

3D photography

Digital Visual Effects, Spring 2008

Yung-Yu Chuang

2008/5/13

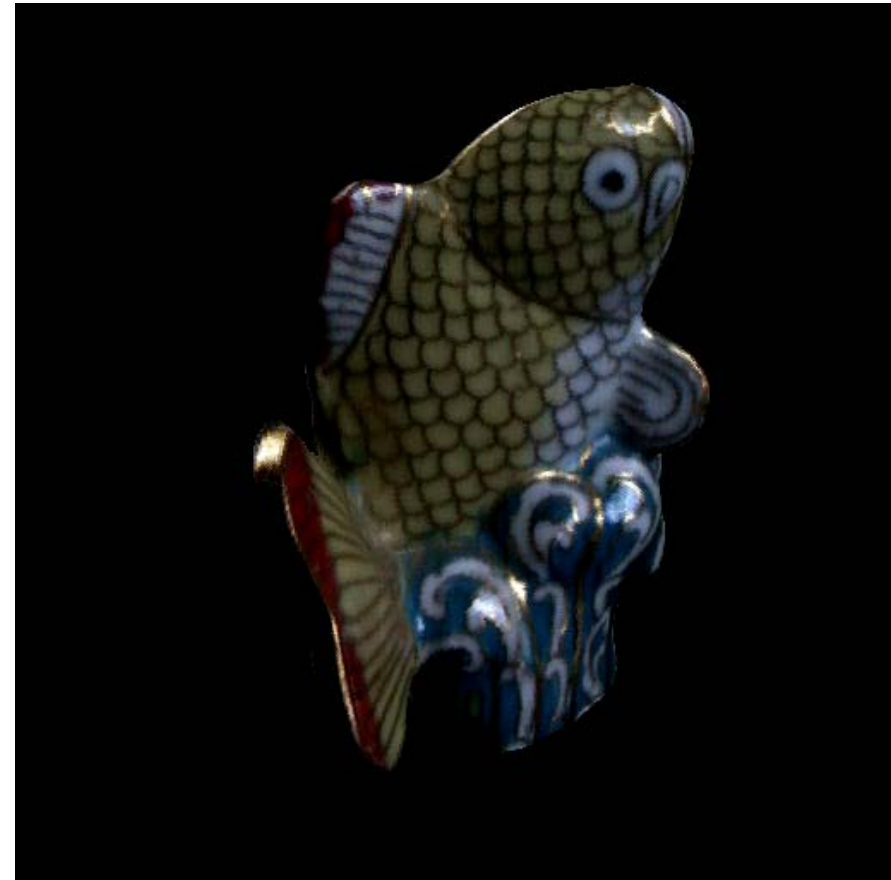
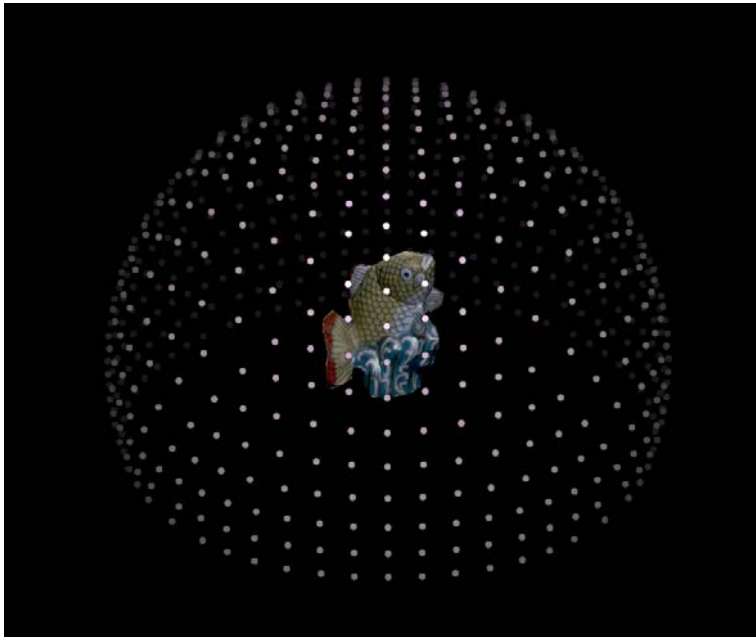
with slides by Szymon Rusinkiewicz, Richard Szeliski, Steve Seitz and Brian Curless

Announcements

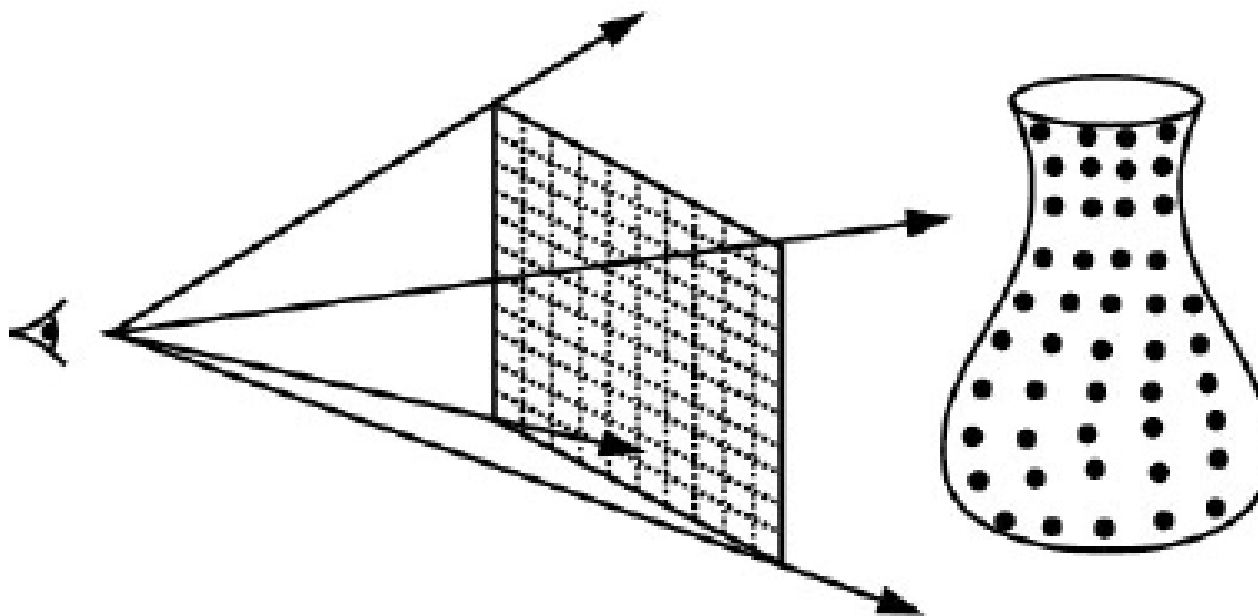
- Project #3 is due on 5/19. Two videos. Send us links only.
- We may have final project proposal presentation on 5/27. Please send me your team members and topic by 5/26.
- Final project demo day will be 1:30pm on 6/25 (Wed). Room to be announced.

