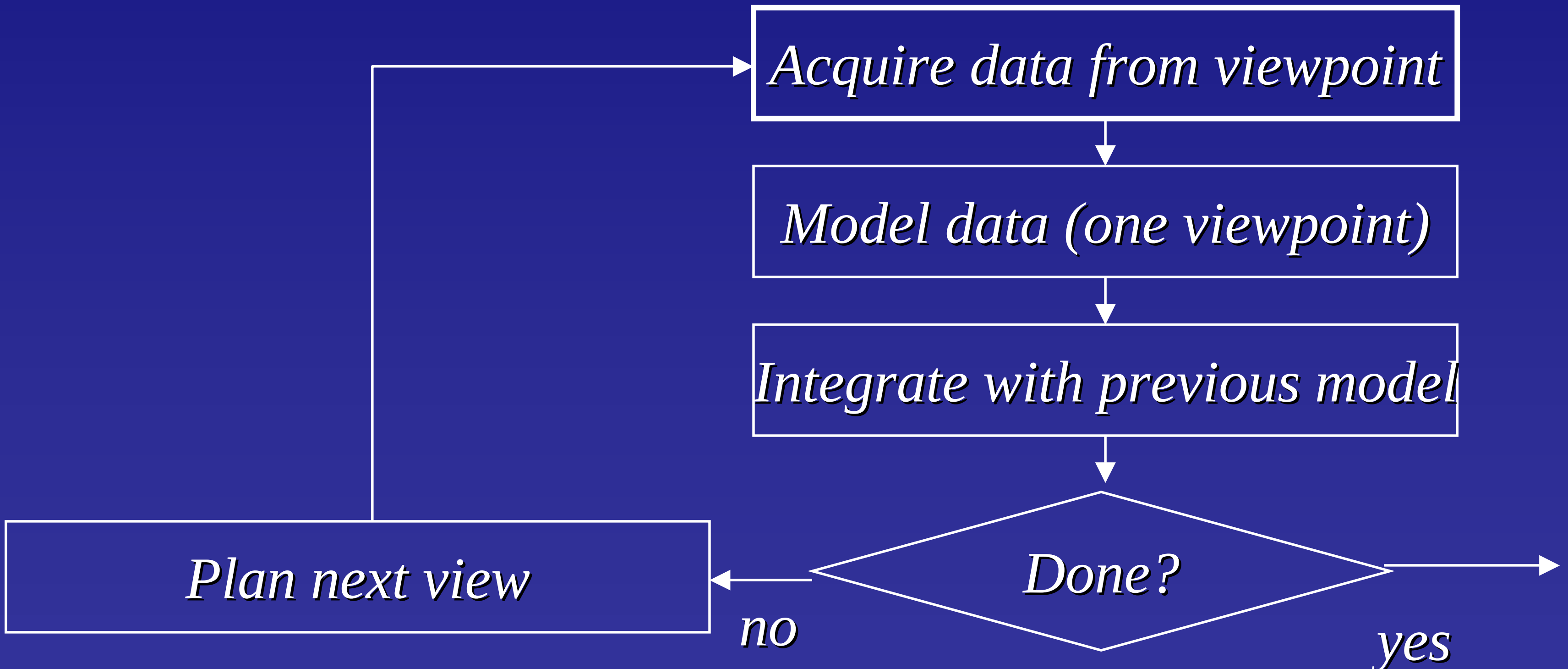
- Virtual environment generation
  - acquire model for use in VRML, entertainment, etc
- Reverse engineering
  - acquiring a model from a part copying/modification
- Part inspection
  - compare acquired model to “acceptable” model
- 3D FAX
  - transmit acquired model to remote RP machine
- Architectural site modeling

Talk Outline

- Introduction, motivation, & background
- Solid modeling of range images
- Sensor viewpoint planning
- Conclusions & future research

An Introduction to a Solution

- How do people solve this problem?
- An algorithm:



Previous Work

- Computer Vision:
  - oriented towards recognition task
- Computer Graphics:
  - oriented towards recovery task

  - [Dickenson et al. 1994] [Wu & Levine 1994]
  - [Thompson et al. 1996] [Hoppe 1994] [Fua 1992]
  - [Connolly 1989] [Tarbox & Gottshlich 1995]
  - meshes & zippering: [Turk 1994] [Rutishauser ‘94]
  - meshes & iso-surface extraction: [Curless 1996]

Previous Work: Issues

- Final model is not closed
- Single object only
- Shape and topological restrictions
- Assumed complete surface sampling
- No modeling of scene occlusion

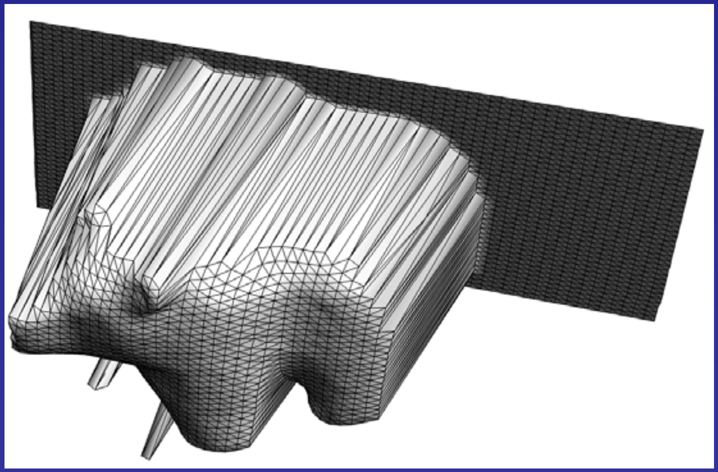
Modeling Desiderata

- no topological constraints on object / scene
- resulting model is “watertight”
- incremental model improvement
- model acquisition in few views
- ability to support sensor viewpoint planning

- Our method attempts to deal with each of these issues

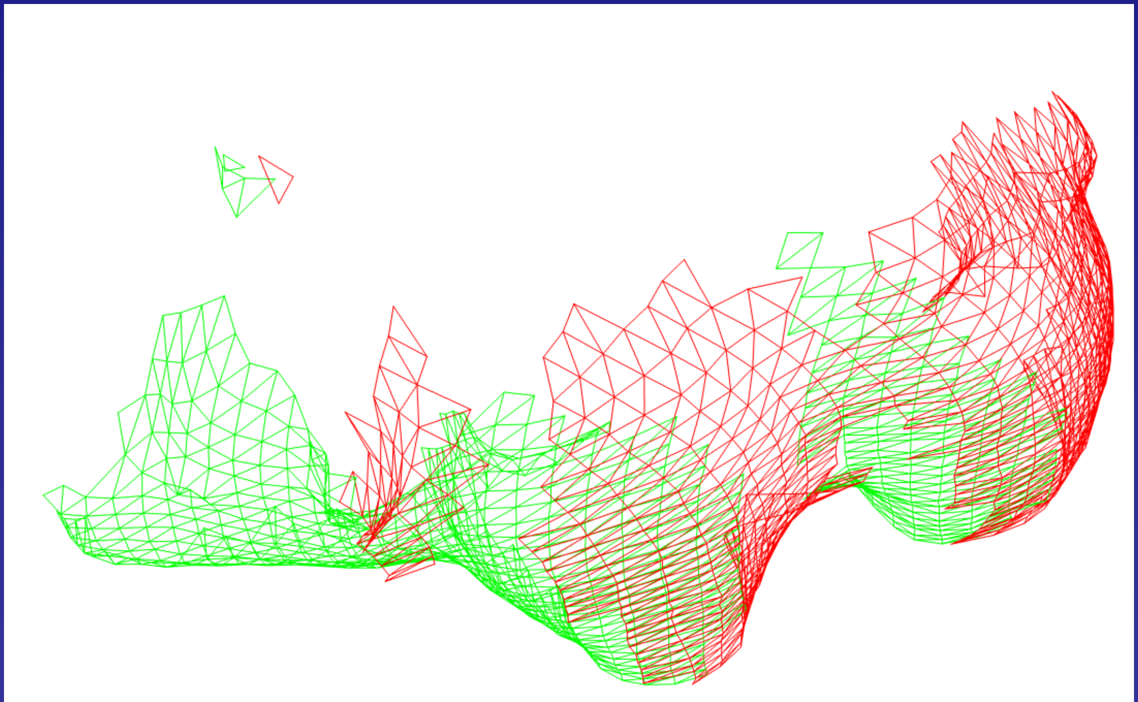
Modeling with Mesh Surfaces

- Advantages:
  - able to accurately model many shapes
  - effective at varying resolutions
  - simple!



Modeling with Mesh Surfaces

- Integration:
  - find overlap between 2 meshes
  - clip & retriangulate

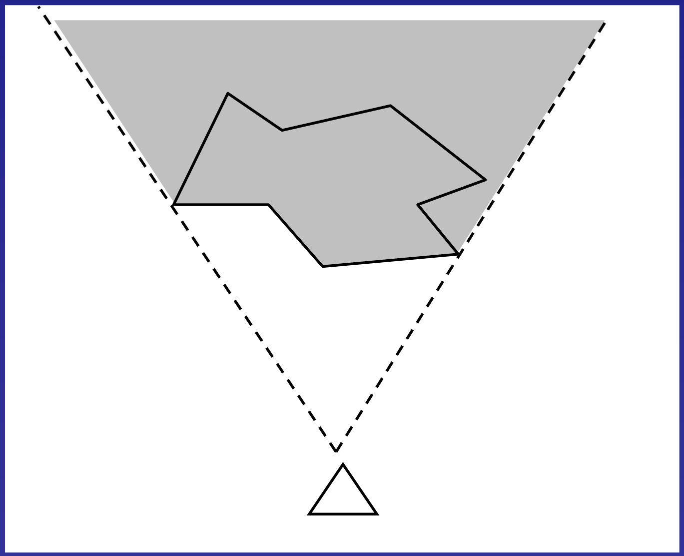


Modeling with Mesh Surfaces

- Issues:
  - not closed until surface is completely imaged
  - no representation of space occluded from sensor
  - integration of non-manifold surfaces not well-defined
- Our solution: use mesh surface & volume!
  - build solid model from each range image
  - represent both imaged surface & occluded volume
  - integrate using set intersection

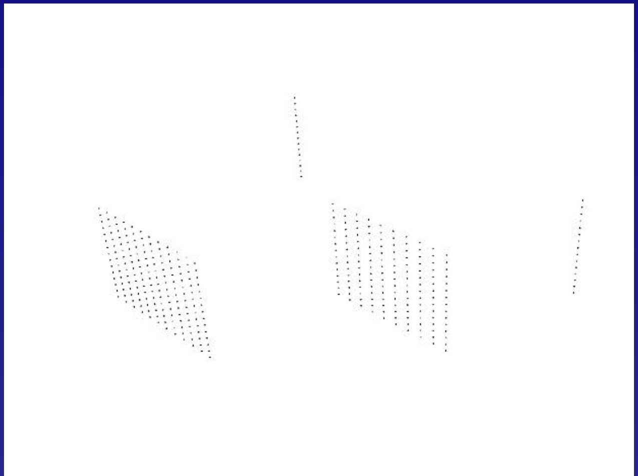
Sensor Model

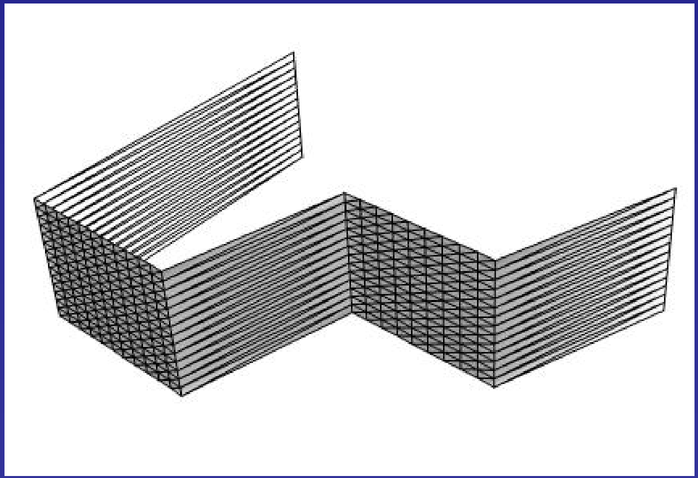
- Rangefinder acts as point light source in plane
- Motion of rangefinder is linear, perpendicular to scanning plane



Sensor Imaging Characteristics

- Range image:
  - rectangular sampling
- Image has structure:
  - surface elements
  - occlusion elements

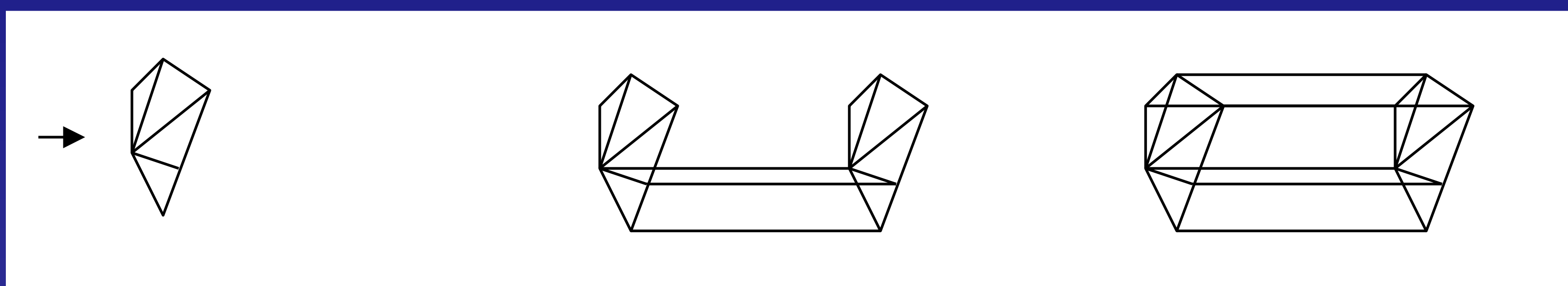


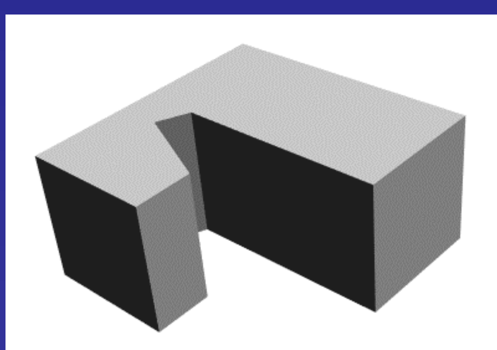
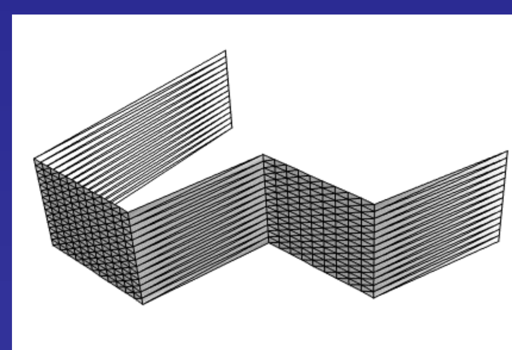
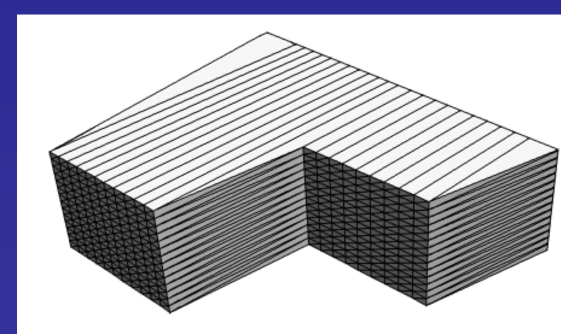


Constructing the Solid

- Sweep mesh surface *M* to form a solid *S* by:

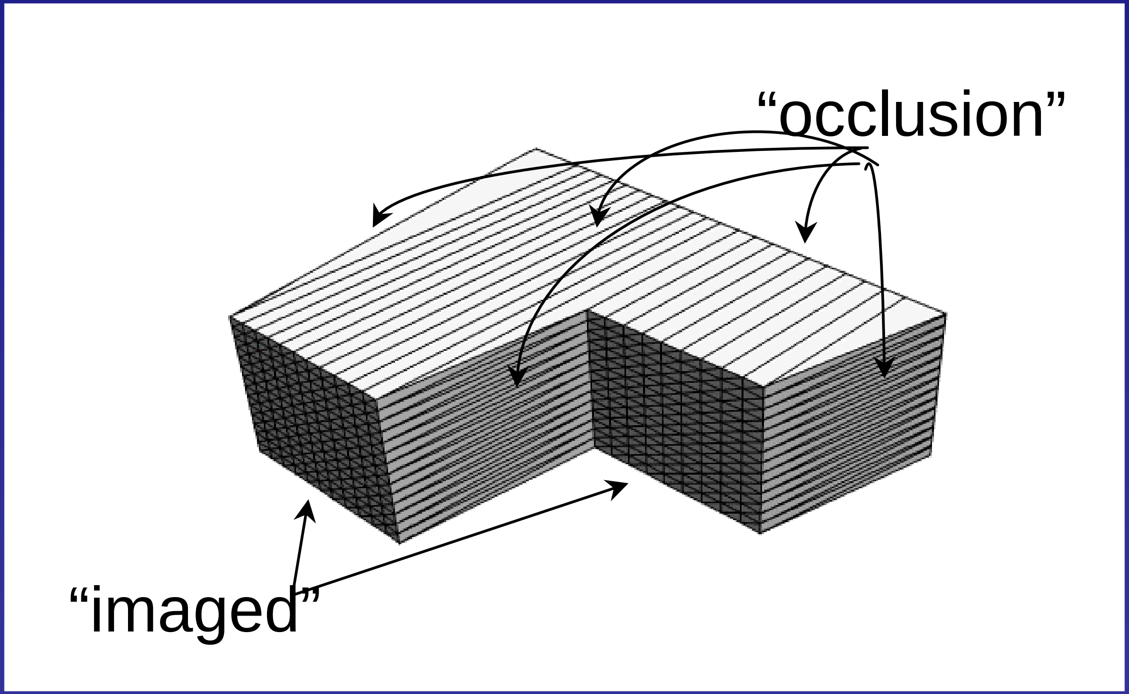
$$S = \bigcup_{\forall m} extrude\,(m),\; m \in M$$