3D photography

- Acquisition of **geometry** and material

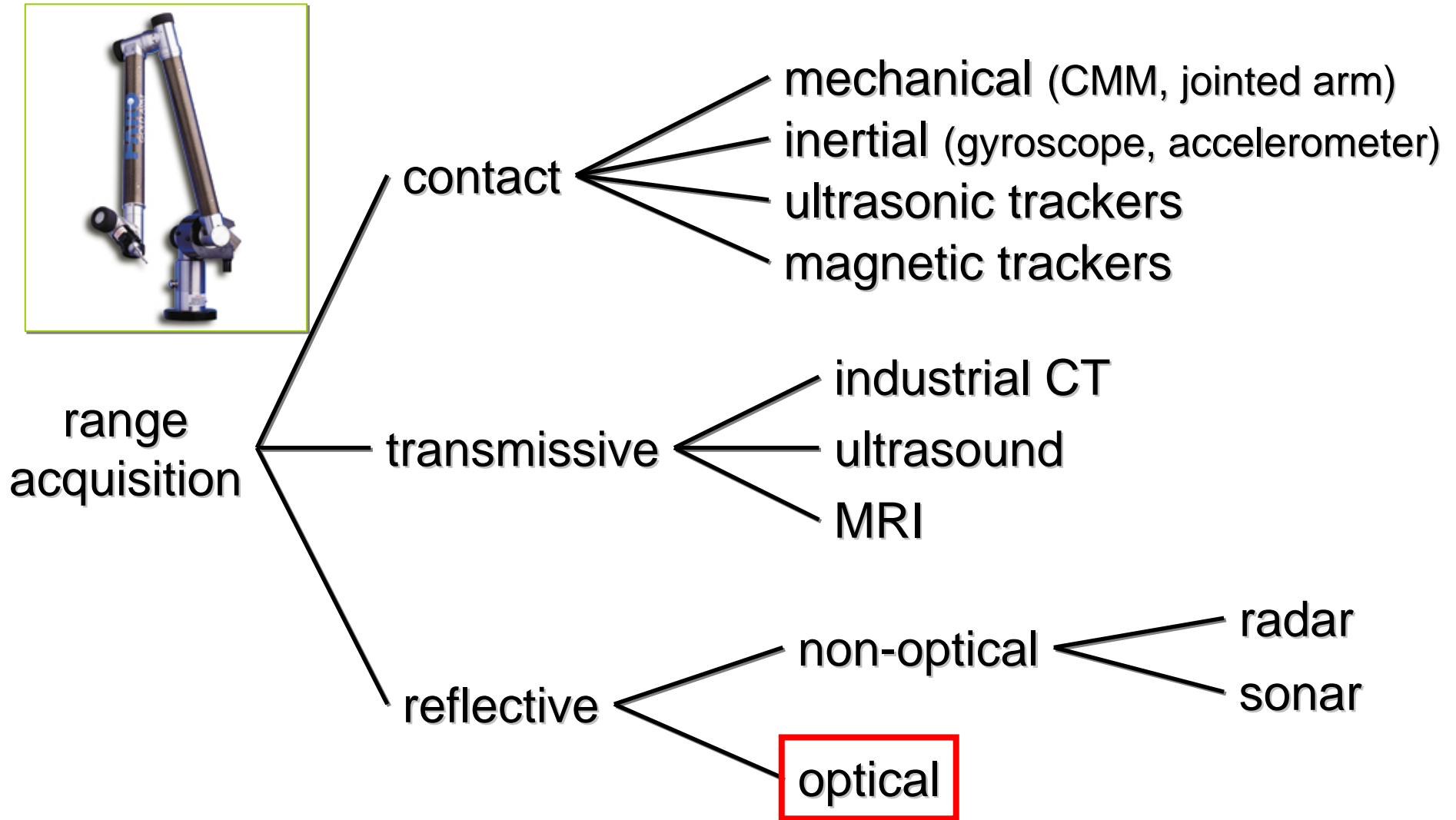


Range acquisition

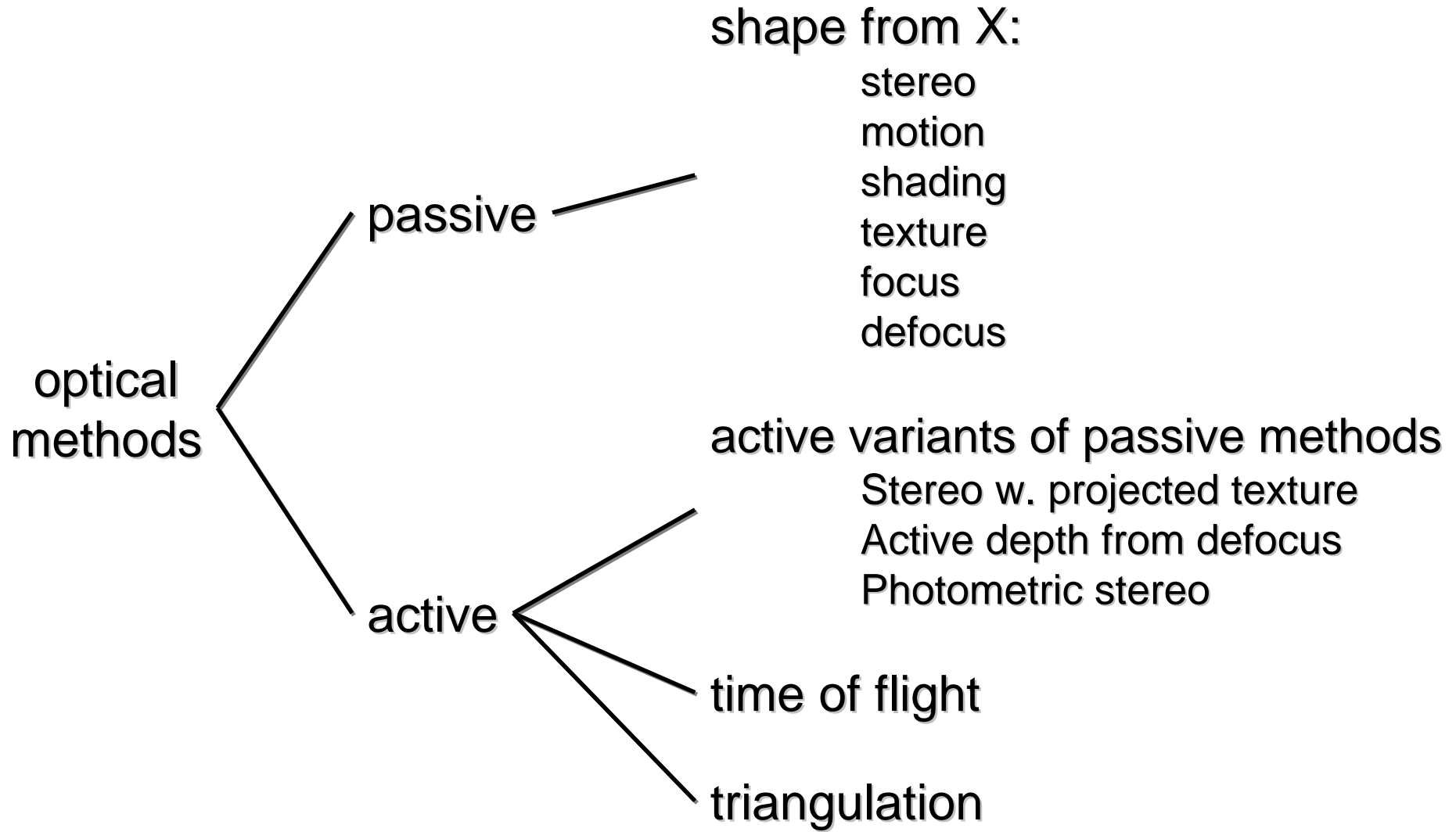


Range image

Range acquisition taxonomy



Range acquisition taxonomy



Outline

- Passive approaches
 - Stereo
 - Multiview approach
- Active approaches
 - Triangulation
 - Shadow scanning
- Active variants of passive approaches
 - Photometric stereo
 - Example-based photometric stereo

Passive approaches

Stereo

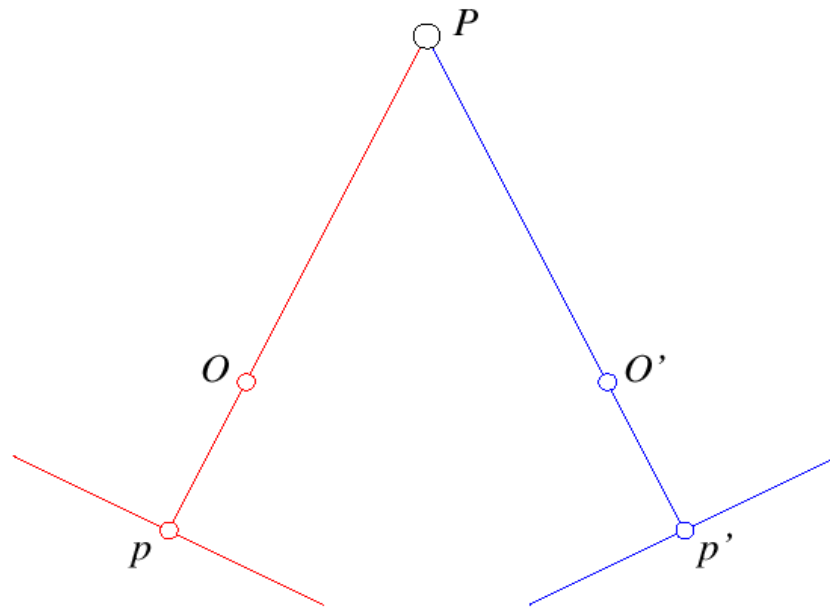


Public Library, Stereoscopic Looking Room, Chicago, by Phillips, 1923



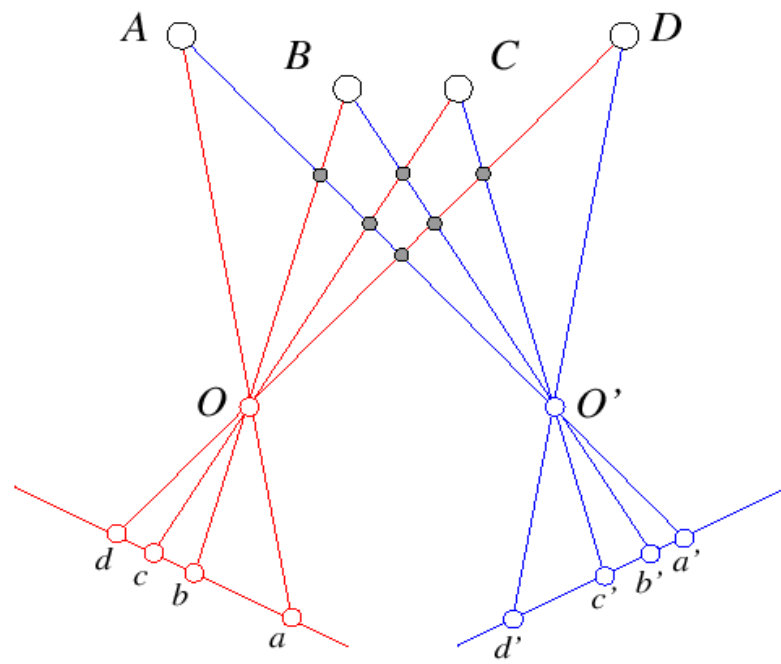
Stereo

- One distinguishable point being observed
 - The preimage can be found at the intersection of the rays from the focal points to the image points



Stereo

- Many points being observed
 - Need some method to establish correspondences



Components of stereo vision systems

- Camera calibration
- Image rectification: simplifies the search for correspondences
- Correspondence: which item in the left image corresponds to which item in the right image
- Reconstruction: recovers 3-D information from the 2-D correspondences

Epipolar geometry

- Epipolar constraint: corresponding points must lie on conjugate epipolar lines
 - Search for correspondences becomes a 1-D problem

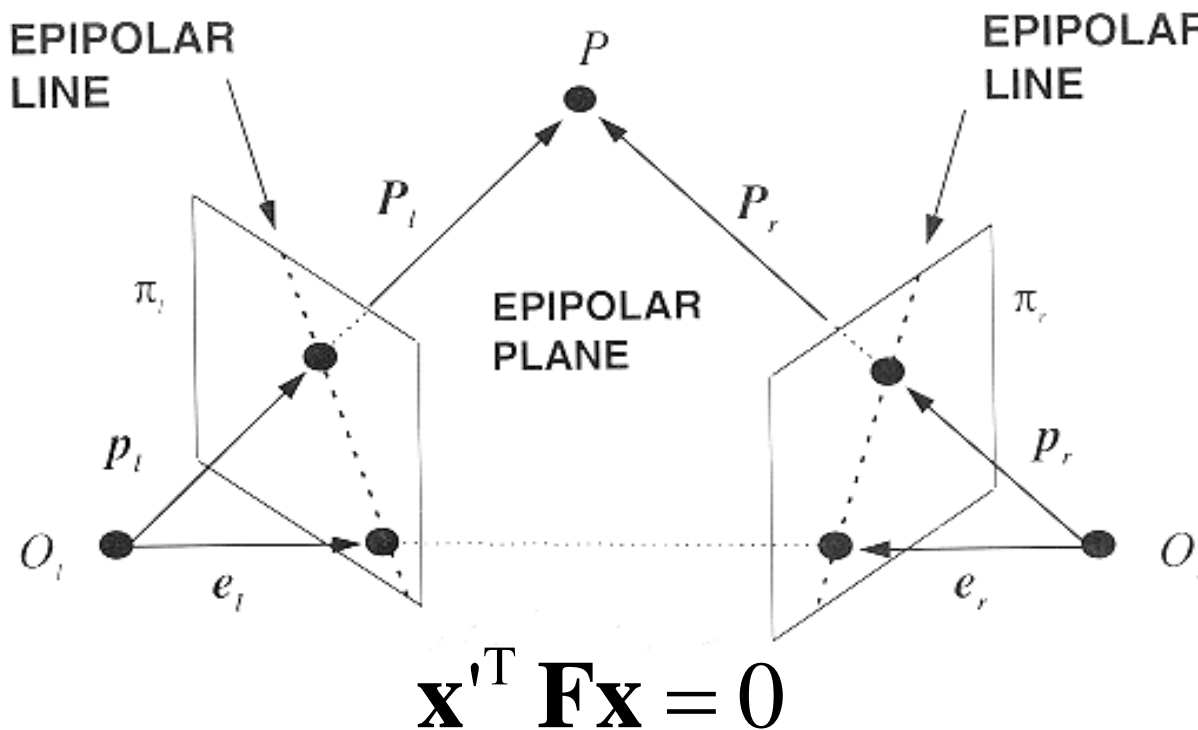
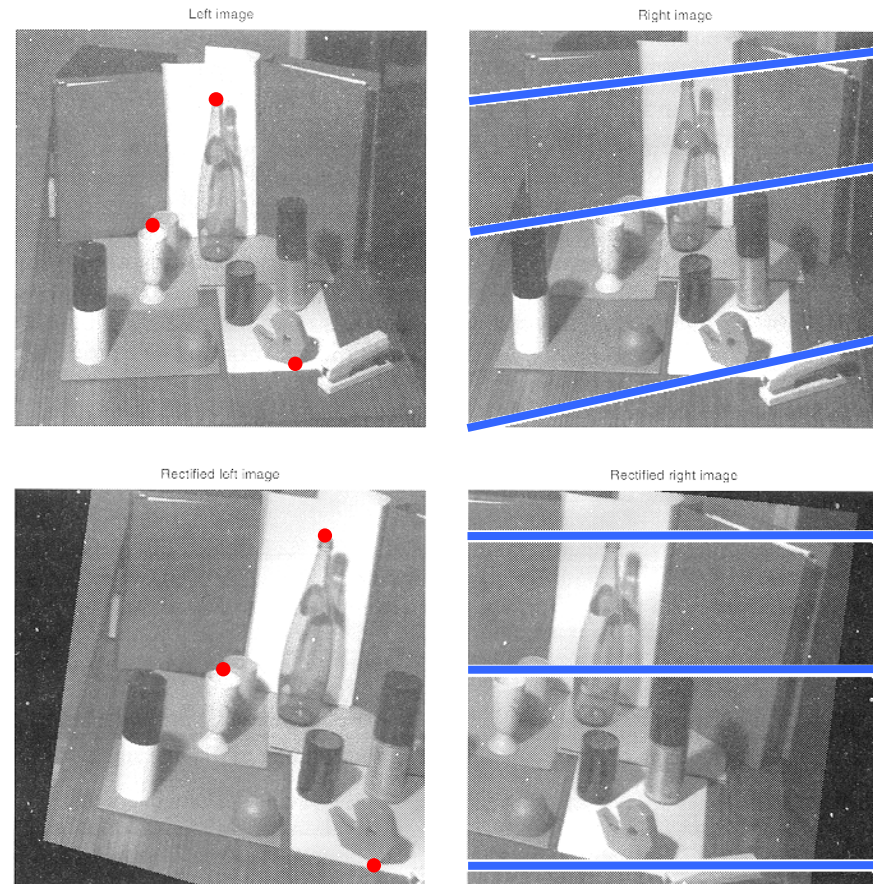


Image rectification

- Warp images such that conjugate epipolar lines become collinear and parallel to u axis