Surface Type Attributes

- Surfaces in the model are tagged:
  - “imaged” for those directly imaged by the sensor
  - “occlusion” for those due to scene occlusion



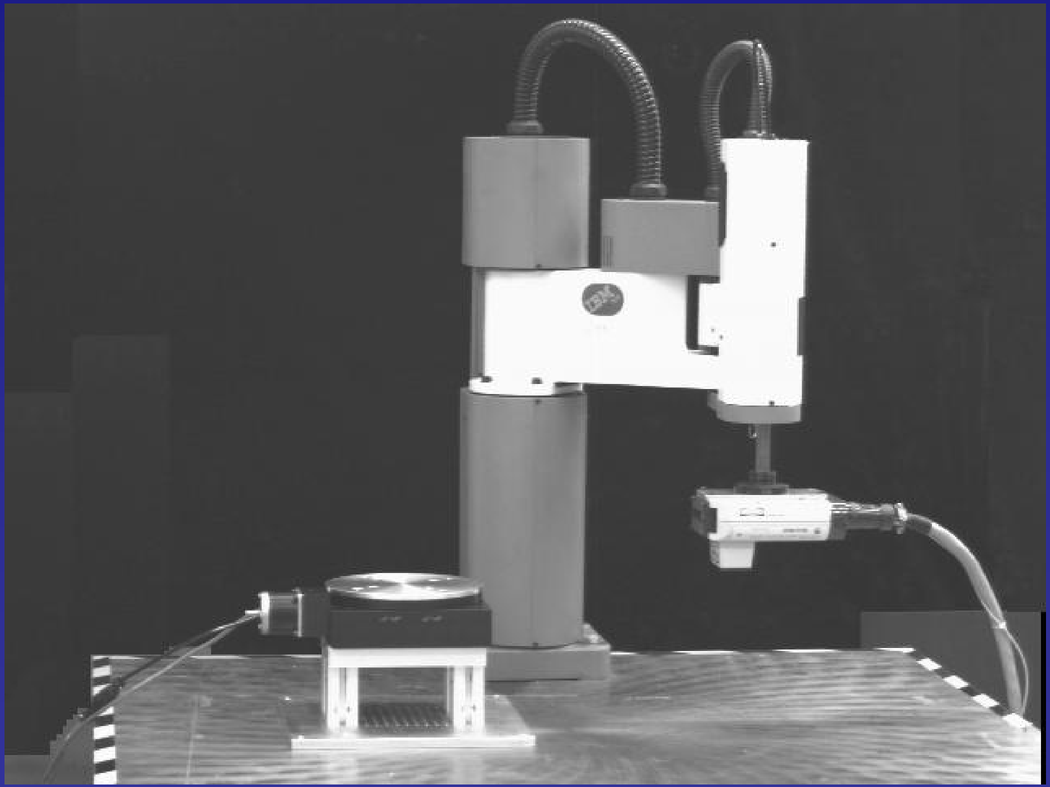
Merging Models from Different Views

- Each model is a solid, therefore we may use regularized set intersection to integrate:
  - initialize composite model to the entire workspace
  - acquire a new solid model
  - update the composite model by set intersection with new model

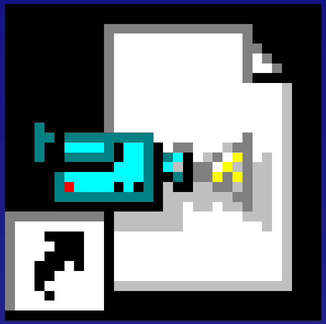
- Assumption:
  - the set described by each model must be a superset of the actual imaged object

Experimental Setup

- Sensor: Servo-Robot laser rangefinder
- Motion: IBM 7575 SCARA robot and turntable

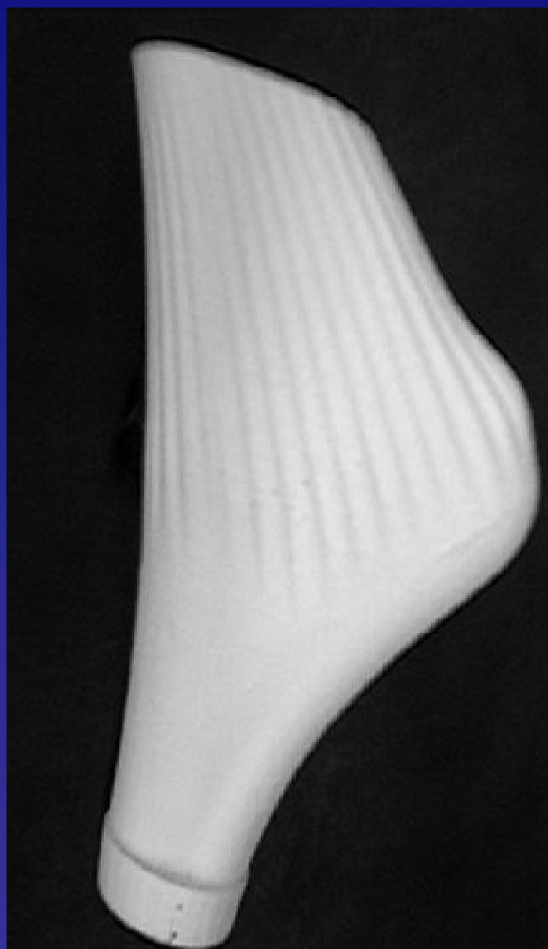


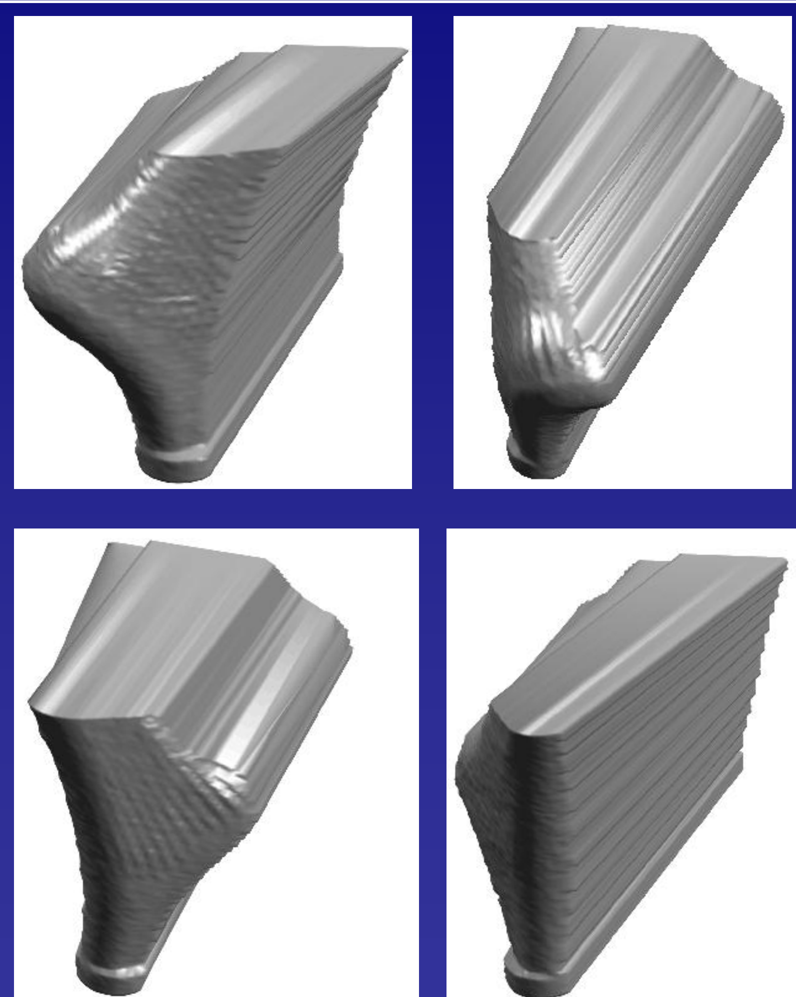
Movie!!!