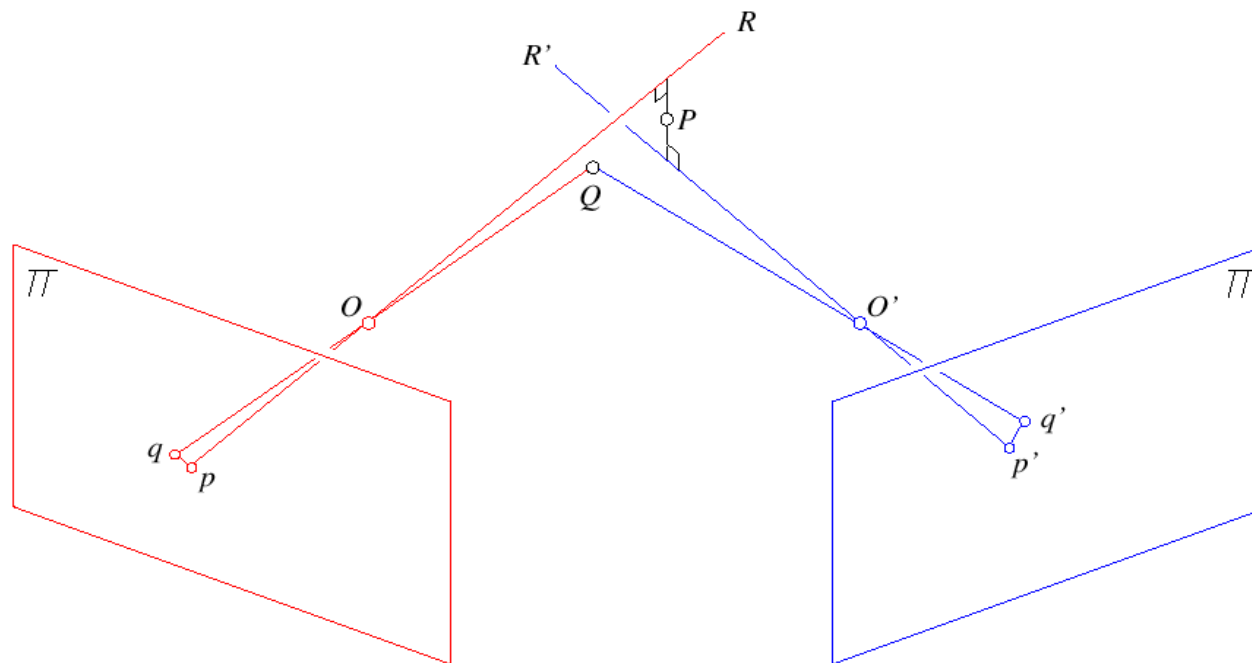


Disparity

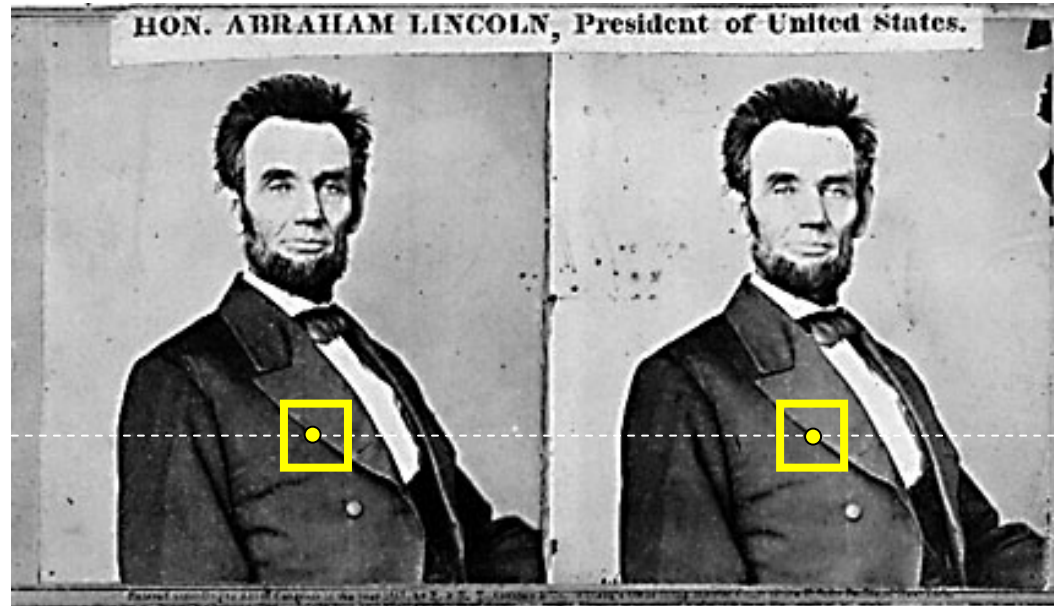
- With rectified images, disparity is just (horizontal) displacement of corresponding features in the two images
 - Disparity = 0 for distant points
 - Larger disparity for closer points
 - Depth of point proportional to $1/\text{disparity}$

Reconstruction

- Geometric
 - Construct the line segment perpendicular to R and R' that intersects both rays and take its mid-point



Basic stereo algorithm



For each epipolar line

For each pixel in the left image

- compare with every pixel on same epipolar line in right image
- pick pixel with minimum match cost

Improvement: match *windows*

Basic stereo algorithm

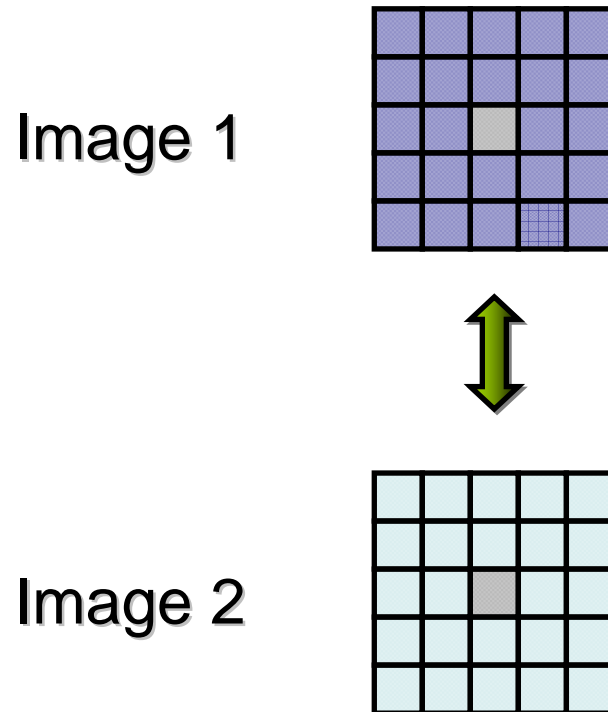
- For each pixel
 - For each disparity
 - For each pixel in window
 - Compute difference
 - Find disparity with minimum SSD

Reverse order of loops

- For each disparity
 - For each pixel
 - For each pixel in window
 - Compute difference
- Find disparity with minimum SSD at each pixel

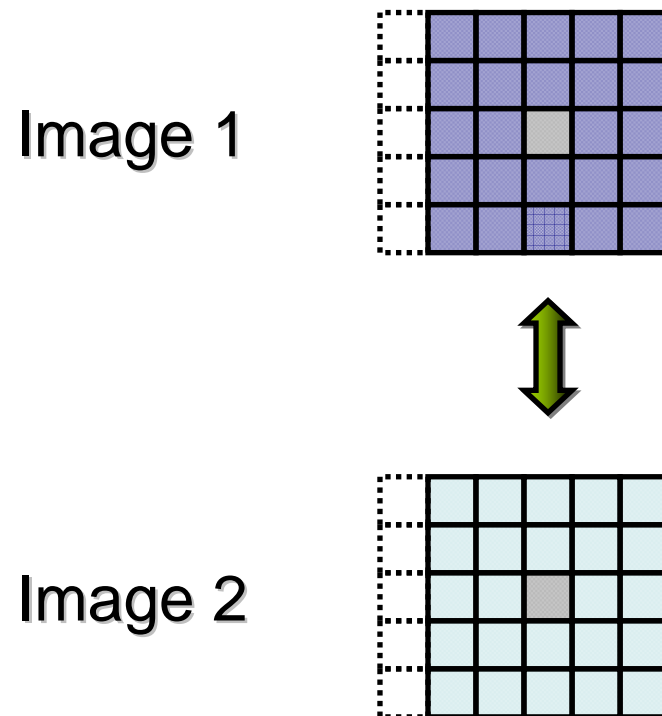
Incremental computation

- Given SSD of a window, at some disparity



Incremental computation

- Want: SSD at next location



Incremental computation

- Subtract contributions from leftmost column, add contributions from rightmost column

Image 1

-					+
-					+
-					+
-					+
-					+

Image 2

-					+
-					+
-					+
-					+
-					+

Selecting window size

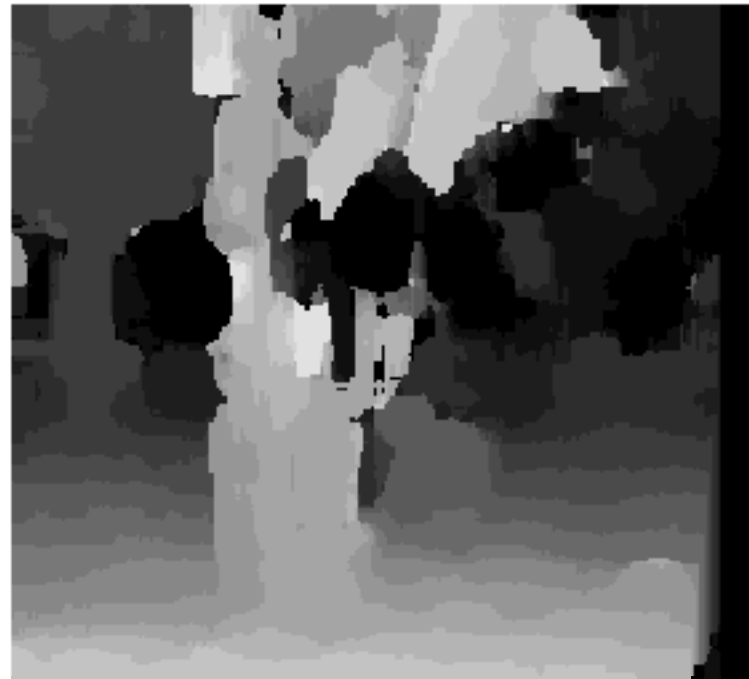
- Small window: more detail, but more noise
- Large window: more robustness, less detail
- Example:



Selecting window size



3 pixel window



20 pixel window

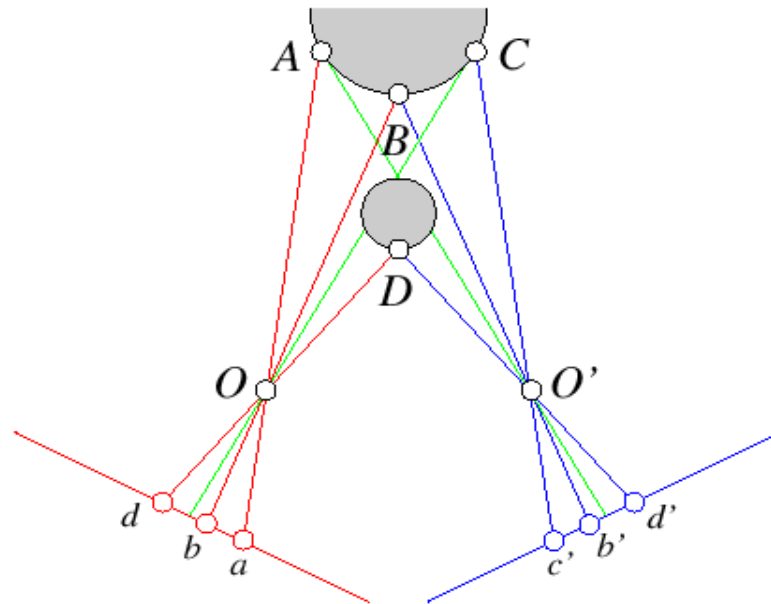
Non-square windows

- Compromise: have a large window, but higher weight near the center
- Example: Gaussian
- Example: Shifted windows



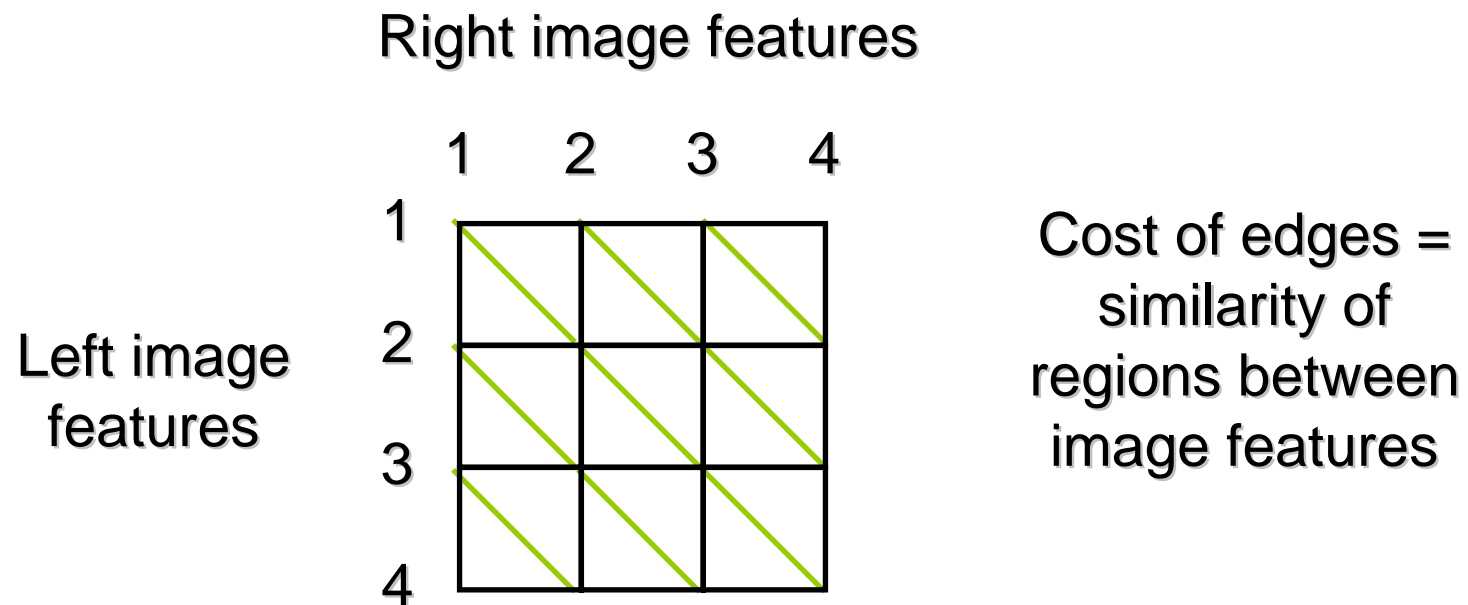
Ordering constraint

- Order of matching features usually the same in both images
- But not always: occlusion



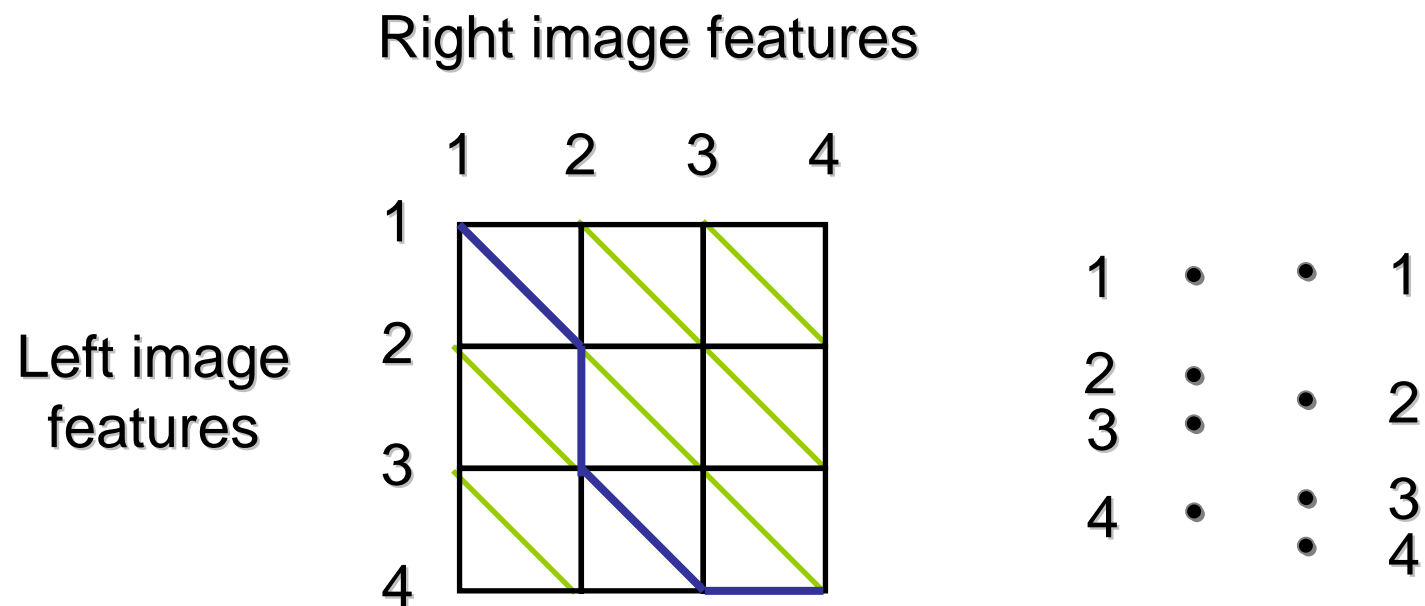
Dynamic programming

- Treat feature correspondence as graph problem



Dynamic programming

- Find min-cost path through graph



Energy minimization

- Another approach to improve quality of correspondences
- Assumption: disparities vary (mostly) smoothly
- Minimize energy function:

$$E_{\text{data}} + \lambda E_{\text{smoothness}}$$

- E_{data} : how well does disparity match data
- $E_{\text{smoothness}}$: how well does disparity match that of neighbors – regularization

Energy minimization

- If data and energy terms are nice (continuous, smooth, etc.) can try to minimize via gradient descent, etc.
- In practice, disparities only piecewise smooth
- Design smoothness function that doesn't penalize large jumps too much
 - Example: $V(\alpha, \beta) = \min(|\alpha - \beta|, K)$

Stereo as energy minimization

- Matching Cost Formulated as Energy
 - “data” term penalizing bad matches

$$D(x, y, d) = |\mathbf{I}(x, y) - \mathbf{J}(x + d, y)|$$

- “neighborhood term” encouraging spatial smoothness

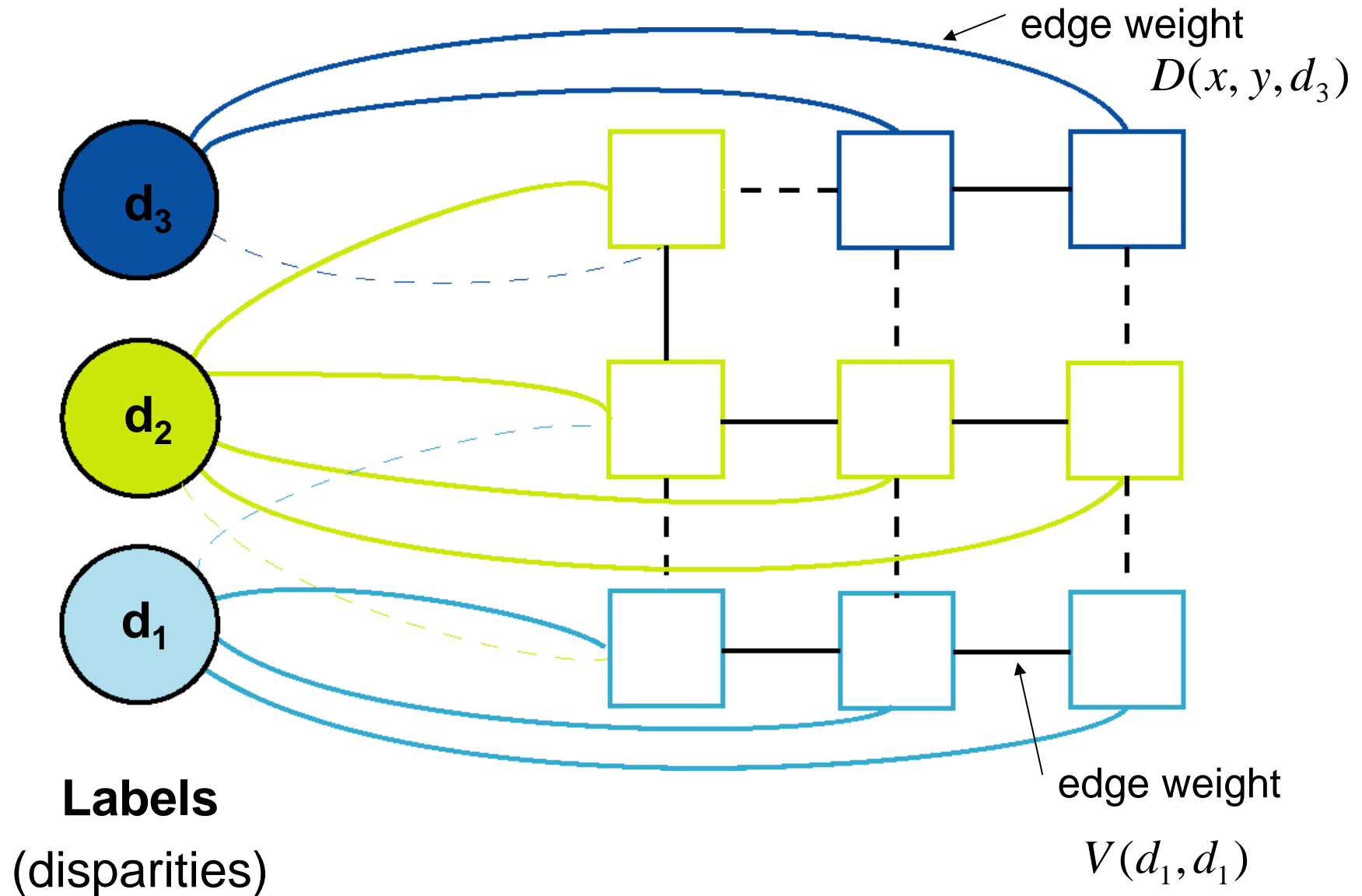
$$\begin{aligned} V(d_1, d_2) &= \text{cost of adjacent pixels with labels } d_1 \text{ and } d_2 \\ &= |d_1 - d_2| \quad (\text{or something similar}) \end{aligned}$$