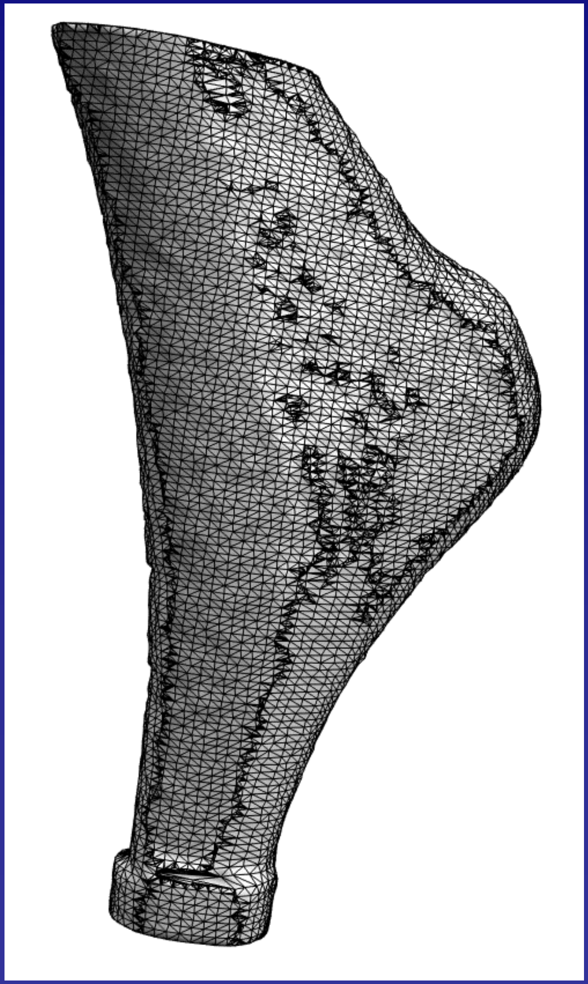
cvpr.lnk

Example: Hip Replacement

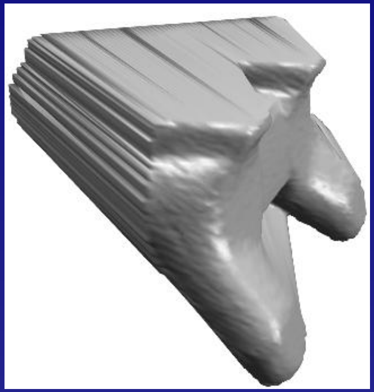


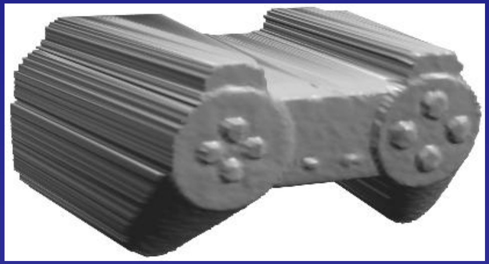


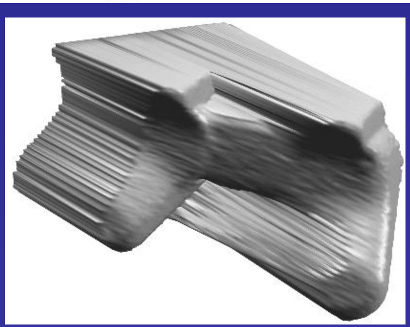
Example: Hip Replacement



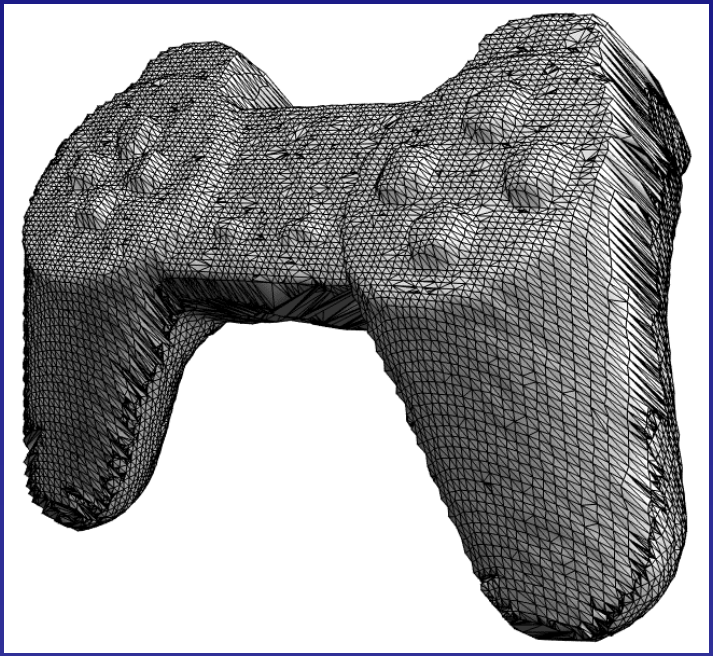
Example: Video Game Controller



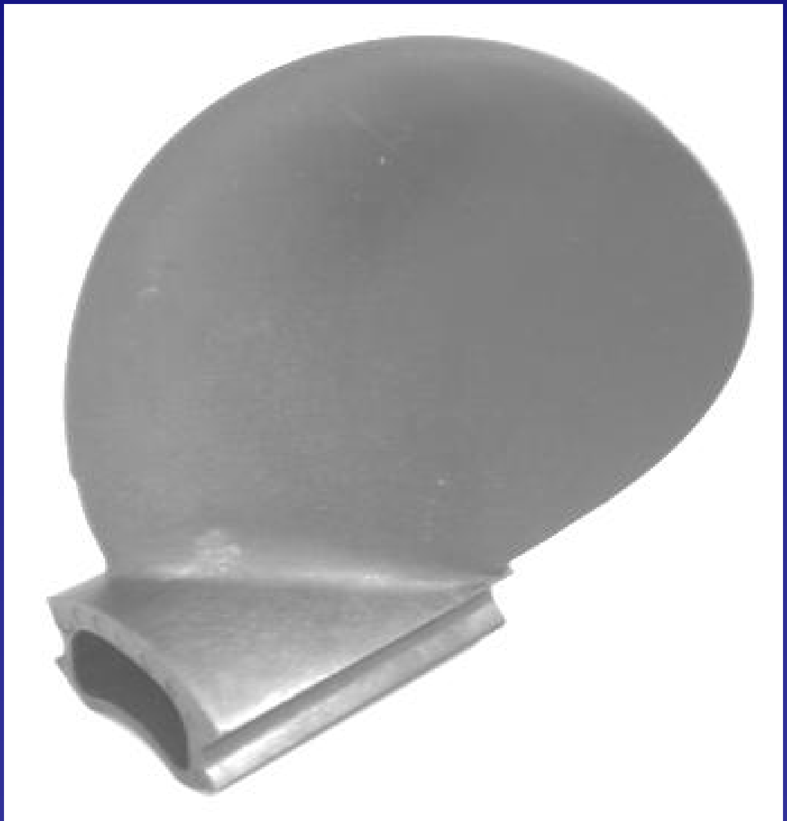


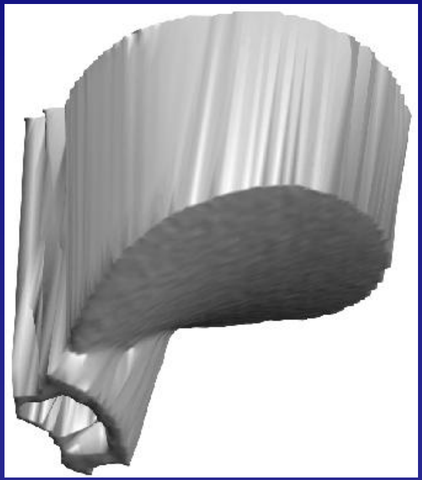


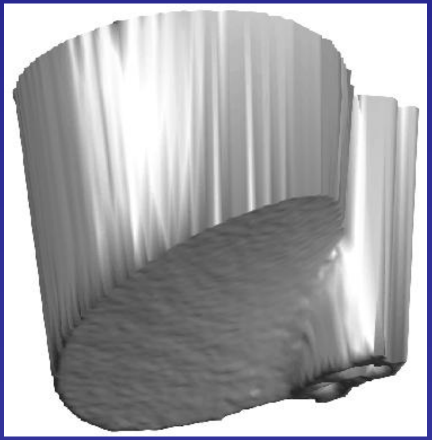
Example: Video Game Controller



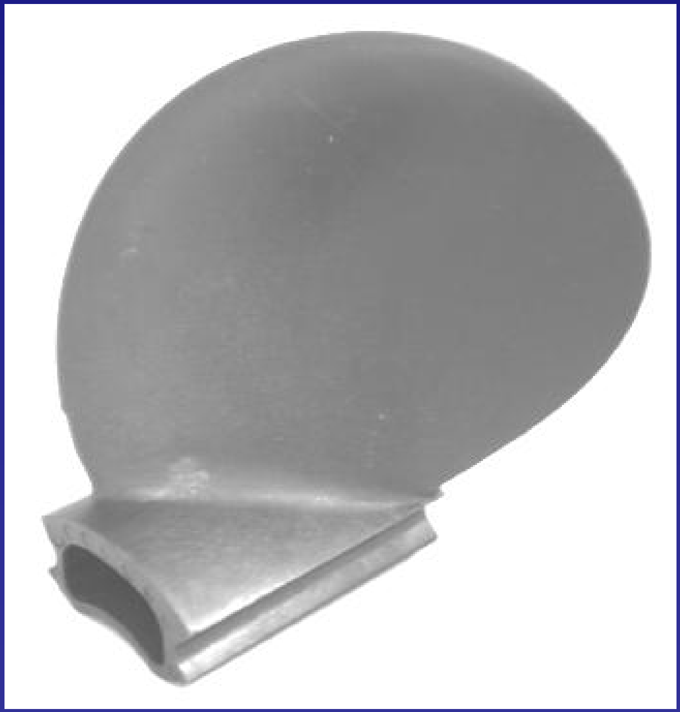
Example: Propeller Blade

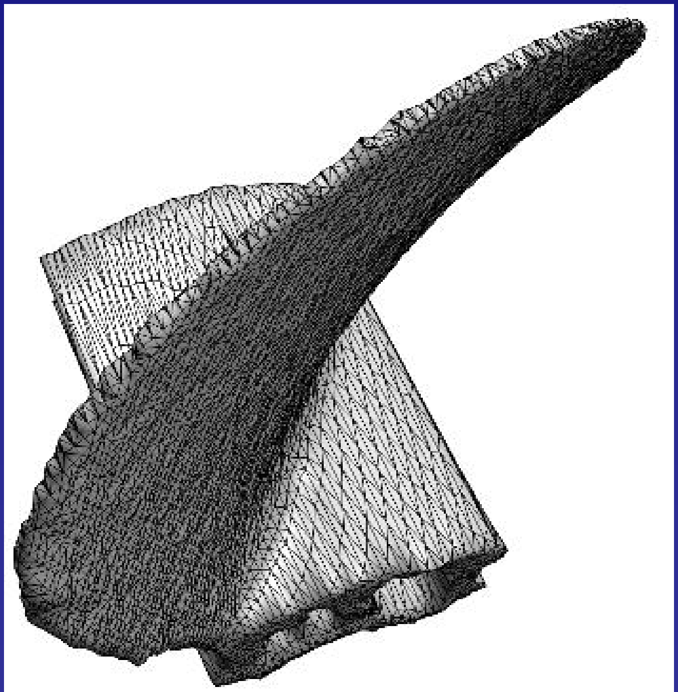




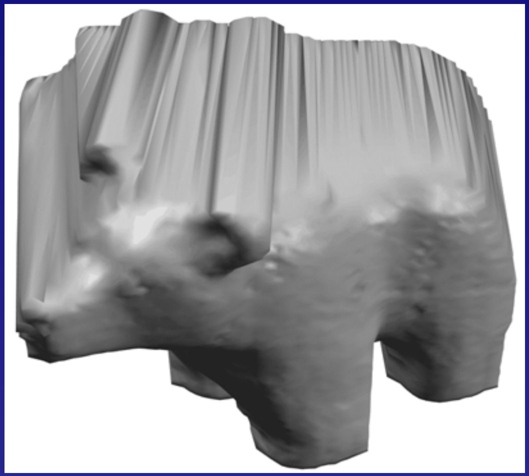
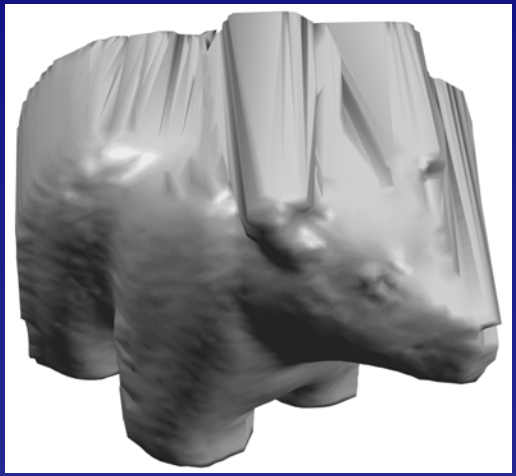
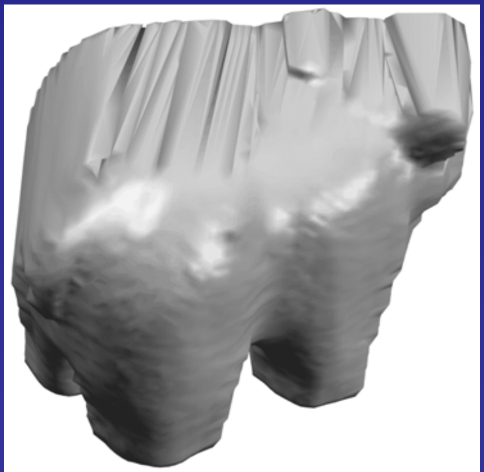
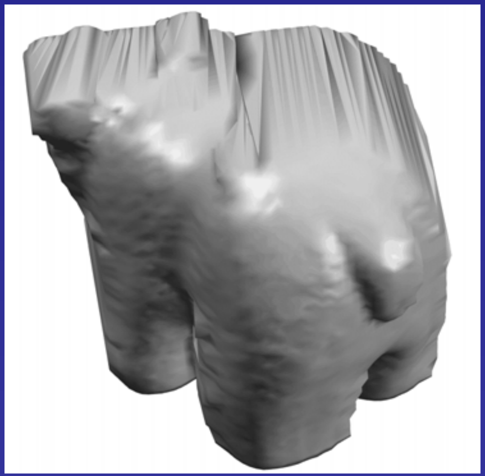


Example: Propeller Blade

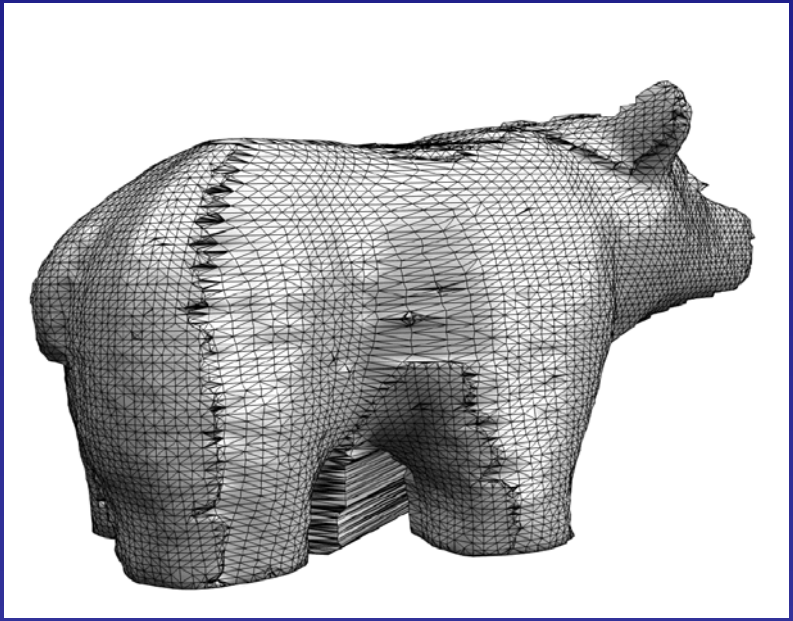




Example: Toy Bear in 4 Views

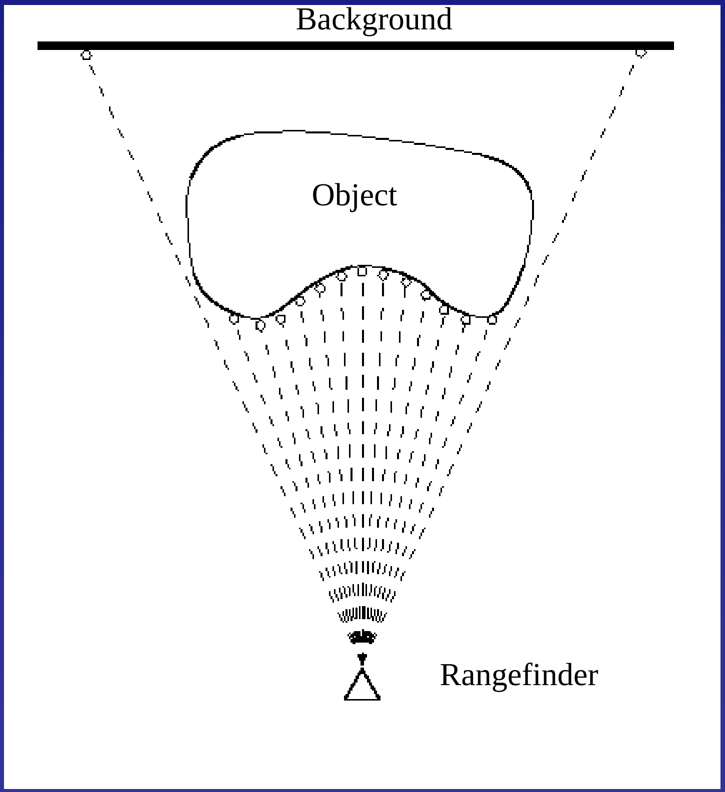





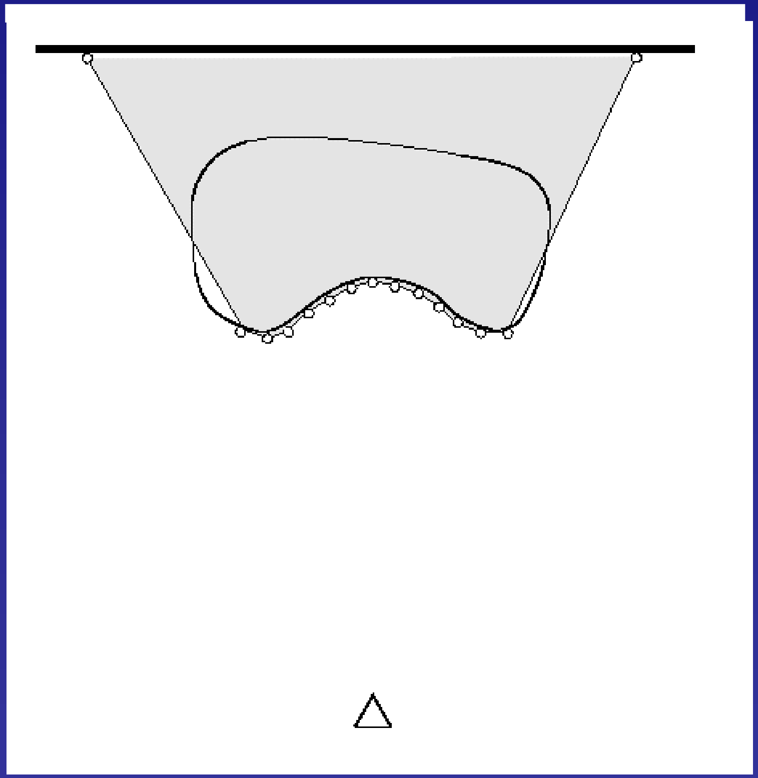
Example: Toy Bear in 4 Views



Effects of Sampling on Integration

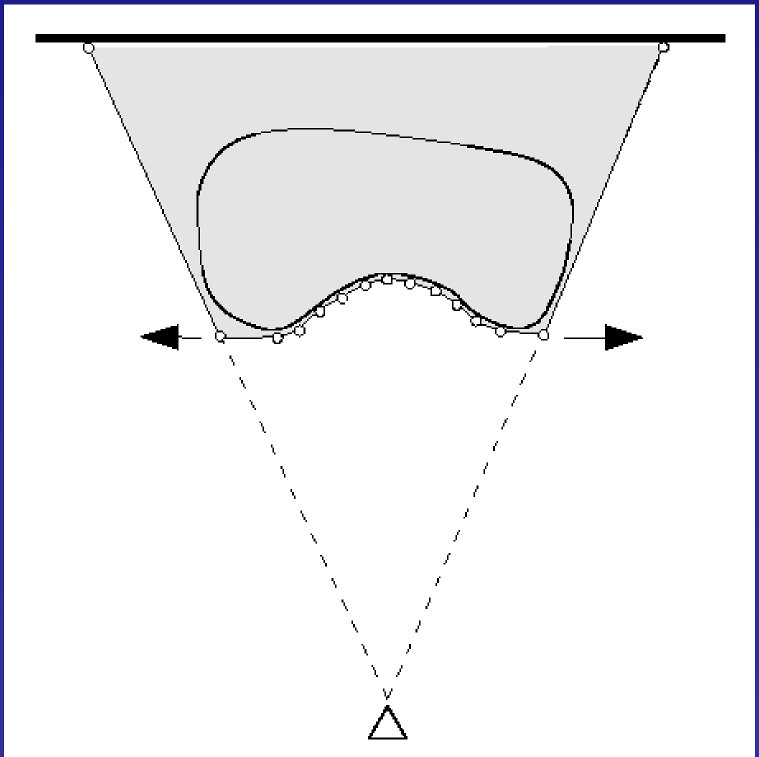
- Violation of sampling theorem: mesh elements do not necessarily correspond to object surfaces!

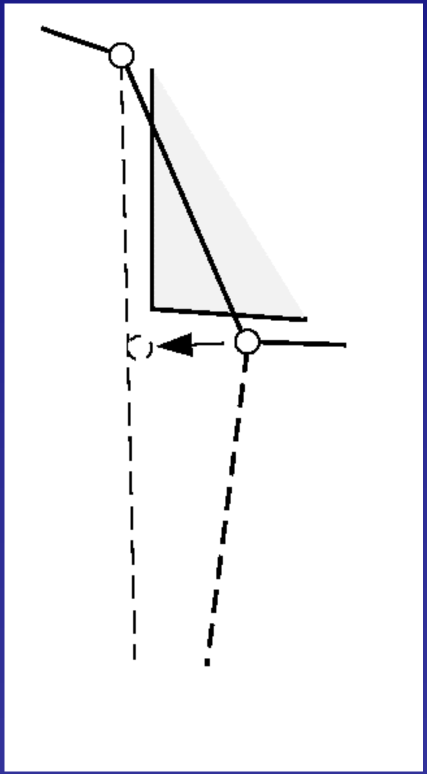




Effects of Sampling on Integration

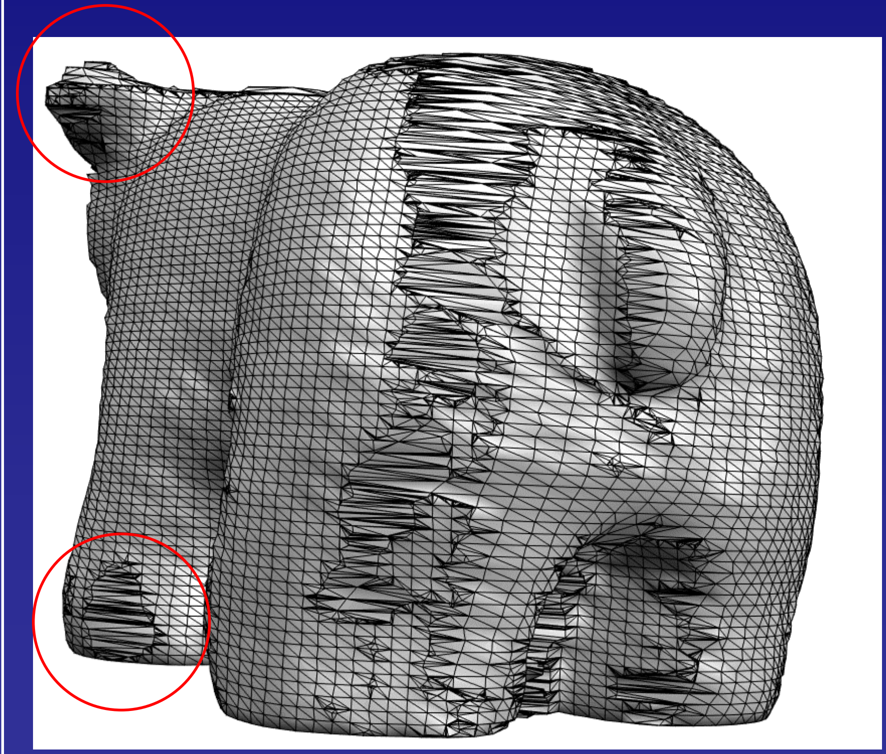
- Surfaces requiring dilation may be identified by their surface type attribute



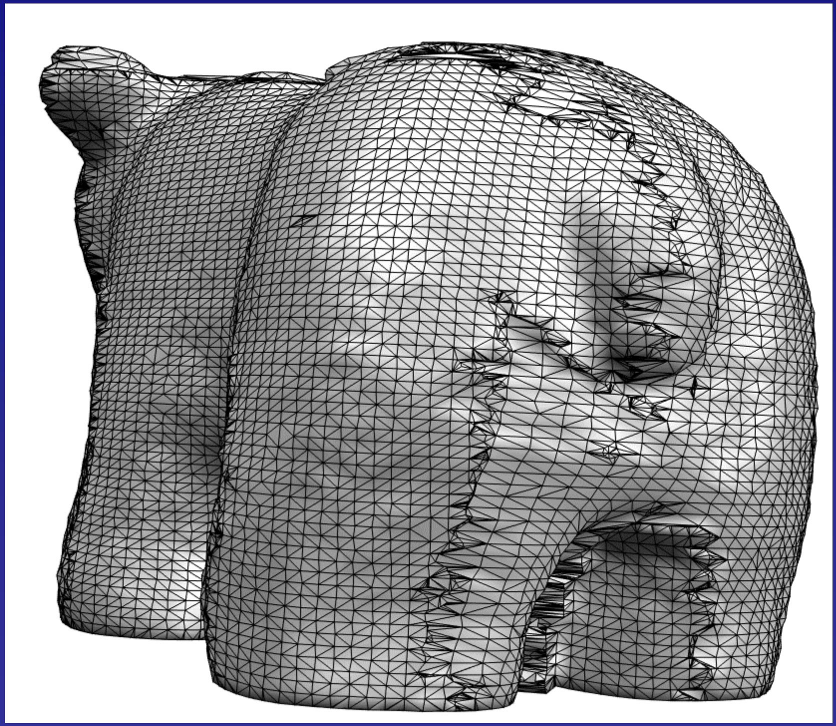


Effects of Sampling on Integration

- Results:



without dilation



with dilation

Integration of View Planning

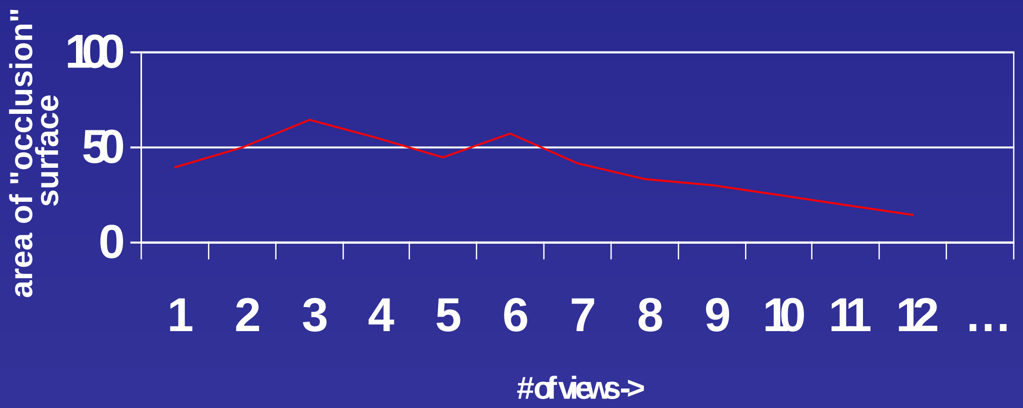
- Why plan sensor viewpoints?
  - To ensure adequate model acquisition
  - To reduce the number of sensing operations
- How does one determine the next view?
  - Must represent what has not been sensed
  - Must determine a viewpoint that will improve model

Previous Work:

- Planning in unknown environments
  - [Connolly 1989]
  - [Maver & Bajcsy 1990]
  - [Whaite & Ferrie 1992]
  - [Kutulakos 1994]
  - [Pito 1997]
- Planning in known environments
  - [Sedas-Gersey 1993]
  - [Abrams 1997]
  - [Tarabanis et al. 1995]

Strategy in Viewpoint Planning

- At what point should planning be done?
  - at the outset?
  - after some intial model?
  - never?