$$E = \sum_{(x,y)} D(x, y, d_{x,y}) + \sum_{\text{neighbors } (x_1,y_1),(x_2,y_2)} V(d_{x_1,y_1}, d_{x_2,y_2})$$

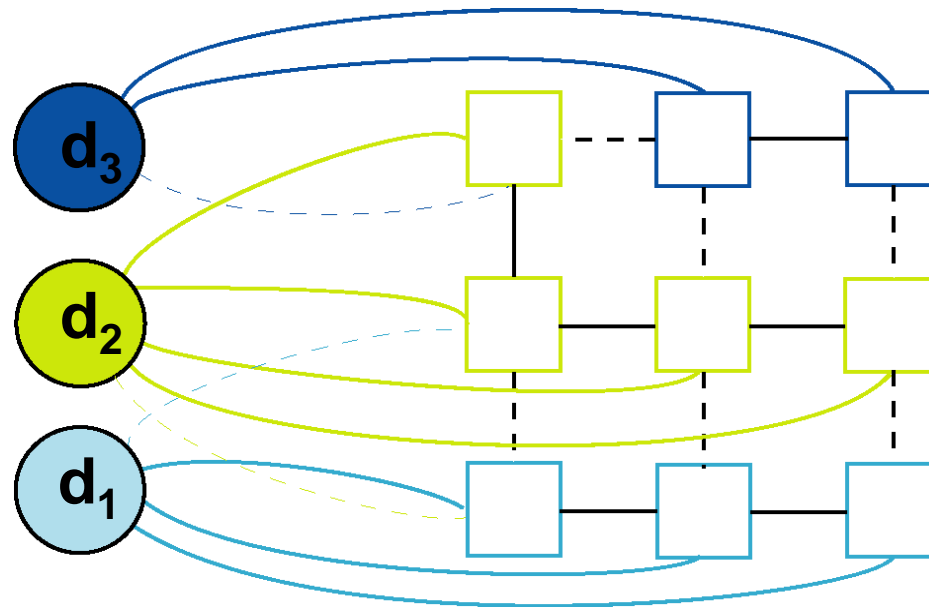
Energy minimization

- Hard to find global minima of non-smooth functions
 - Many local minima
 - Provably NP-hard
- Practical algorithms look for approximate minima (e.g., simulated annealing)

Energy minimization via graph cuts

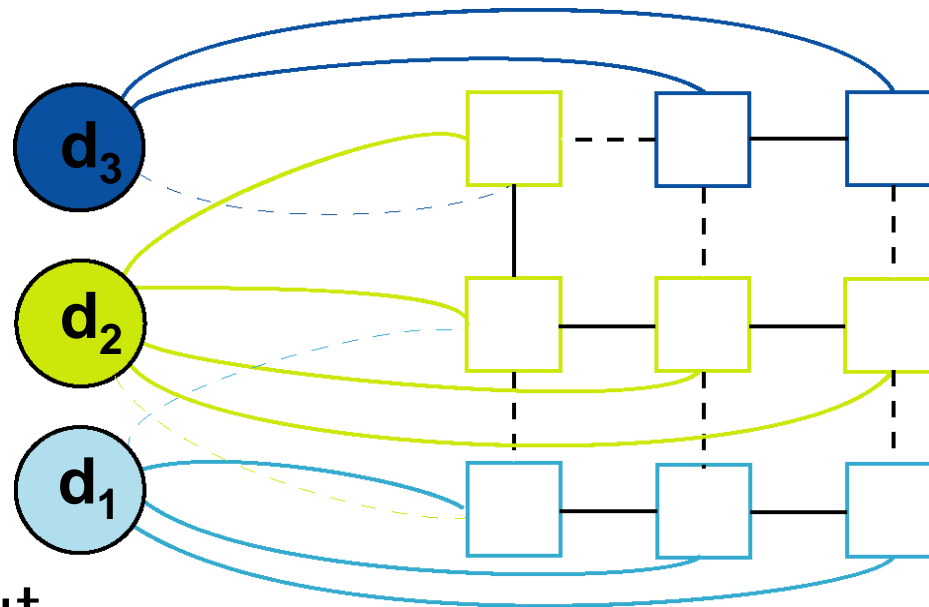


Energy minimization via graph cuts



- Graph Cost
 - Matching cost between images
 - Neighborhood matching term
 - Goal: figure out which labels are connected to which pixels

Energy minimization via graph cuts

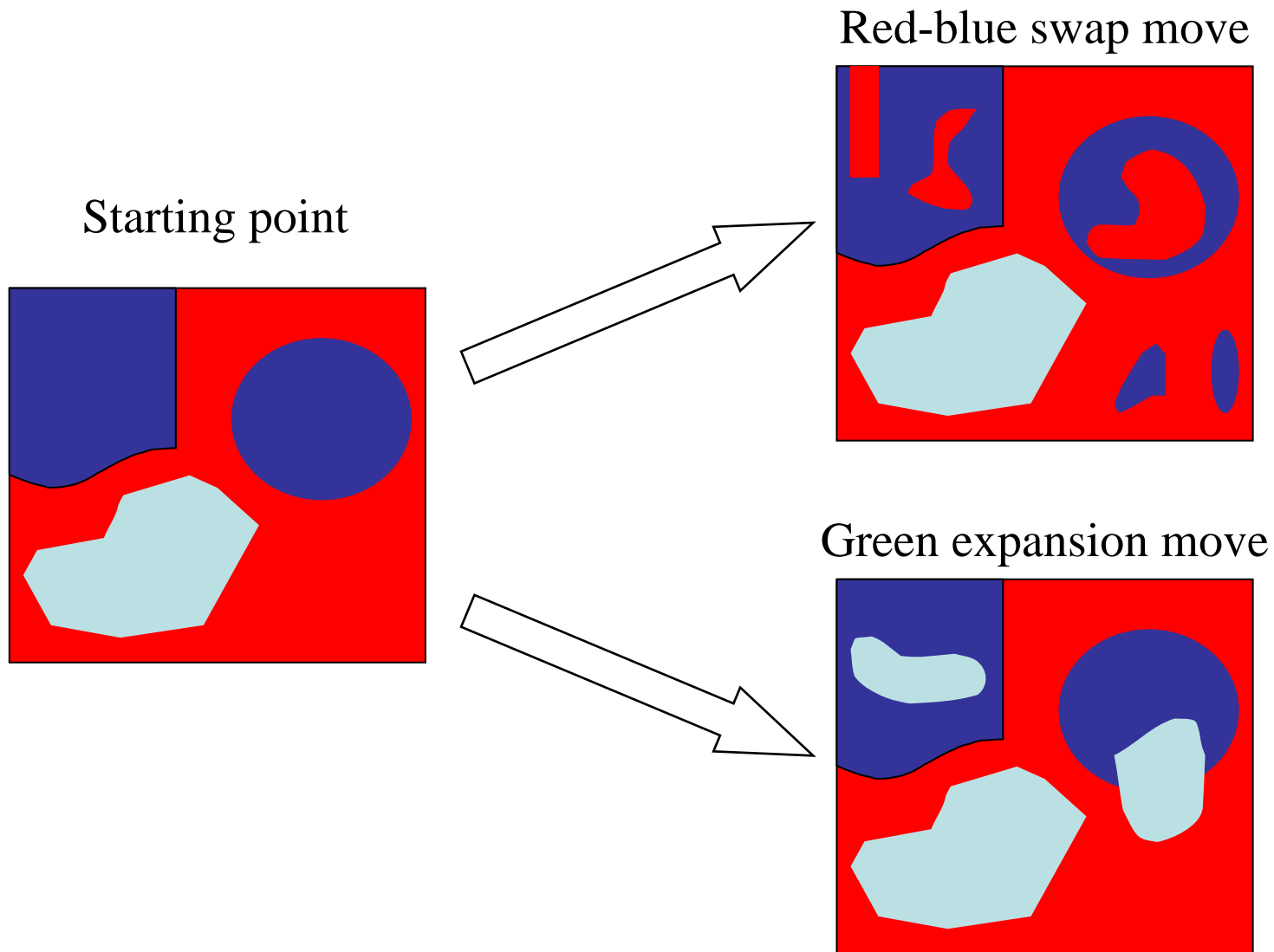


- Graph Cut
 - Delete enough edges so that
 - each pixel is (transitively) connected to exactly one label node
 - Cost of a cut: sum of deleted edge weights
 - Finding min cost cut equivalent to finding global minimum of energy function

Computing a multiway cut

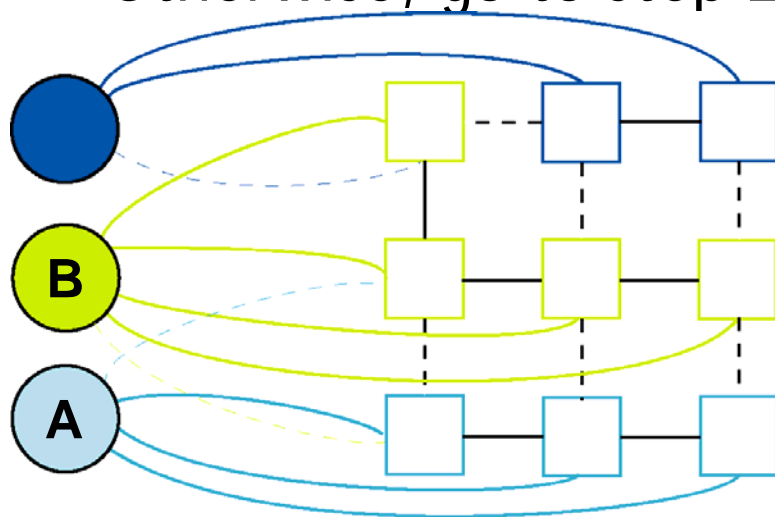
- With 2 labels: classical min-cut problem
 - Solvable by standard flow algorithms
 - polynomial time in theory, nearly linear in practice
 - More than 2 terminals: NP-hard
 - [Dahlhaus *et al.*, STOC '92]
- Efficient approximation algorithms exist
 - Within a factor of 2 of optimal
 - Computes local minimum in a strong sense
 - even very large moves will not improve the energy
 - Yuri Boykov, Olga Veksler and Ramin Zabih, [Fast Approximate Energy Minimization via Graph Cuts](#), International Conference on Computer Vision, September 1999.

Move examples

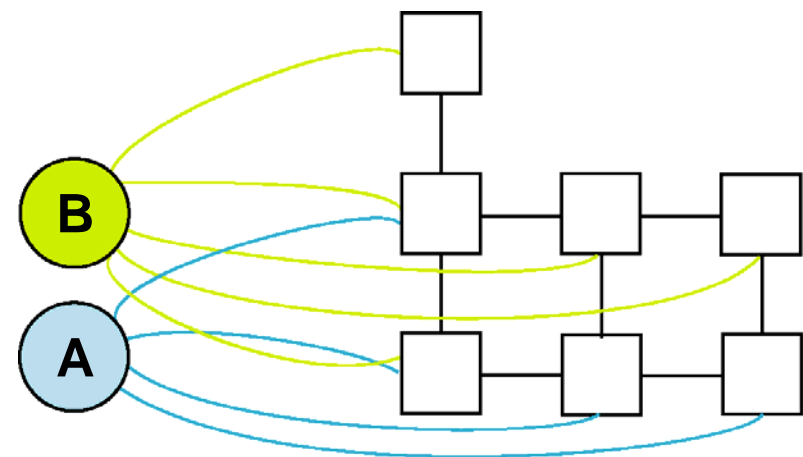


The swap move algorithm

1. Start with an arbitrary labeling
2. Cycle through every label pair (A, B) in some order
 - 2.1 Find the lowest E labeling within a single AB -swap
 - 2.2 Go there if E is lower than the current labeling
3. If E did not decrease in the cycle, we're done
Otherwise, go to step 2



Original graph



AB subgraph
(run min-cut on this graph)

The expansion move algorithm

1. Start with an arbitrary labeling
2. Cycle through every label A in some order
 - 2.1 Find the lowest E labeling within a single A -expansion
 - 2.2 Go there if it E is lower than the current labeling
3. If E did not decrease in the cycle, we're done
Otherwise, go to step 2

Stereo results

- Data from University of Tsukuba



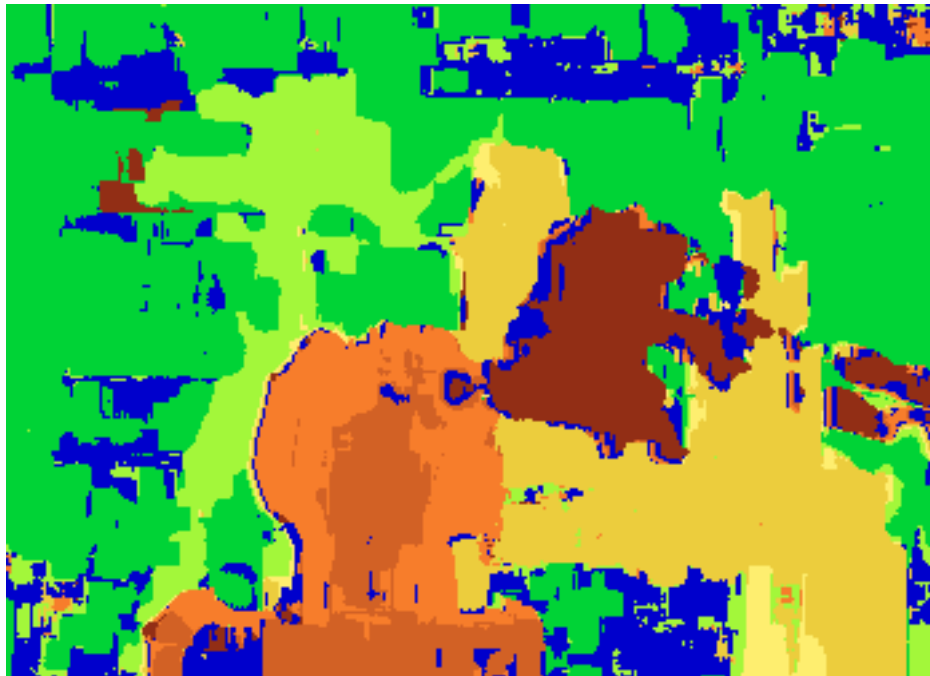
scene



ground truth

<http://cat.middlebury.edu/stereo/>

Results with window correlation



normalized correlation
(best window size)



ground truth

Results with graph cuts



graph cuts
(Potts model E ,
expansion move algorithm)



ground truth

Stereo evaluation

vision.middlebury.edu

stereo • mview • MRF • flow

Stereo

Evaluation • Datasets • Code • Submit

[Daniel Scharstein](#) • [Richard Szeliski](#)

Welcome to the Middlebury Stereo Vision Page, formerly located at www.middlebury.edu/stereo. This website accompanies our taxonomy and comparison of two-frame stereo correspondence algorithms [1]. It contains:

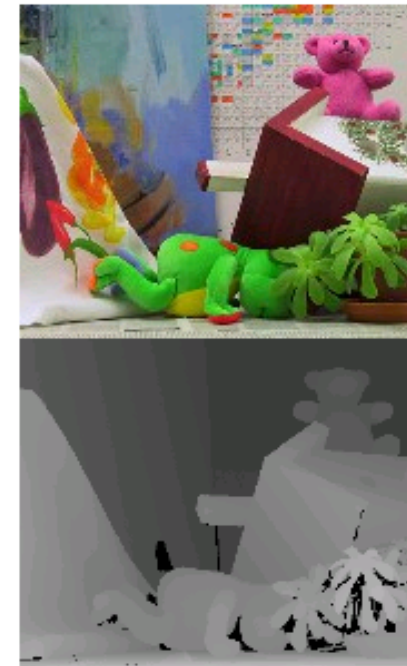
- An [on-line evaluation](#) of current algorithms
- Many [stereo datasets](#) with ground-truth disparities
- Our [stereo correspondence software](#)
- An [on-line submission script](#) that allows you to evaluate your stereo algorithm in our framework

How to cite the materials on this website:

We grant permission to use and publish all images and numerical results on this website. If you report performance results, we request that you cite our paper [1]. Instructions on how to cite our datasets are listed on the [datasets page](#). If you want to cite this website, please use the URL "vision.middlebury.edu/stereo/".

References:

- [1] D. Scharstein and R. Szeliski. [A taxonomy and evaluation of dense two-frame stereo correspondence algorithms](#). *International Journal of Computer Vision*, 47(1/2/3):7-42, April-June 2002.
[Microsoft Research Technical Report MSR-TR-2001-81](#), November 2001.