Integration of View Planning

- Caveats for methods in unknown environs:
  - scene self-occlusion not considered
  - use brute -force raycasting
  - discrete solutions only

- Our approach:
  - acquire preliminary model first, <u>then</u> plan
  - plan to acquire “occlusion” surfaces (targets)
  - apply static sensor planning methods
  - solutions (plans) are in continuous space

Sensor Viewpoint Planning

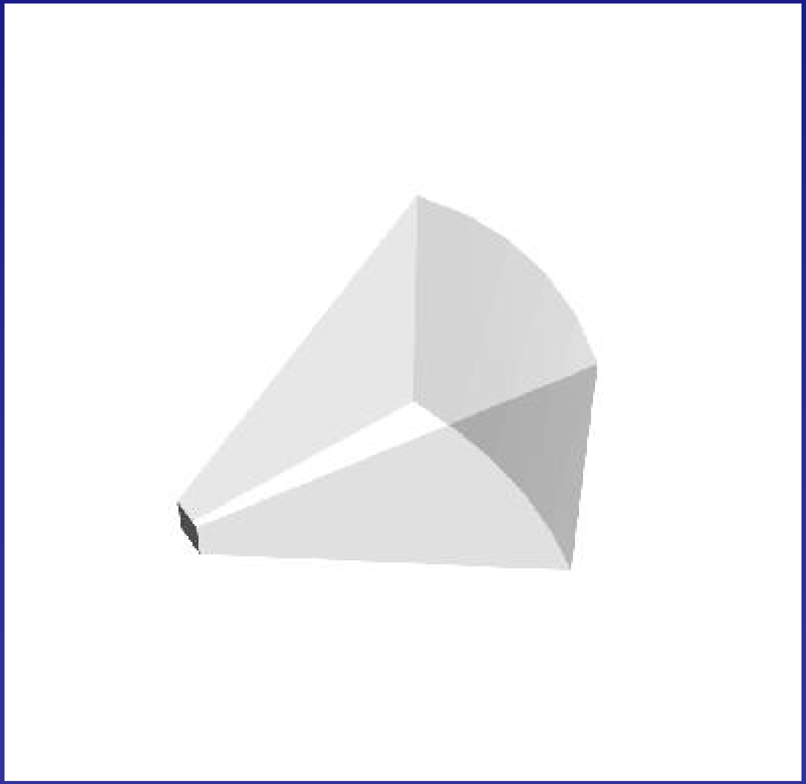
- Our approach:
  - target driven - computes visibility for target using the composite model
  - operates in continuous space - solutions are 3-D volumes
  - targets are selected from model surfaces labeled “occlusion”

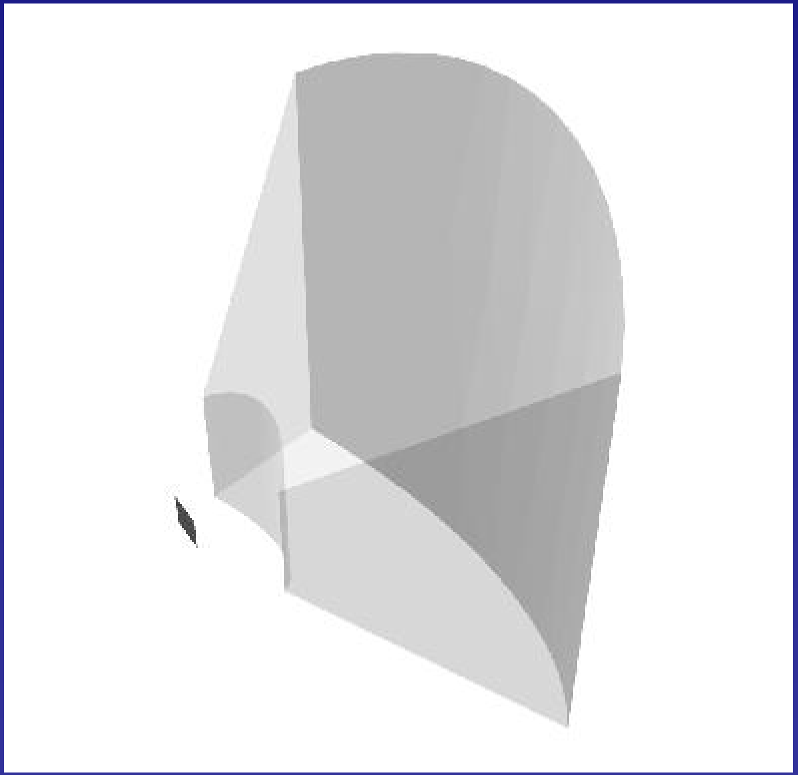
Viewpoint Planning: Method

- Determine constraints on sensor position
  - sensor imaging constraints
  - model occlusion constraints
  - sensor placement constraints
- Integrate constraints to form plan
- Discretize as final step (if necessary)

Sensor Imaging Constraints

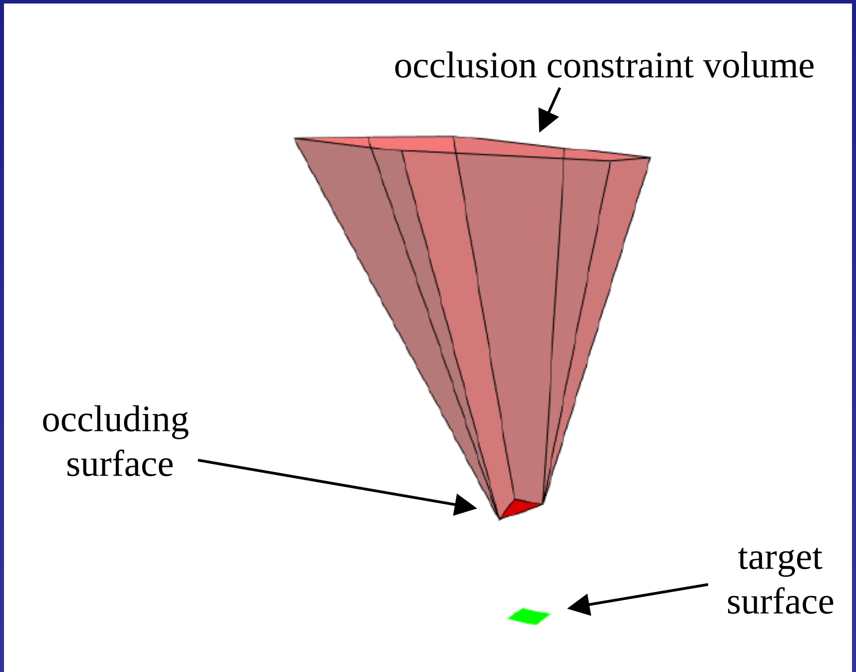
- Describe limits of sensor's ability to acquire data
- rangefinder example: "breakdown angle" α, standoff:





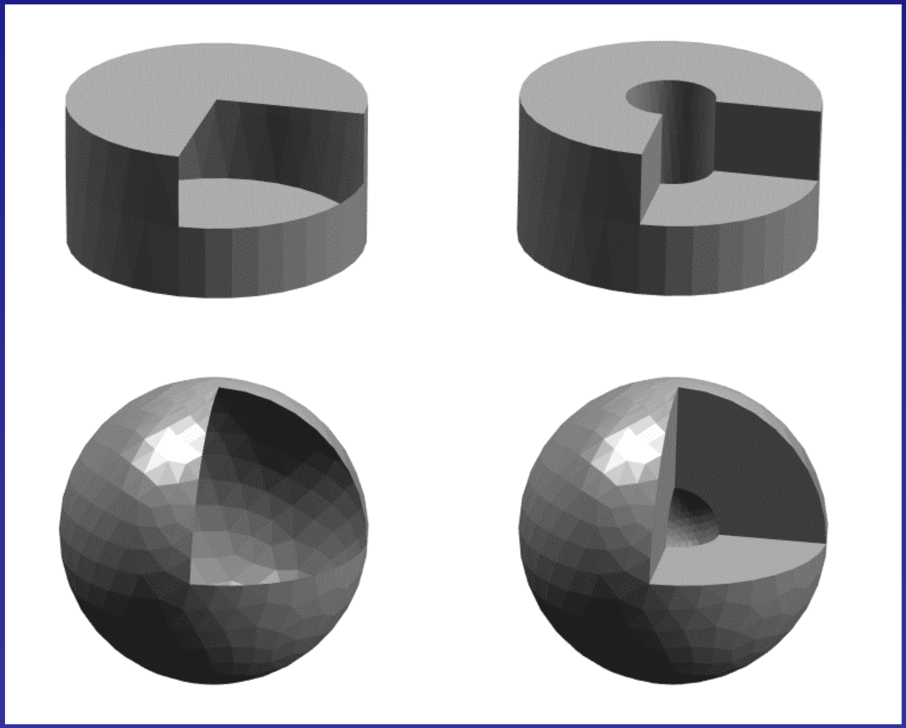
Model Occlusion Constraints

- Describe space occluded from target
- Must be calculated for *all* occluding model surfaces!



Sensor Placement Constraints

- Describe workspace positions available to sensor
- Usually a function of manipulator type



Constraint Integration

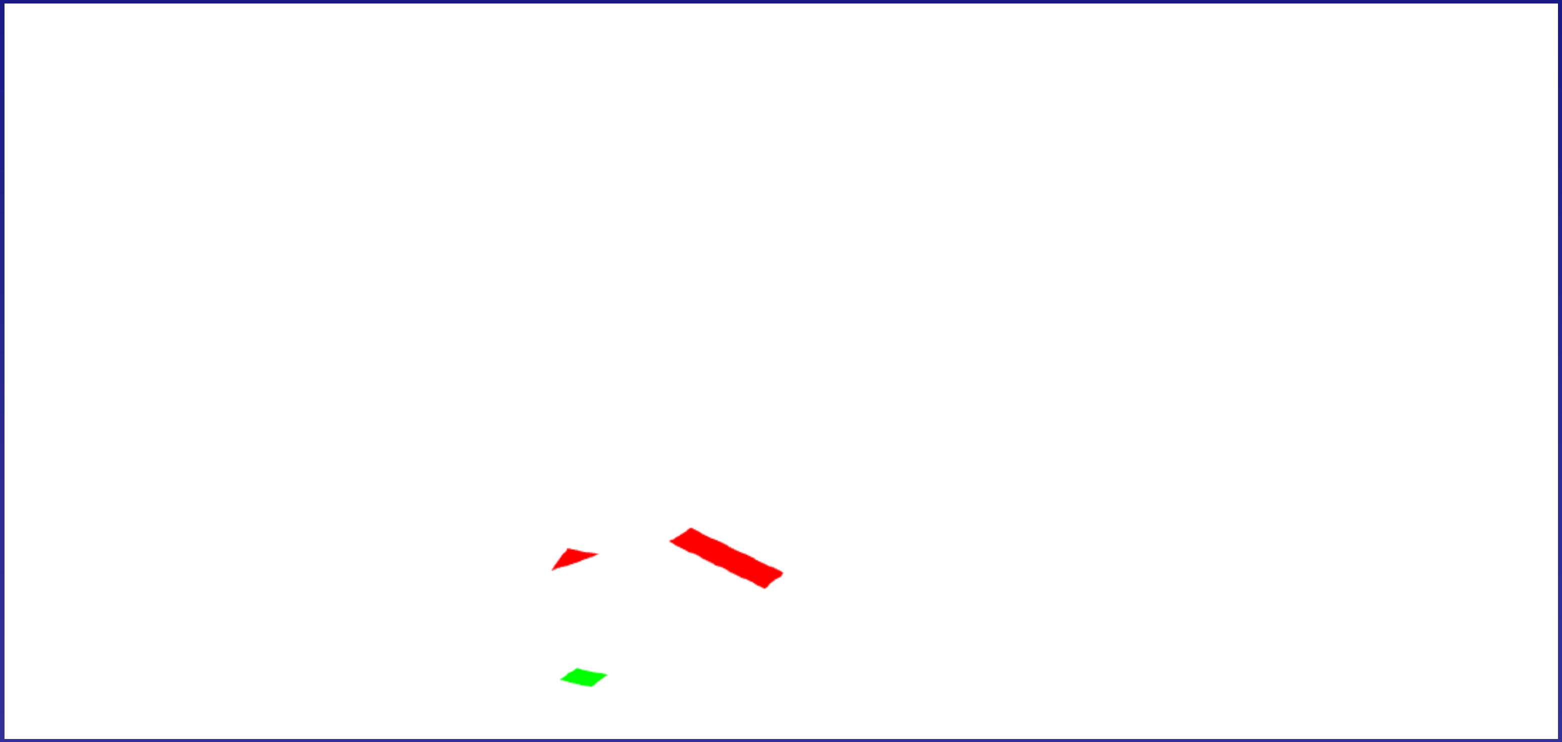
- Constraints are represented using solid modeling primitives
- Plan is calculated by:

(imaging constraints - occlusion constraints)

$\cap$ placement constraints

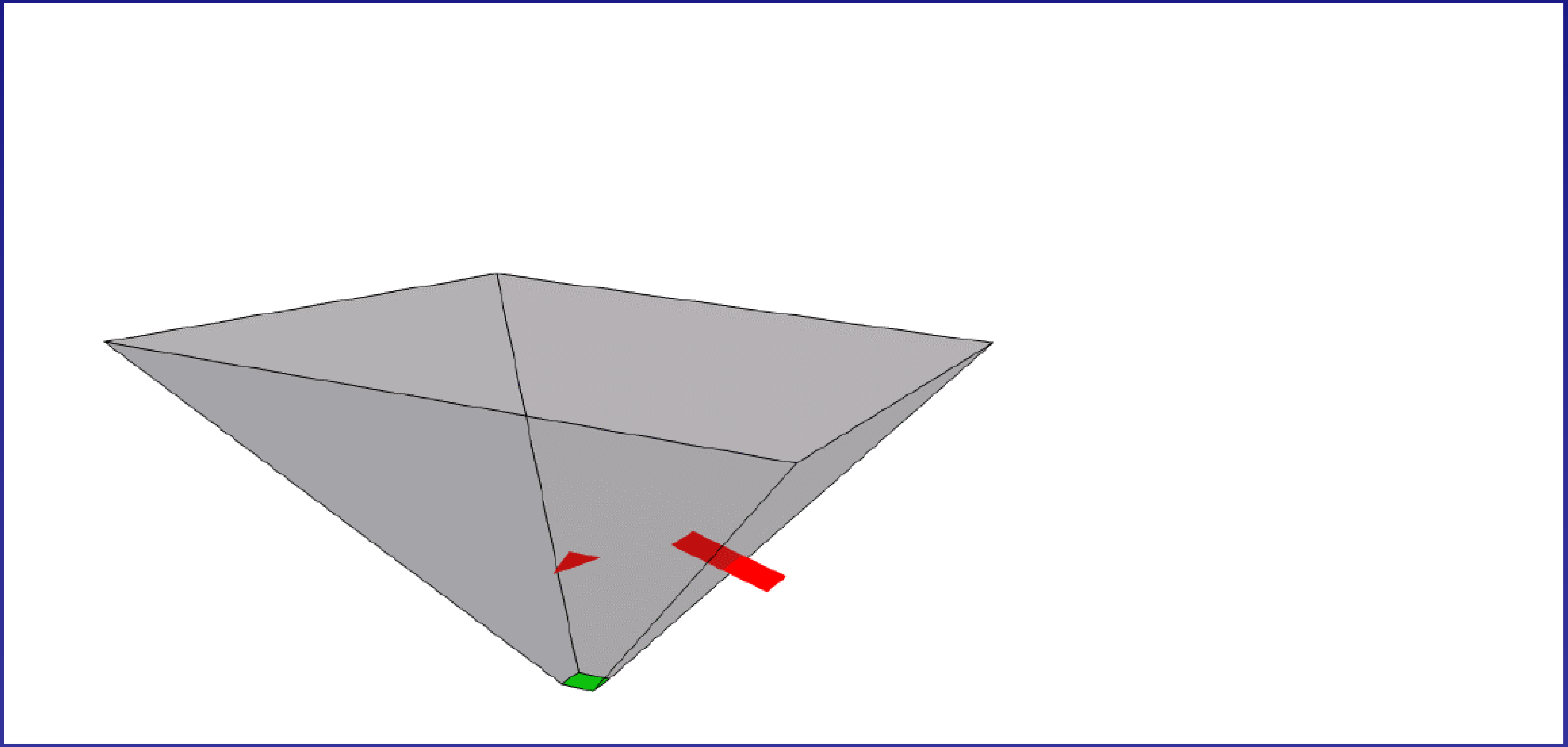
Planning Example

- Target is green, other model surfaces in red:



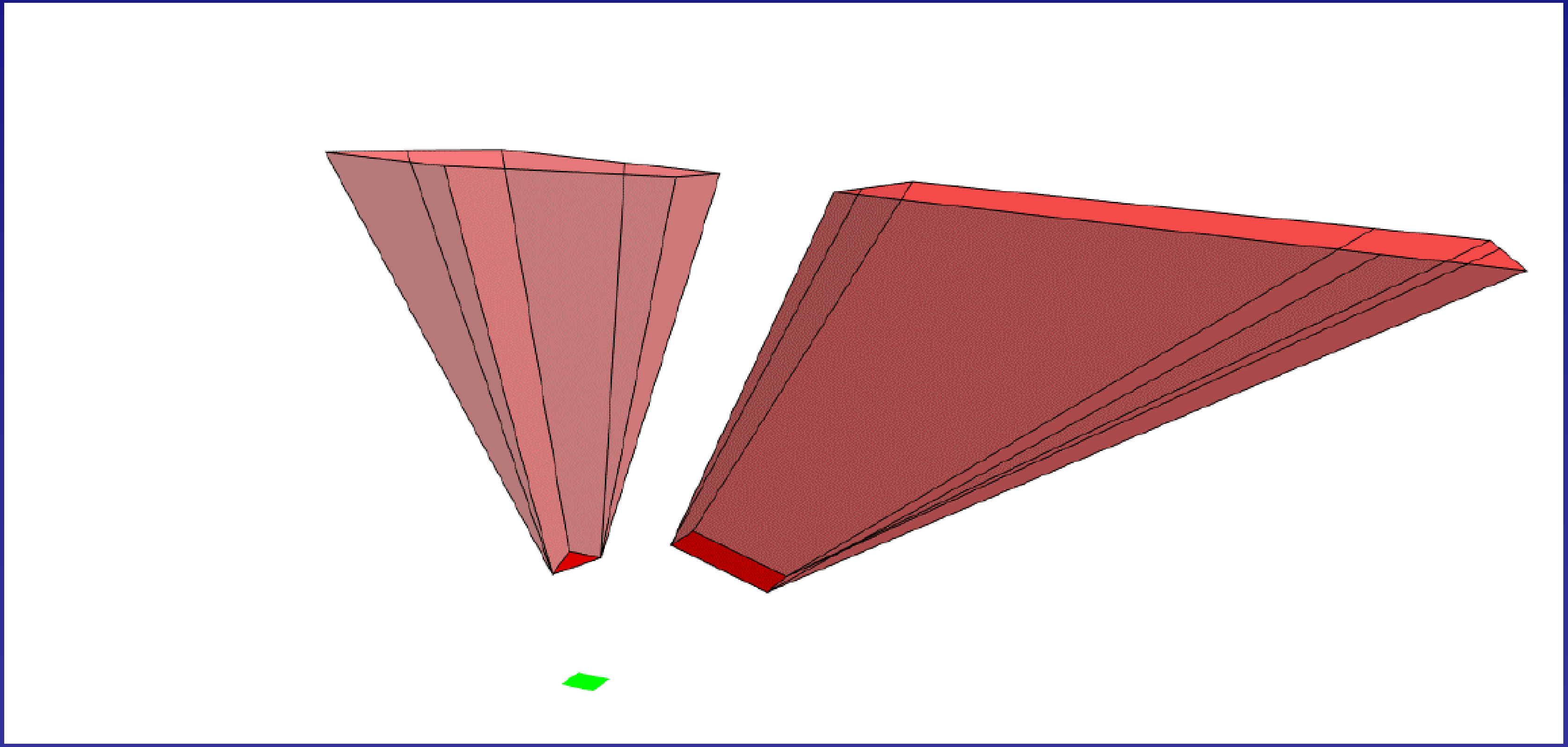
Planning Example

- Compute sensor imaging constraint:



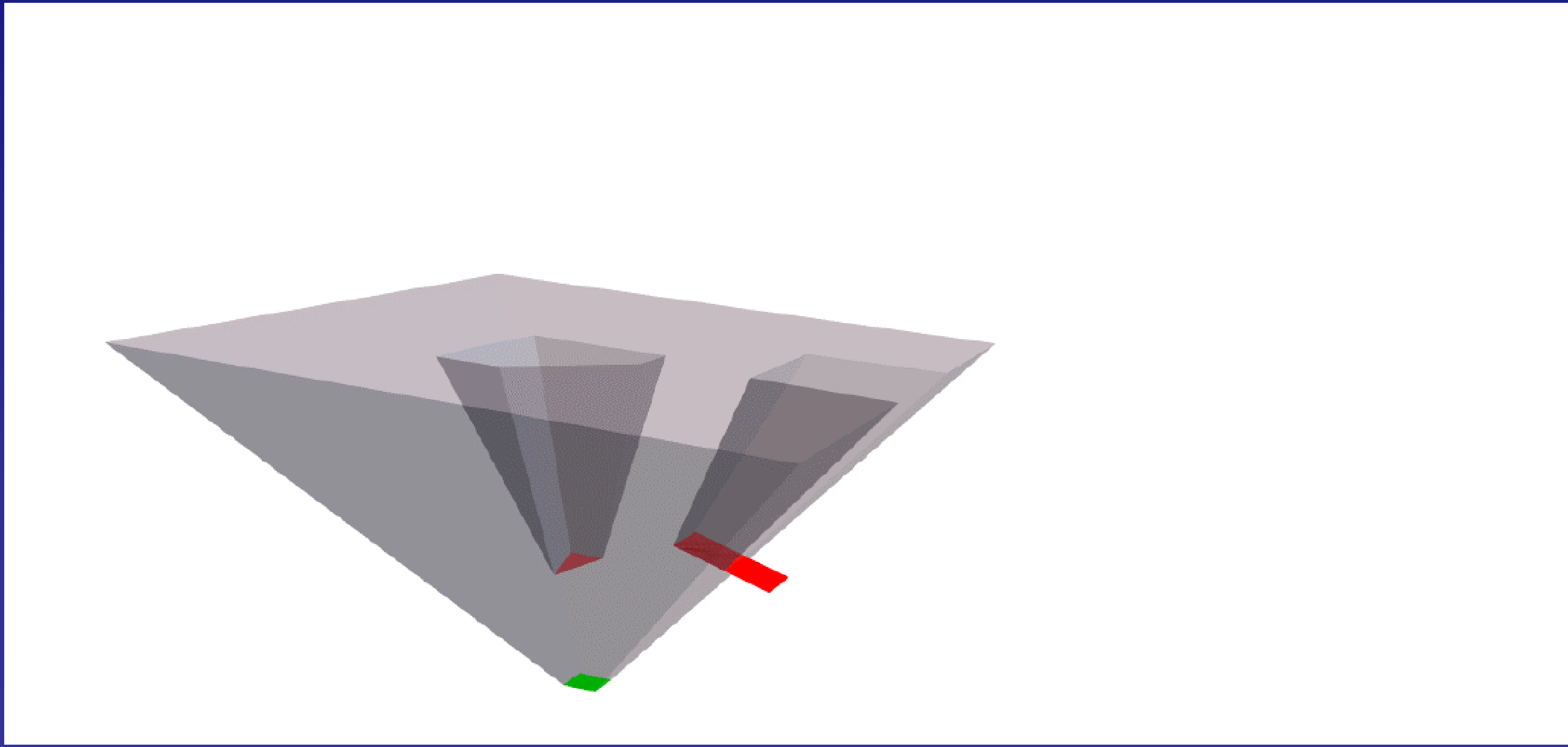
Planning Example

- Compute model occlusion constraints:



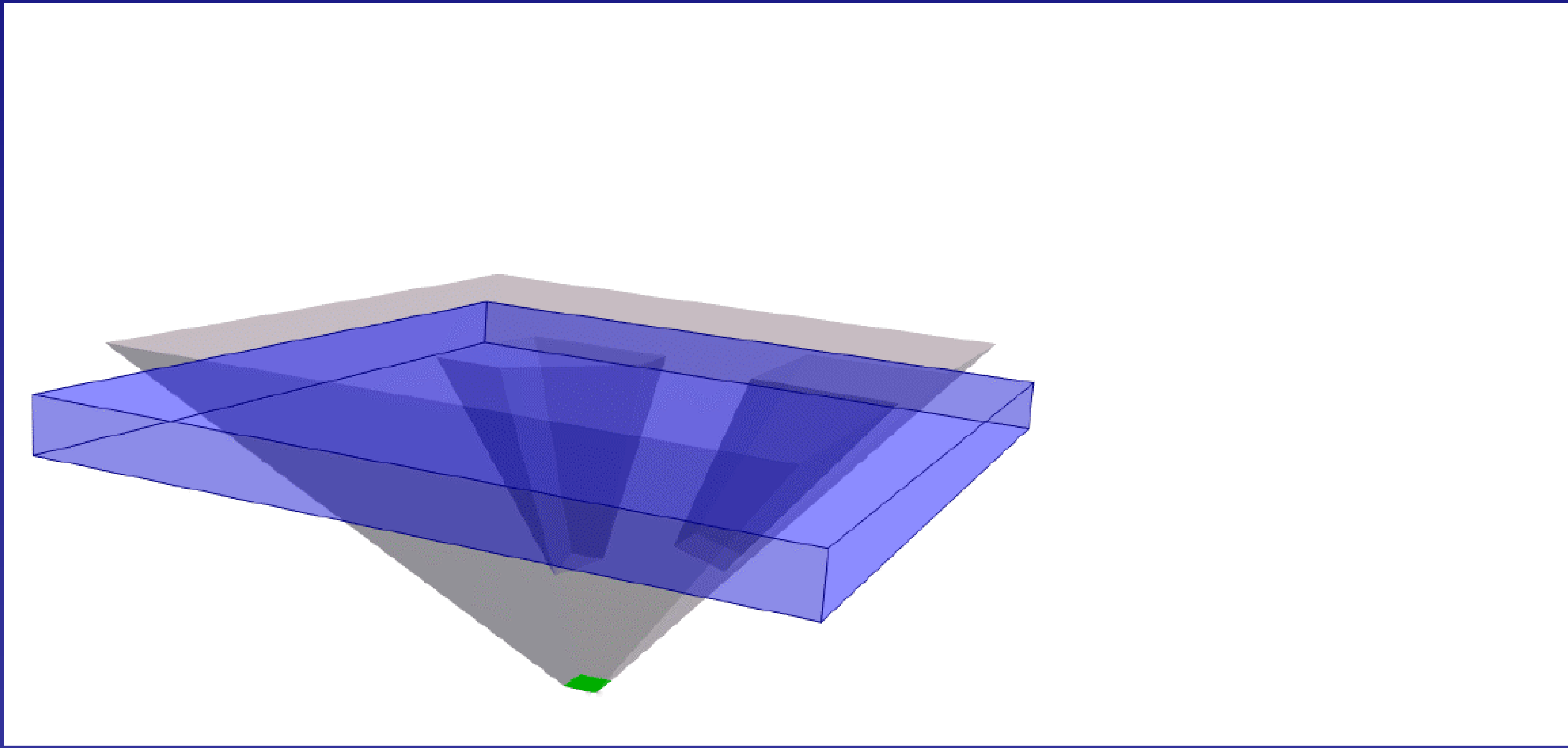
Planning Example

- Compute visibility volume



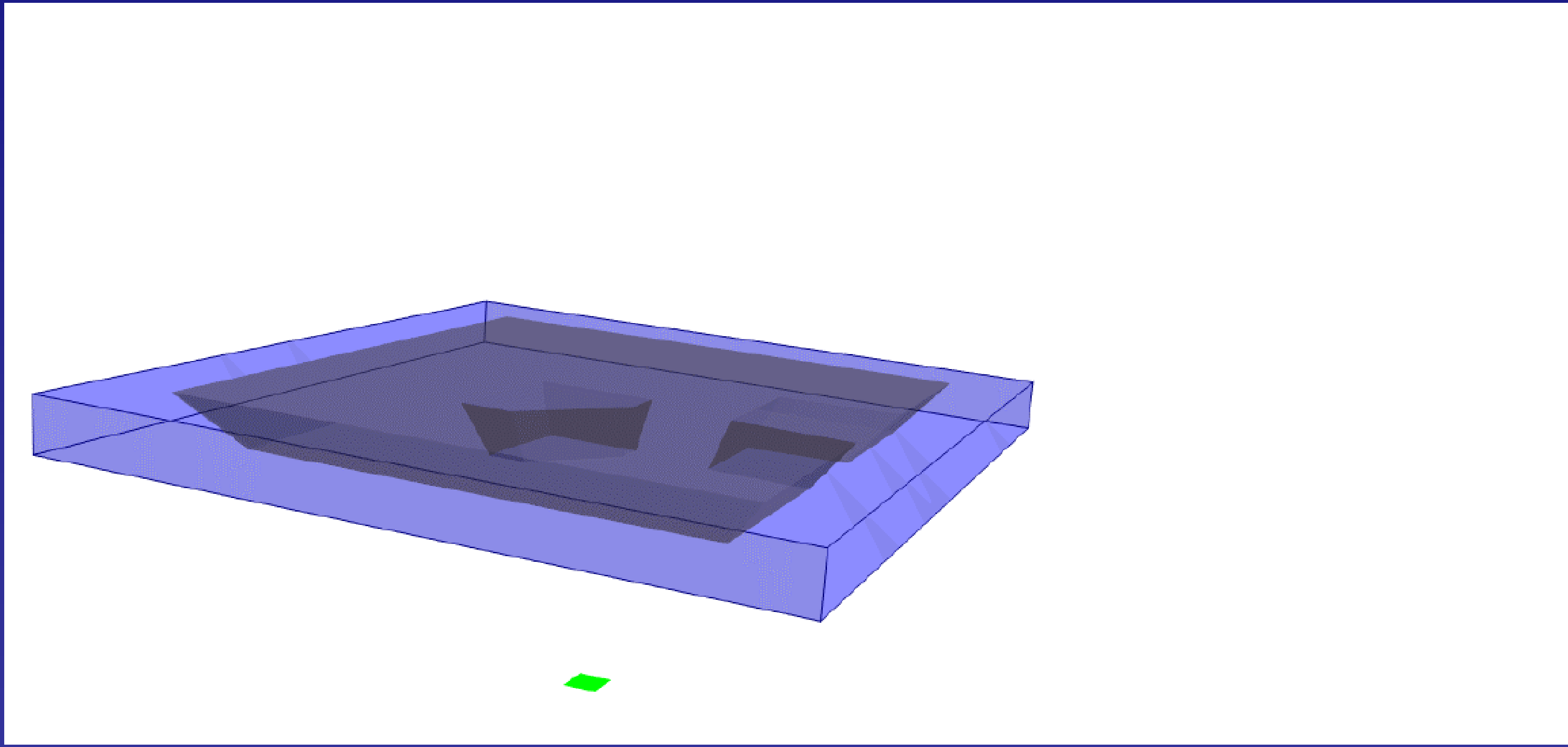
Planning Example

- Compute sensor placement constraint:



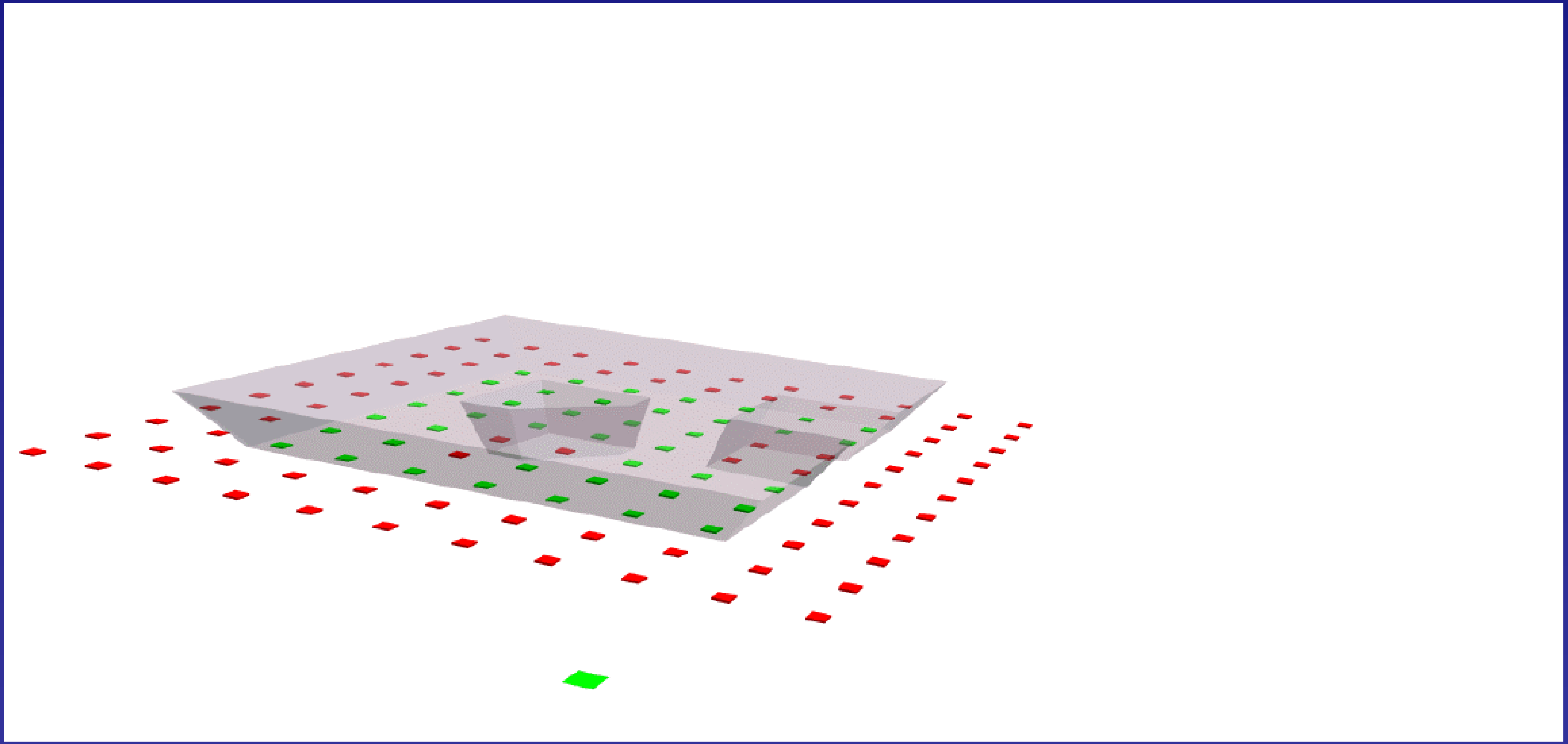
Planning Example

- Intersect placement constraint with visibility volume:

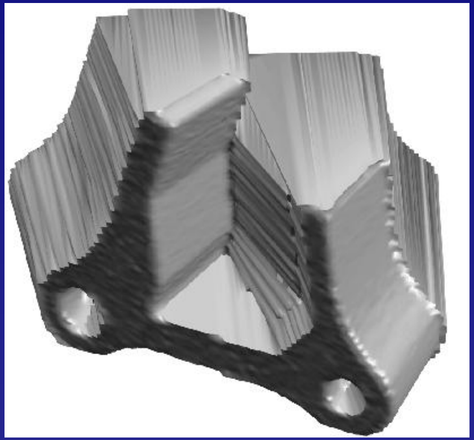


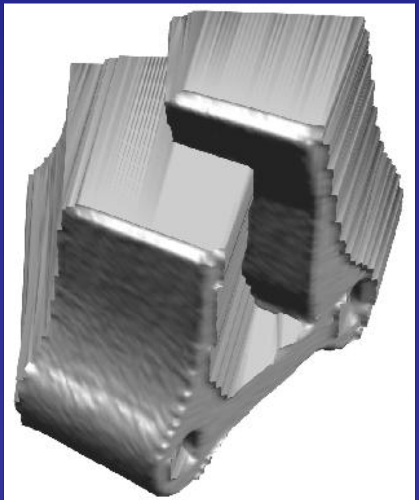
Planning Example

- Discretize sensor space and test for inclusion:

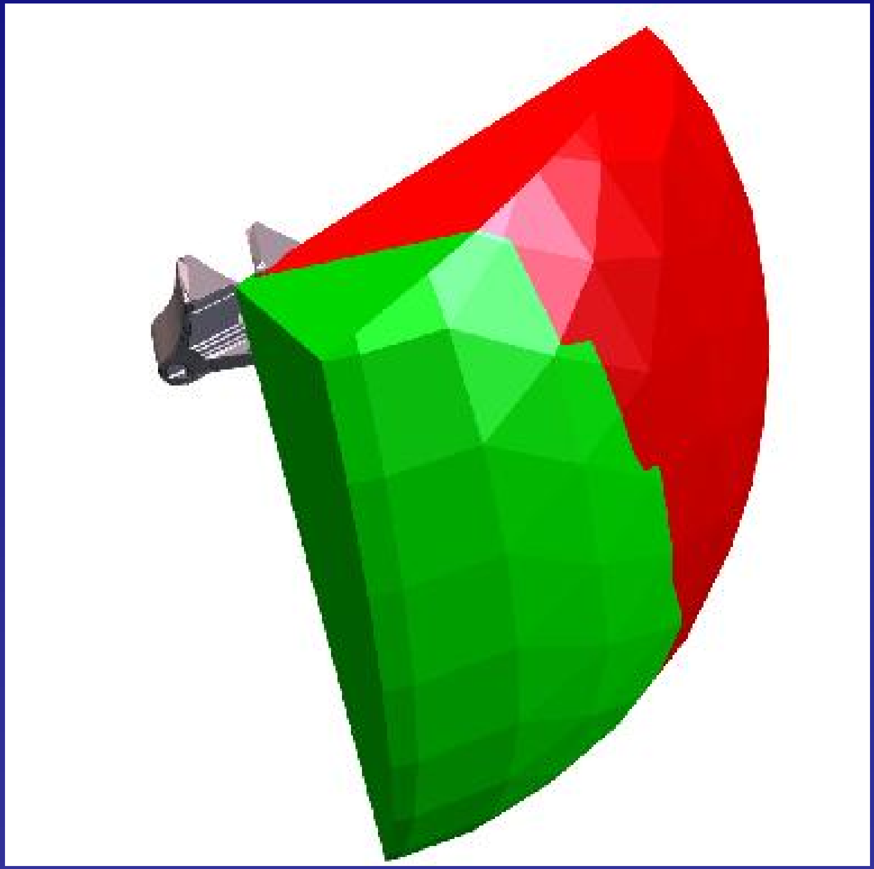


Example of View Planning: Strut
Target: Occluded Surface of Max. Area

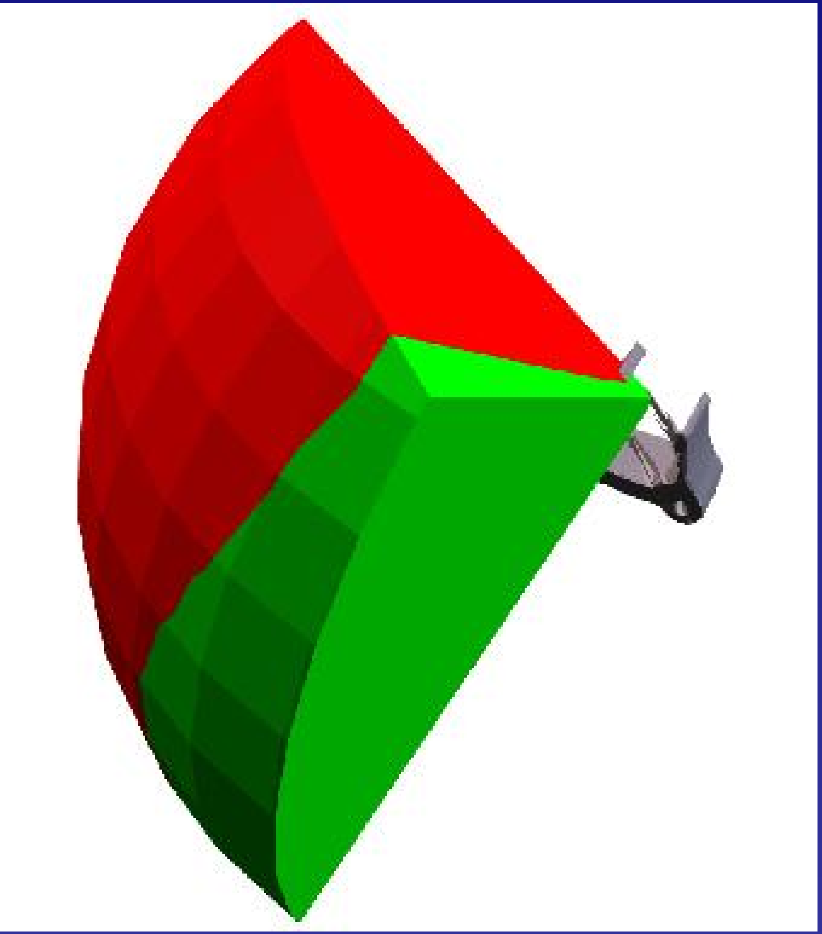




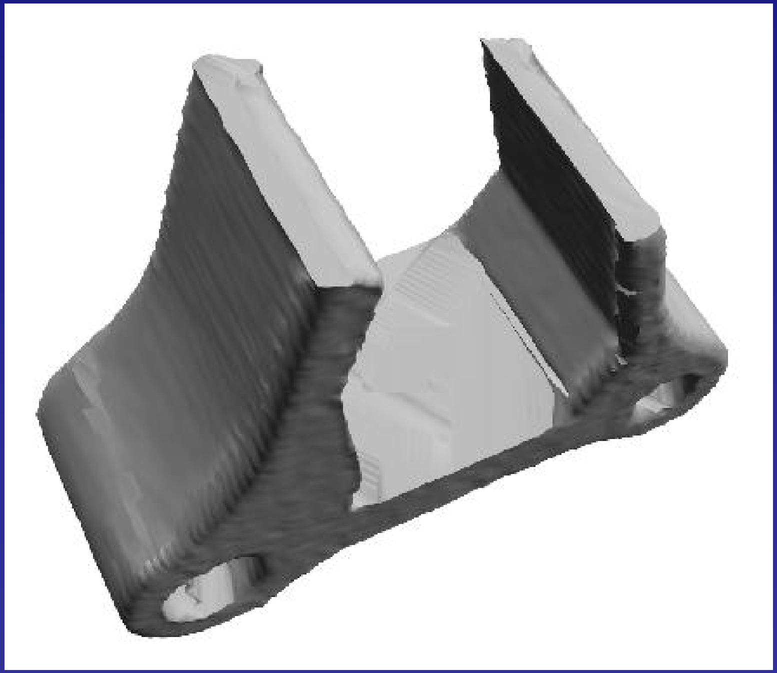
Example of View Planning: Strut



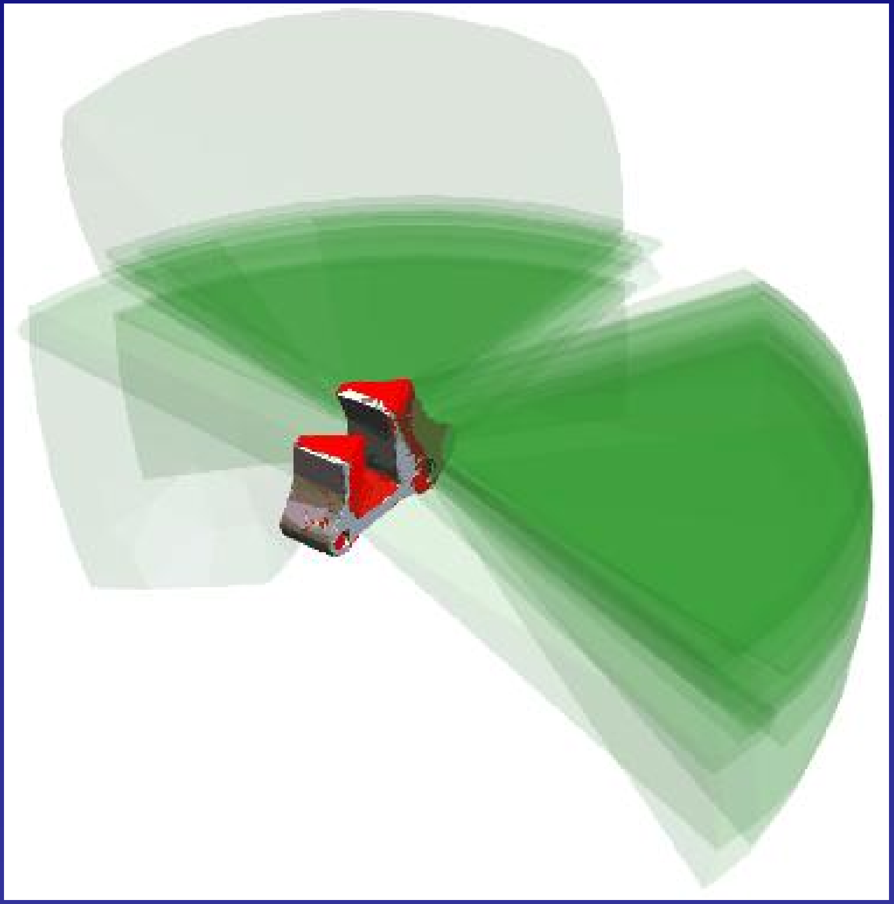
Example of View Planning: Strut



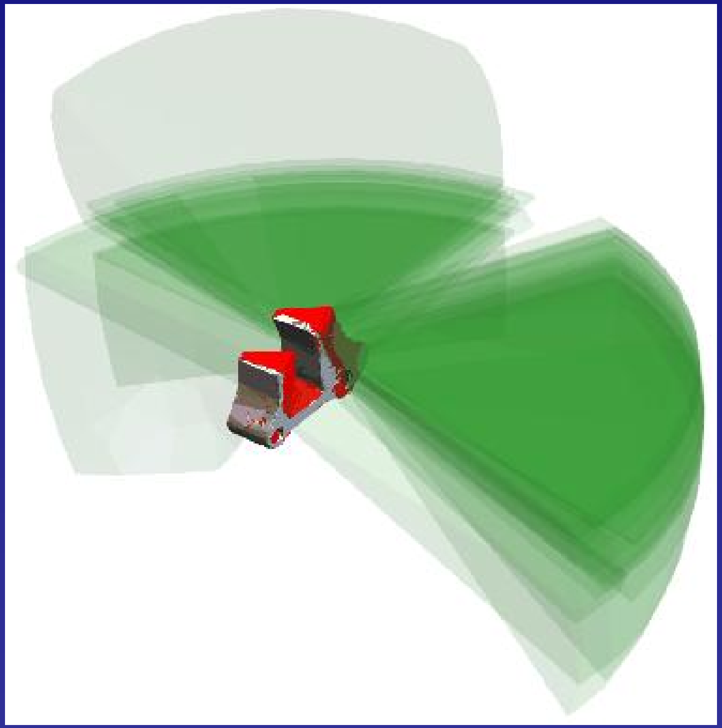
Strut: Final Model

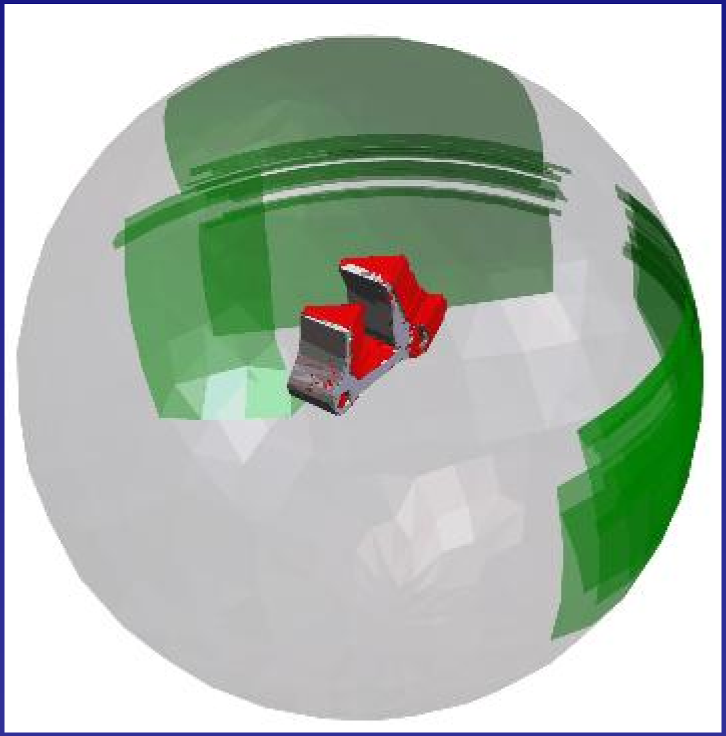


Considering Multiple Targets



Including Sensor Placement Constraints

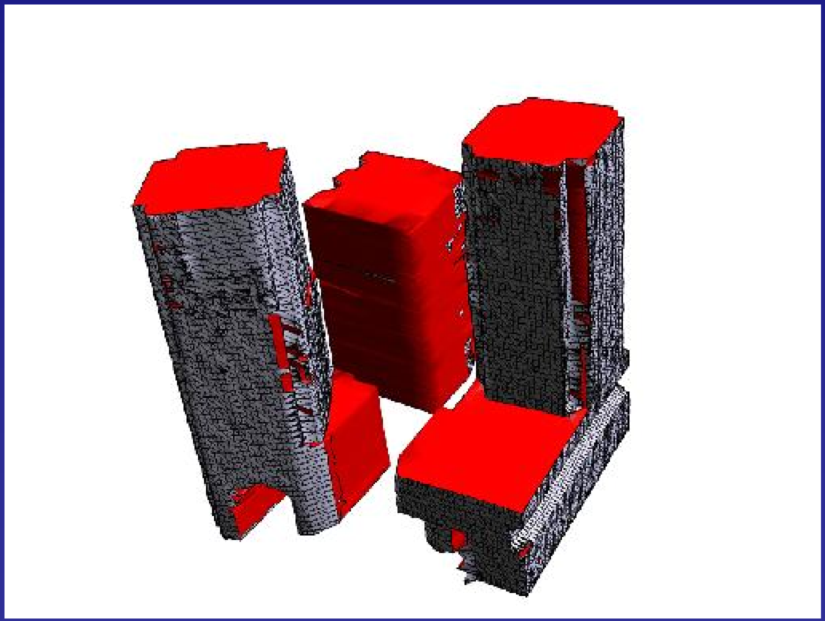


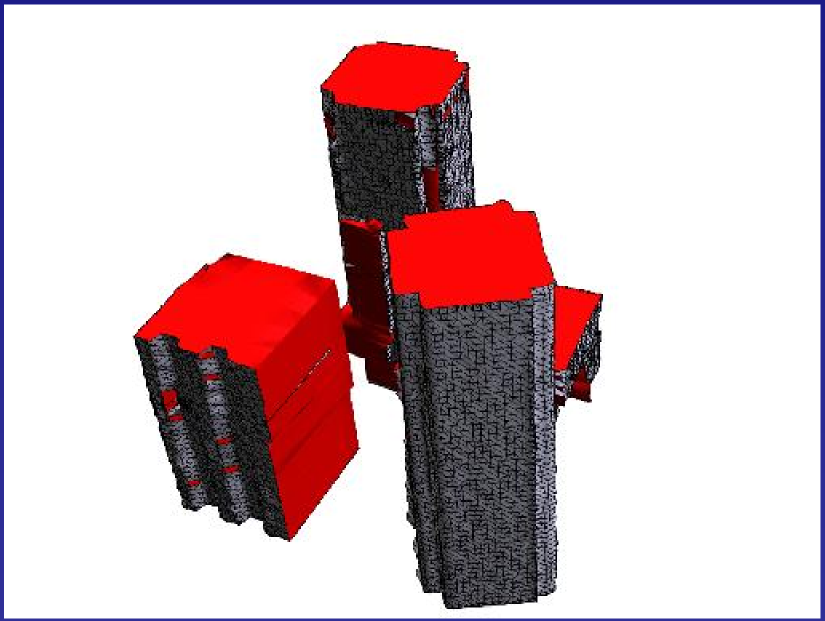


Example: Model City

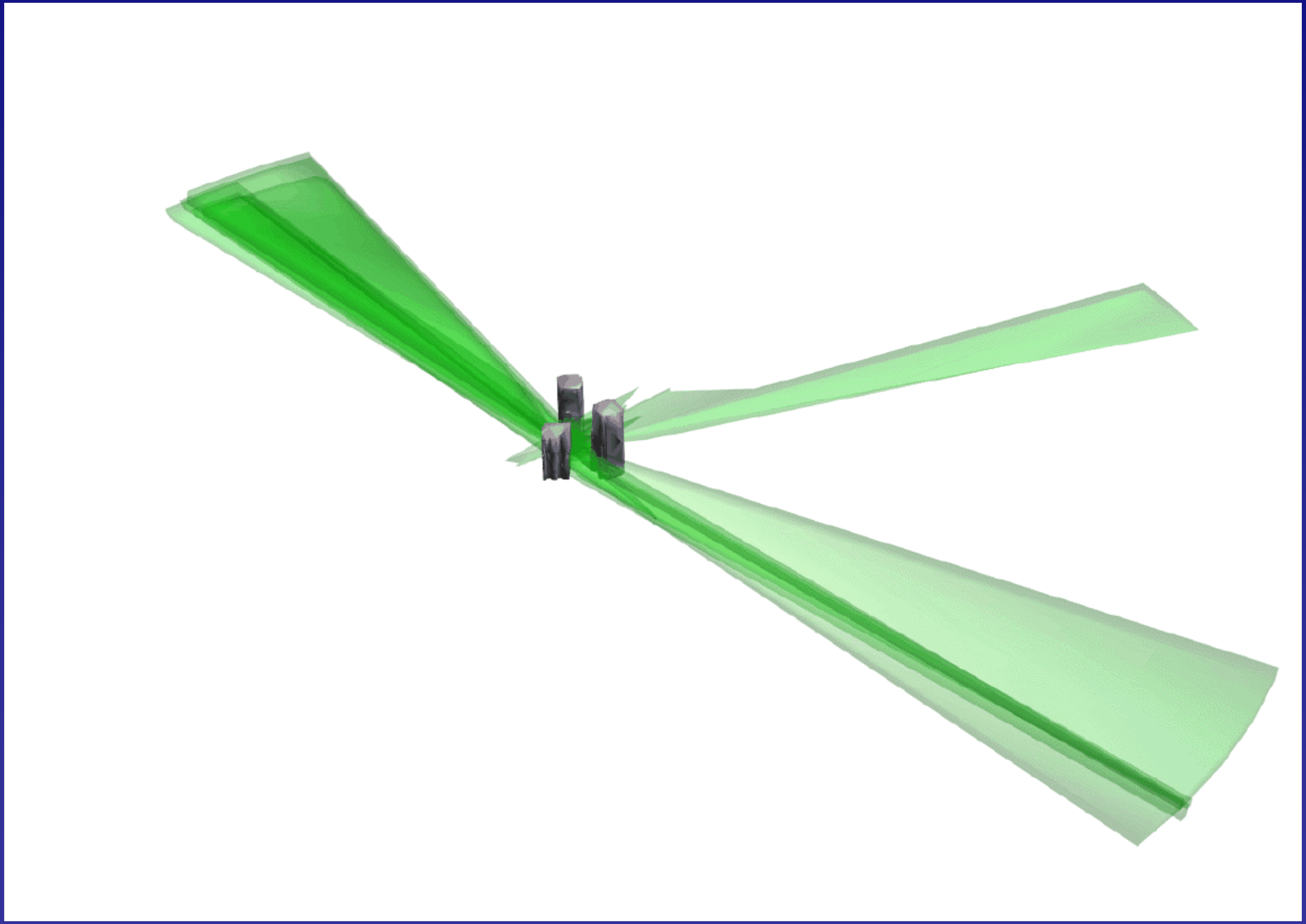
- 3 buildings with very high occlusion:

City Model: After 4 Views

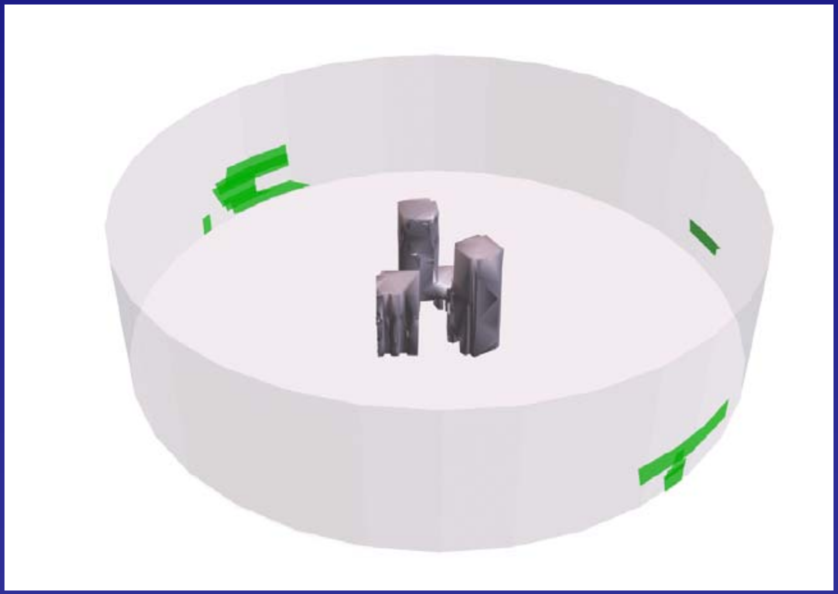


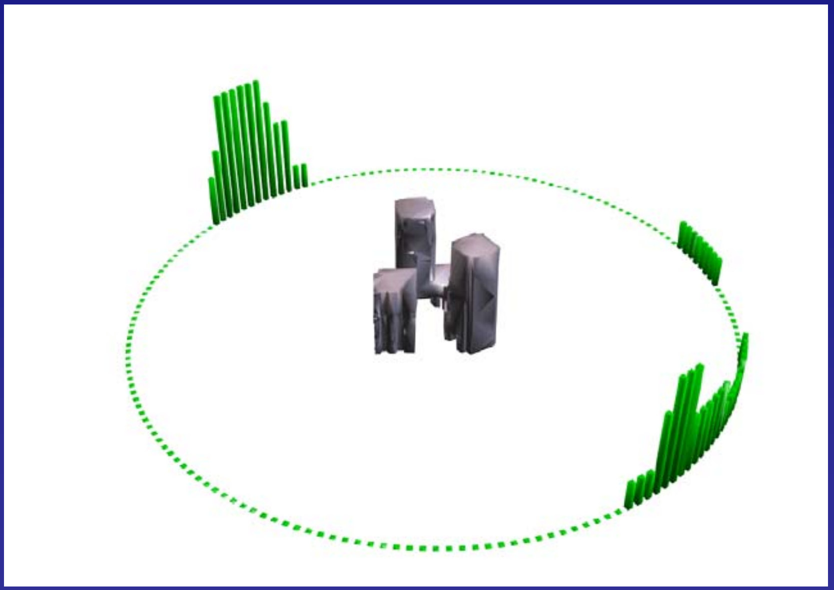


City Model: Target Visibility for 30 Largest Occluded Surfaces by Area

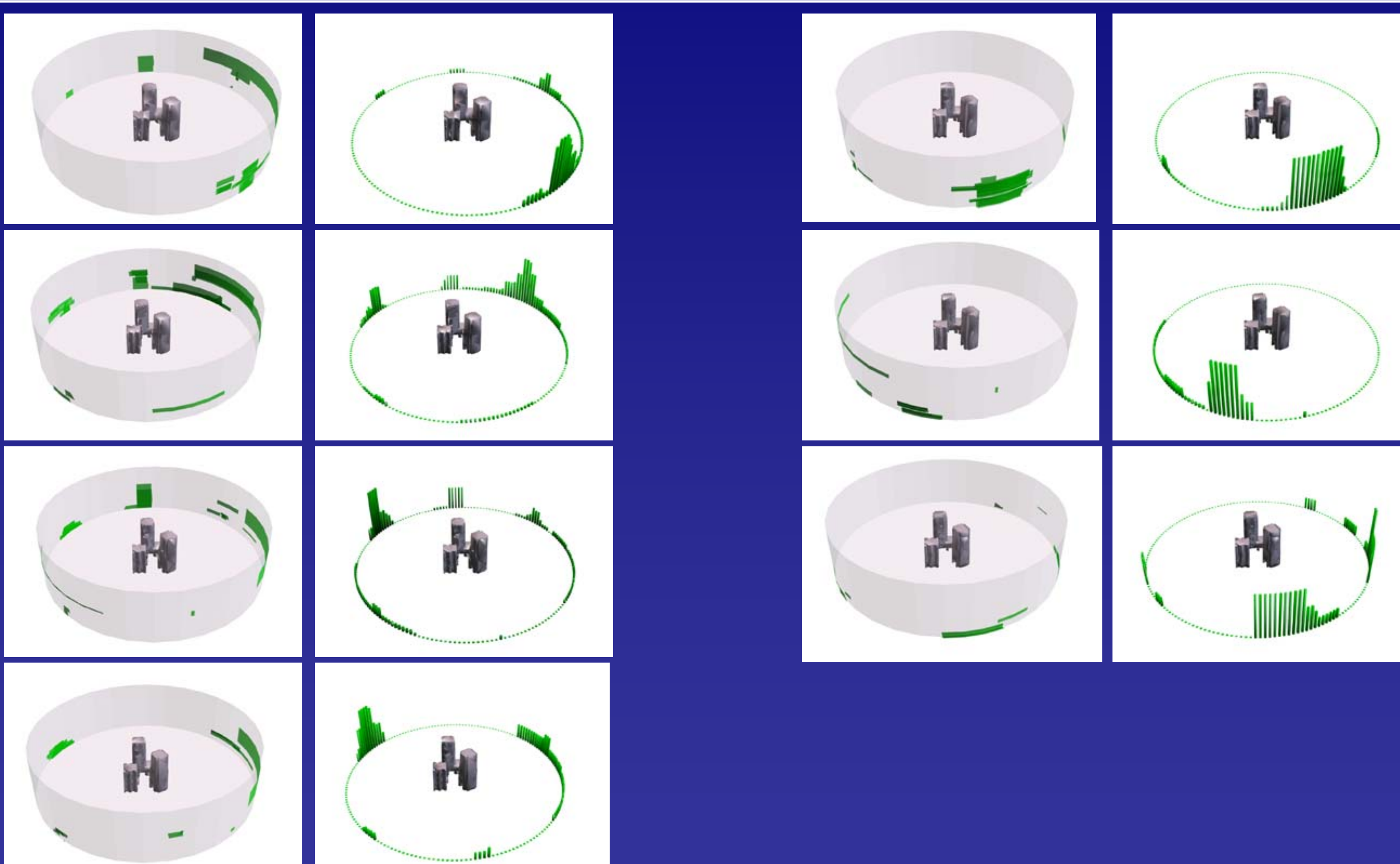


City Model: Placement Constraint

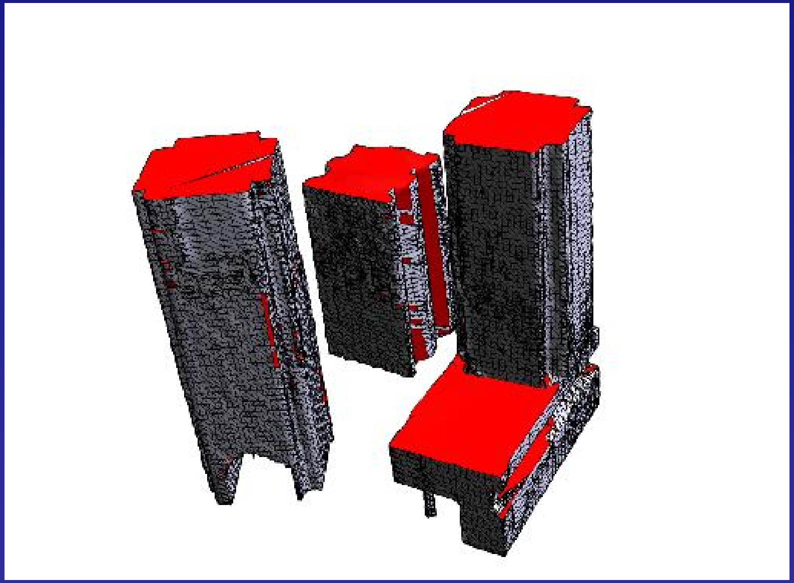


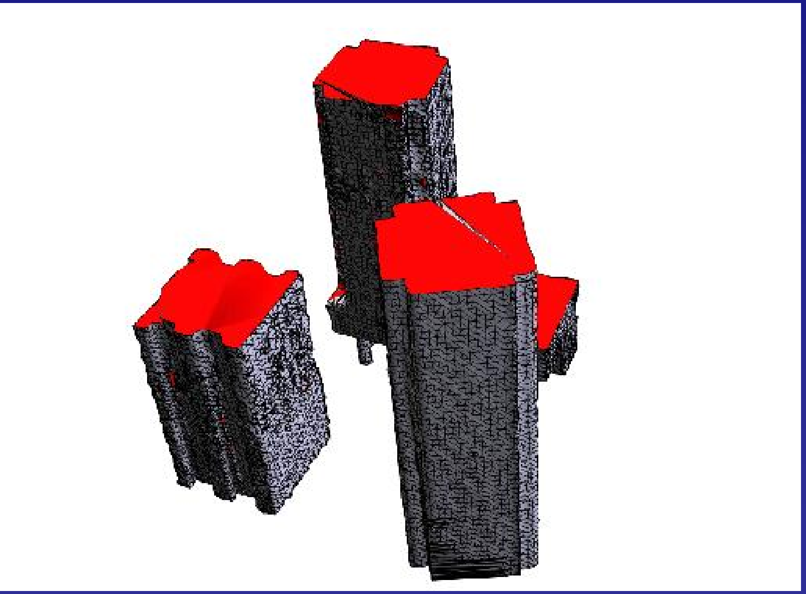


City Model: Total Plan

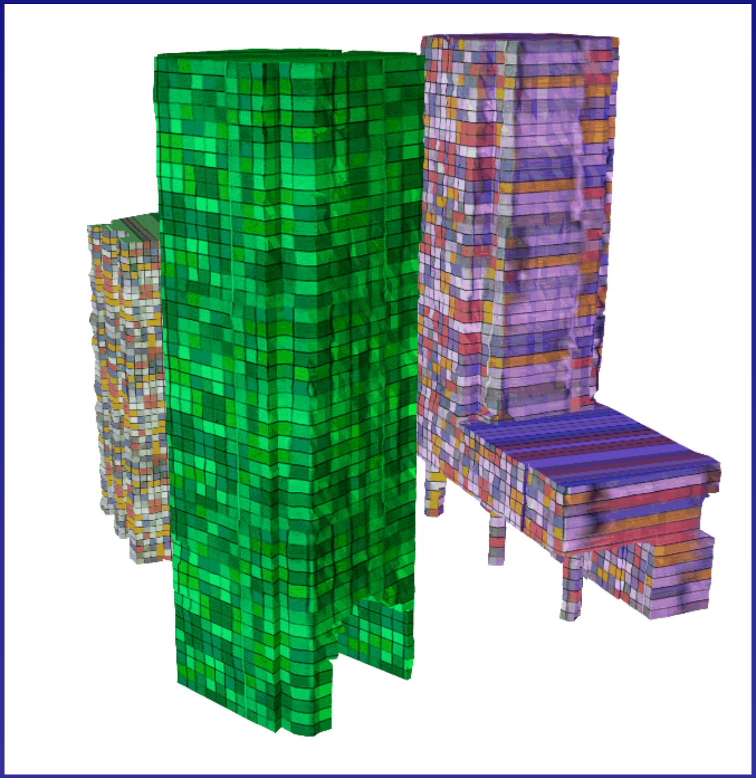


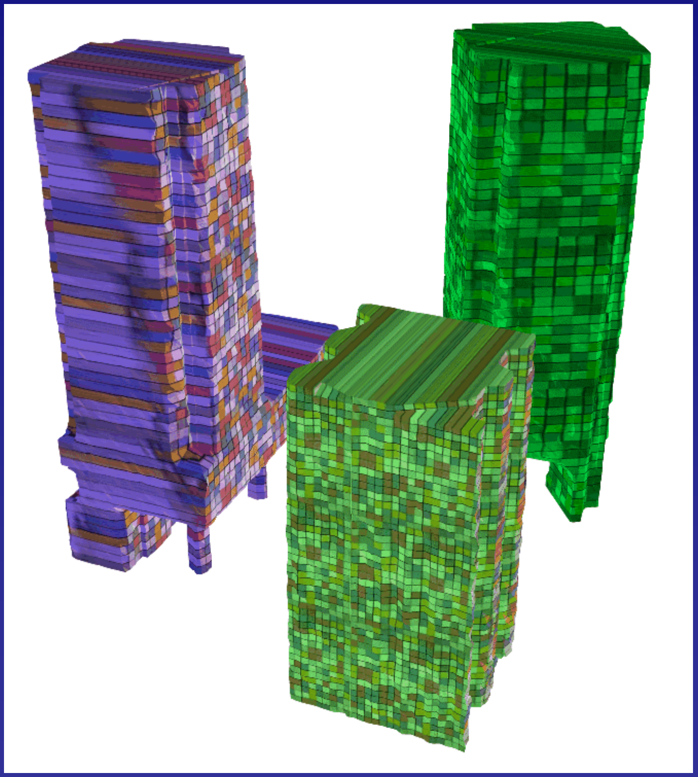
City Model: Final Reconstruction





City Model: Texture Mapped





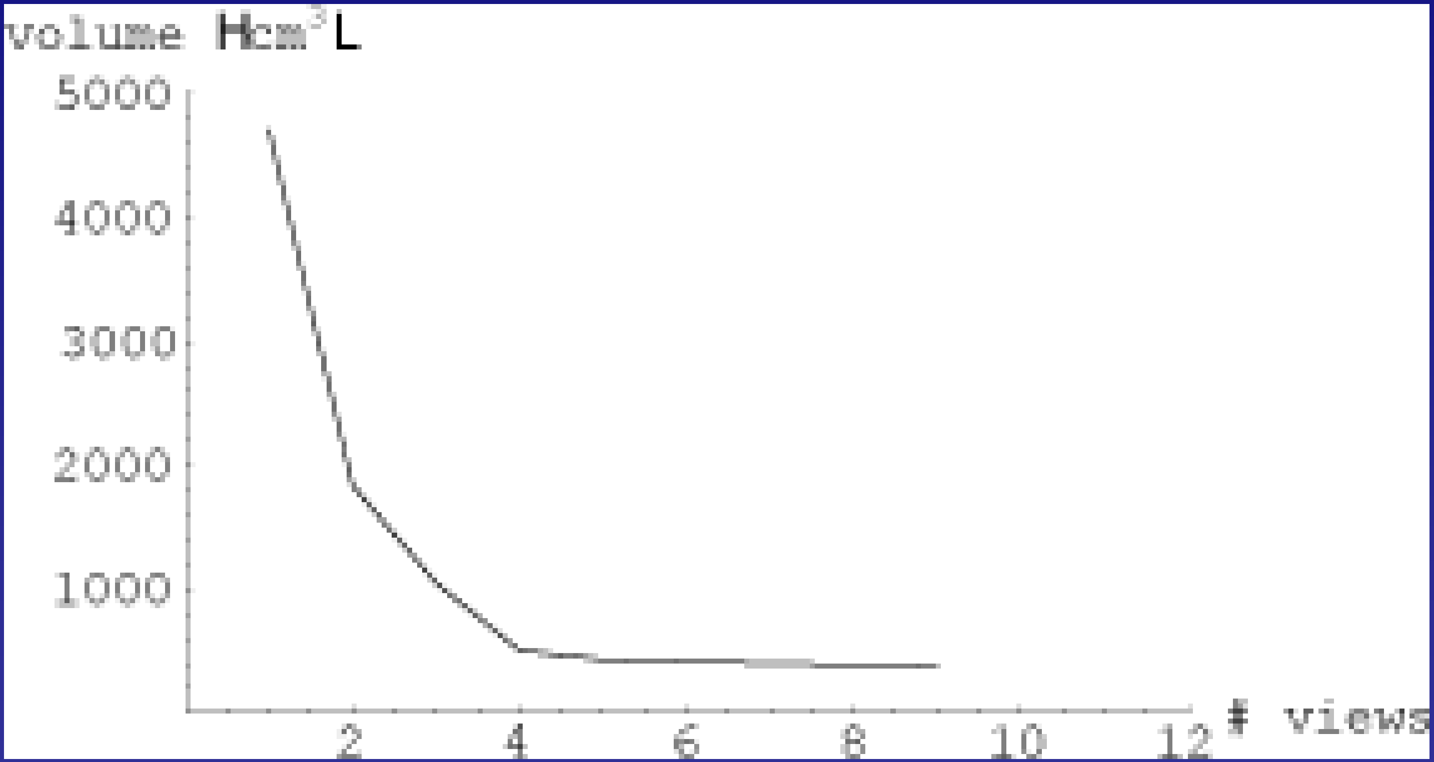
City Model: Analysis

View	Volume	Surface Area	Occ. Area	Plan Area	% Planned
1	4712	1571	1571	-	-
2	1840	1317	942	-	-
3	1052	1151	590	-	-
4	432	658	140	-	-
5	416	656	121	61	50%
6	404	659	104	28	27%
7	391	657	90	12	13%
8	386	647	84	8	10%
9	382	644	75	15	20%
10	380	651	62	7	11%
11	374	622	53	16	30%
12	370	604	36	9	25%

Efficiency Considerations

- Modeling
  - restricted set union mesh “sweep” operation
  - problem subdivision is set union
- Planning:
  - target surface decimation
  - occlusion constraints calculated only for model surfaces *within* imaging constraints volume

City Model: Analysis



Limitations

- Requires “surrounding” views
- No consideration of overlapping sampling in distinct images
- Planning does not consider partial visibility
- Requires a calibrated system

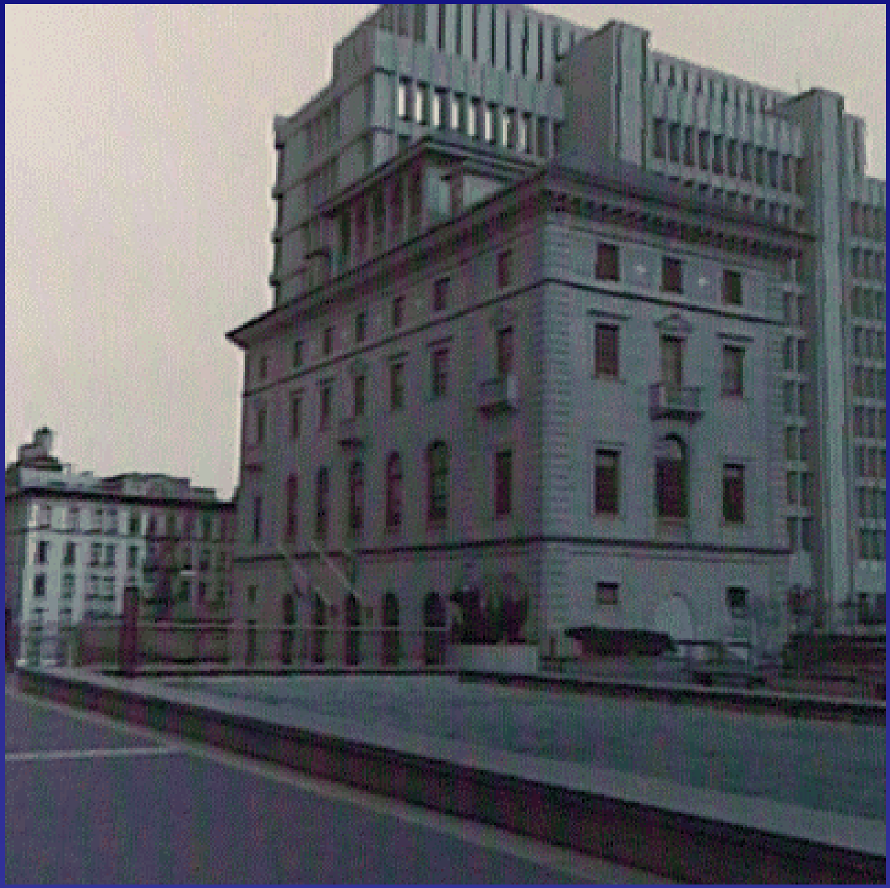
Conclusion

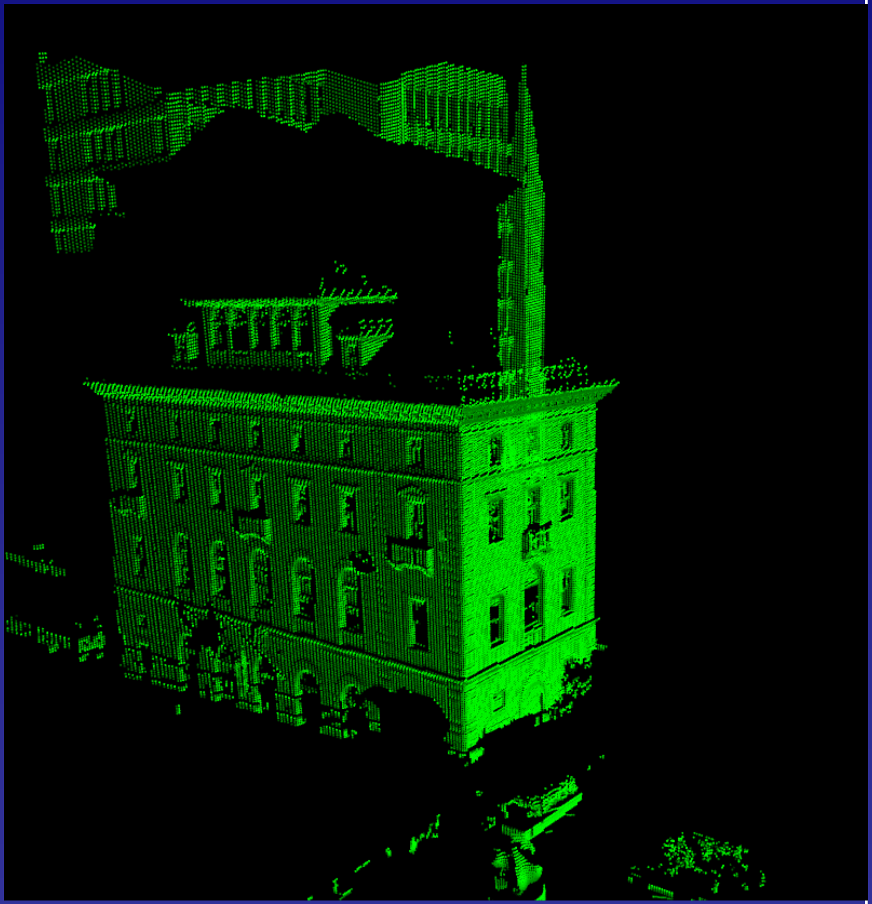
- The system presented has the following attributes:
  - is incremental
  - uses hybrid of both mesh and volume representations
  - represents both imaged and unimaged surfaces for planning the next view
  - determines a solid model at each step

Future Research

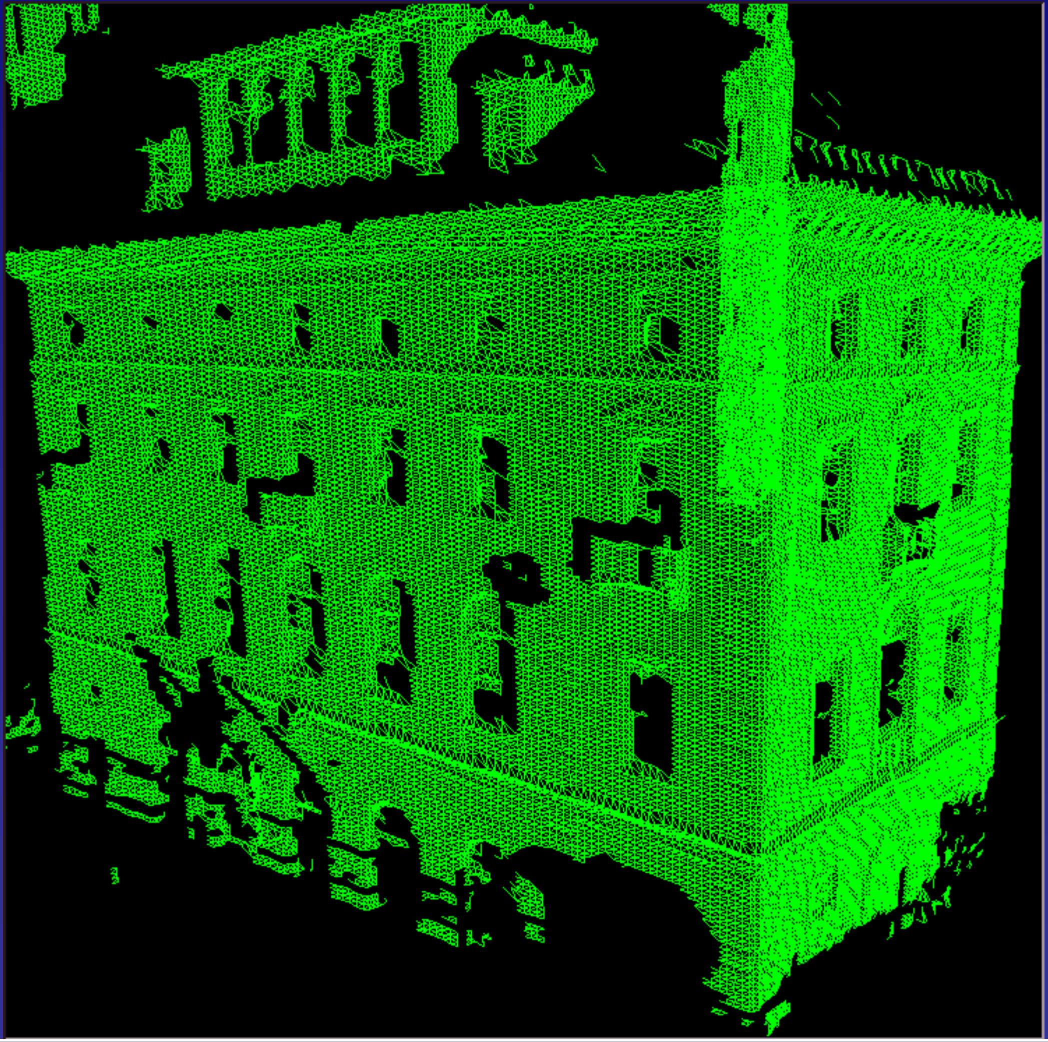
- Outdoor Scene Modeling
- Reduce number of full-model intersections
  - subtraction for some modeling operations
  - extension to other environments
- Optimal viewpoint and target selection
  - partial visibility
- Reduction of model to “canonical” form
  - use symmetry, features, heuristics
  - consider common design principles

Range Scanning Outdoor Structures

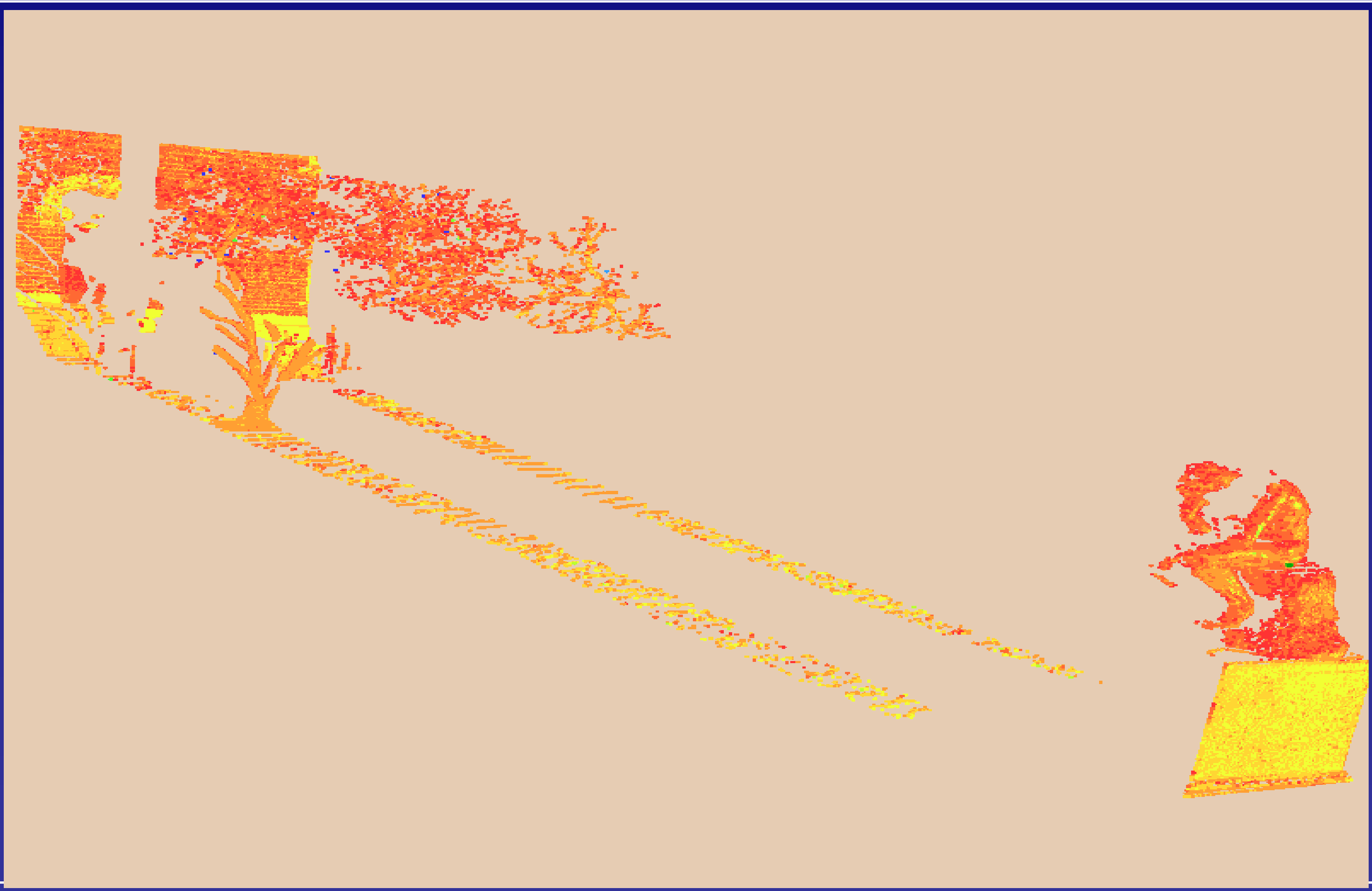




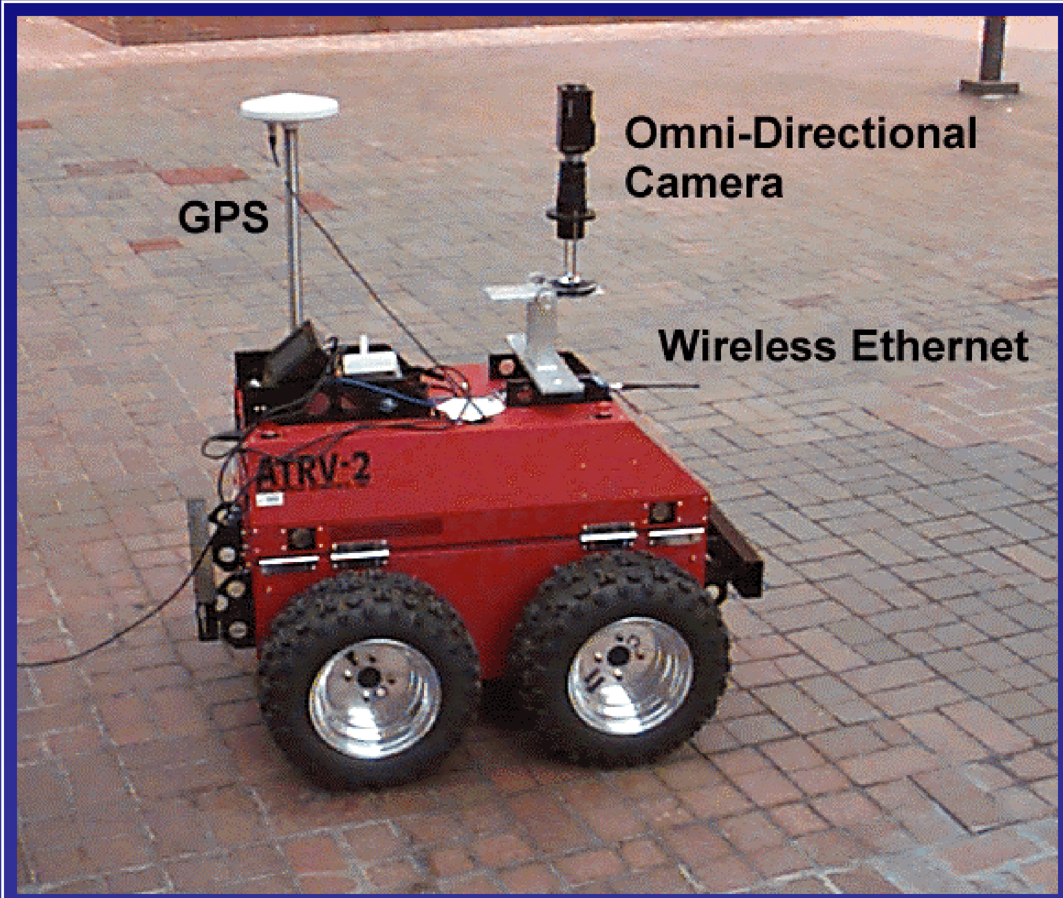
Mesh Surface



‘Scan of Rodin’s **The Thinker**



MOBILE ROBOT FOR SITE MODELING



- Outdoor Mobile Base
- Centimeter accuracy GPS system
- Omni-directional 360º Color Camera
- 80 meter laser scanner
- Integrated sonar units
- Wireless ethernet connection