Stereo—best algorithms

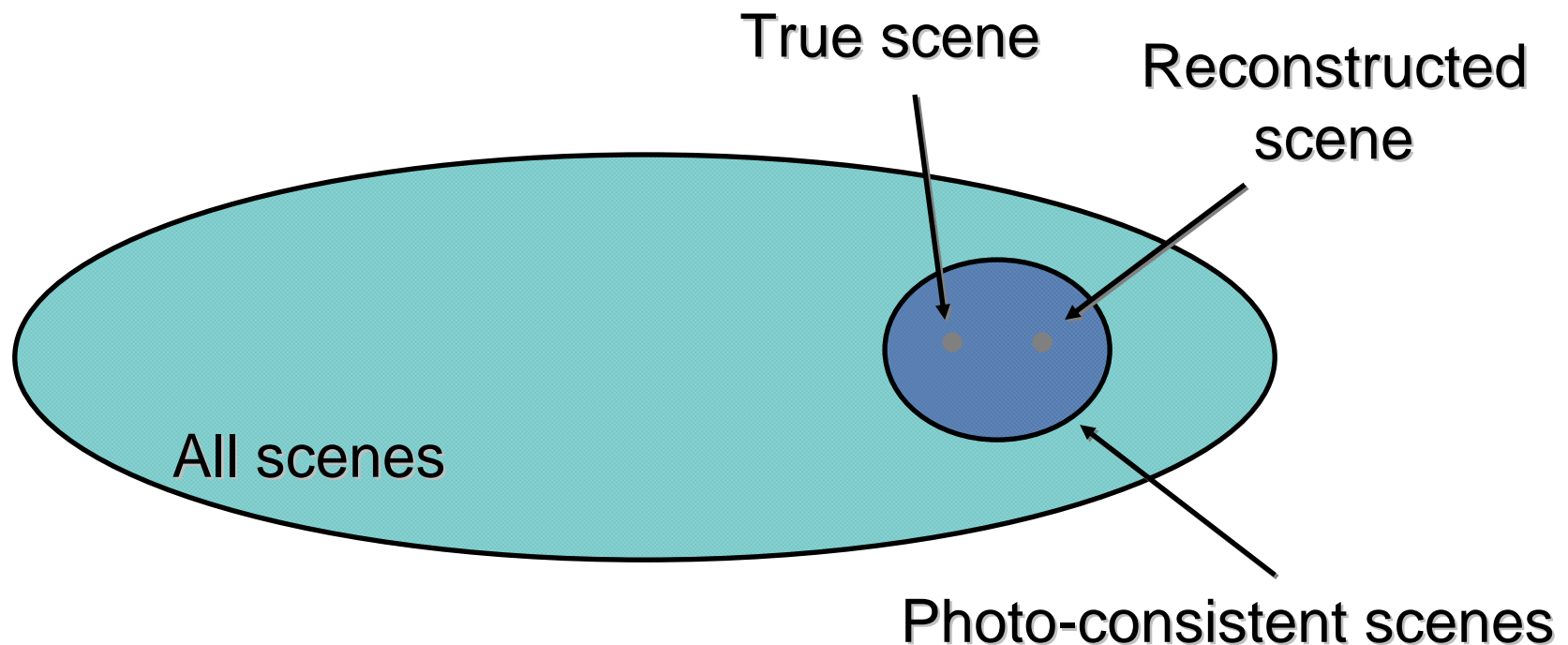
Error Threshold = 1		Sort by nonocc			Sort by all			Sort by disc					
Error Threshold... ▾		▼			▼			▼					
Algorithm	Avg.	<u>Tsukuba</u> ground truth			<u>Venus</u> ground truth			<u>Teddy</u> ground truth			<u>Cones</u> ground truth		
	Rank	nonocc	all	disc	nonocc	all	disc	nonocc	all	disc	nonocc	all	disc
<u>AdaptIngBP [17]</u>	2.8	<u>1.11</u> 6	1.37 3	5.79 7	<u>0.10</u> 1	0.21 2	<u>1.44</u> 1	<u>4.22</u> 4	7.06 2	11.8 4	<u>2.48</u> 1	7.92 2	<u>7.32</u> 1
<u>DoubleBP2 [35]</u>	2.9	<u>0.88</u> 1	<u>1.29</u> 1	<u>4.76</u> 1	<u>0.13</u> 3	0.45 5	1.87 5	<u>3.53</u> 2	8.30 3	<u>9.63</u> 1	<u>2.90</u> 3	8.78 8	7.79 2
<u>DoubleBP [15]</u>	4.9	<u>0.88</u> 2	1.29 2	4.76 2	<u>0.14</u> 5	0.60 13	2.00 7	<u>3.55</u> 3	8.71 5	9.70 2	<u>2.90</u> 4	9.24 11	7.80 3
<u>SubPixDoubleBP [30]</u>	5.6	<u>1.24</u> 10	1.76 13	5.98 8	<u>0.12</u> 2	0.46 6	1.74 4	<u>3.45</u> 1	8.38 4	10.0 3	<u>2.93</u> 5	8.73 7	7.91 4
<u>AdaptOvrSeqBP [33]</u>	9.9	<u>1.69</u> 22	2.04 21	5.64 6	<u>0.14</u> 4	<u>0.20</u> 1	1.47 2	<u>7.04</u> 14	11.1 7	16.4 11	<u>3.60</u> 11	8.96 10	8.84 10
<u>SvmBP+occ [7]</u>	10.8	<u>0.97</u> 4	1.75 12	5.09 4	<u>0.16</u> 6	0.33 3	2.19 8	<u>6.47</u> 8	10.7 6	17.0 14	<u>4.79</u> 24	10.7 21	10.9 20
<u>PlaneFitBP [32]</u>	10.8	<u>0.97</u> 5	1.83 14	5.26 5	<u>0.17</u> 7	0.51 8	1.71 3	<u>6.65</u> 9	12.1 13	14.7 7	<u>4.17</u> 20	10.7 20	10.6 19
<u>AdaptDispCalib [36]</u>	11.8	<u>1.19</u> 8	1.42 4	6.15 9	<u>0.23</u> 9	0.34 4	2.50 11	<u>7.80</u> 19	13.6 21	17.3 17	<u>3.62</u> 12	9.33 12	9.72 15
<u>Seqm+visib [4]</u>	12.2	<u>1.30</u> 15	1.57 5	6.92 18	<u>0.79</u> 21	1.06 18	6.76 22	<u>5.00</u> 5	<u>6.54</u> 1	12.3 5	<u>3.72</u> 13	8.62 6	10.2 17
<u>C-SemiGlob [19]</u>	12.3	<u>2.61</u> 29	3.29 24	9.89 27	<u>0.25</u> 12	0.57 10	3.24 15	<u>5.14</u> 6	11.8 8	13.0 6	<u>2.77</u> 2	8.35 4	8.20 5
<u>SO+borders [29]</u>	12.8	<u>1.29</u> 14	1.71 9	6.83 15	<u>0.25</u> 13	0.53 9	2.26 9	<u>7.02</u> 13	12.2 14	16.3 9	<u>3.90</u> 15	9.85 16	10.2 18
<u>DistinctSM [27]</u>	14.1	<u>1.21</u> 9	1.75 11	6.39 11	<u>0.35</u> 14	0.69 16	2.63 13	<u>7.45</u> 18	13.0 17	18.1 19	<u>3.91</u> 16	9.91 18	8.32 7
<u>CostAqqr+occ [39]</u>	14.3	<u>1.38</u> 17	1.96 17	7.14 19	<u>0.44</u> 16	1.13 19	4.87 19	<u>6.80</u> 11	11.9 10	17.3 16	<u>3.60</u> 10	8.57 5	9.36 13
<u>OverSeqmBP [26]</u>	14.5	<u>1.69</u> 23	1.97 18	8.47 24	<u>0.51</u> 18	0.68 15	4.69 18	<u>6.74</u> 10	11.9 12	15.8 8	<u>3.19</u> 8	8.81 9	8.89 11
<u>SegmentSupport [28]</u>	15.1	<u>1.25</u> 11	1.62 7	6.68 13	<u>0.25</u> 11	0.64 14	2.59 12	<u>8.43</u> 24	14.2 22	18.2 20	<u>3.77</u> 14	9.87 17	9.77 16
<u>RegionTreeDP [18]</u>	15.7	<u>1.39</u> 19	1.64 8	6.85 16	<u>0.22</u> 8	0.57 10	1.93 6	<u>7.42</u> 17	11.9 11	16.8 13	<u>6.31</u> 30	11.9 27	11.8 23
<u>EnhancedBP [24]</u>	16.6	<u>0.94</u> 3	1.74 10	5.05 3	<u>0.35</u> 15	0.86 17	4.34 17	<u>8.11</u> 22	13.3 19	18.5 22	<u>5.09</u> 27	11.1 23	11.0 21

Volumetric multiview approaches

- Goal: find a model consistent with images
- “Model-centric” (vs. image-centric)
- Typically use discretized volume (voxel grid)
- For each voxel, compute occupied / free (for some algorithms, also color, etc.)

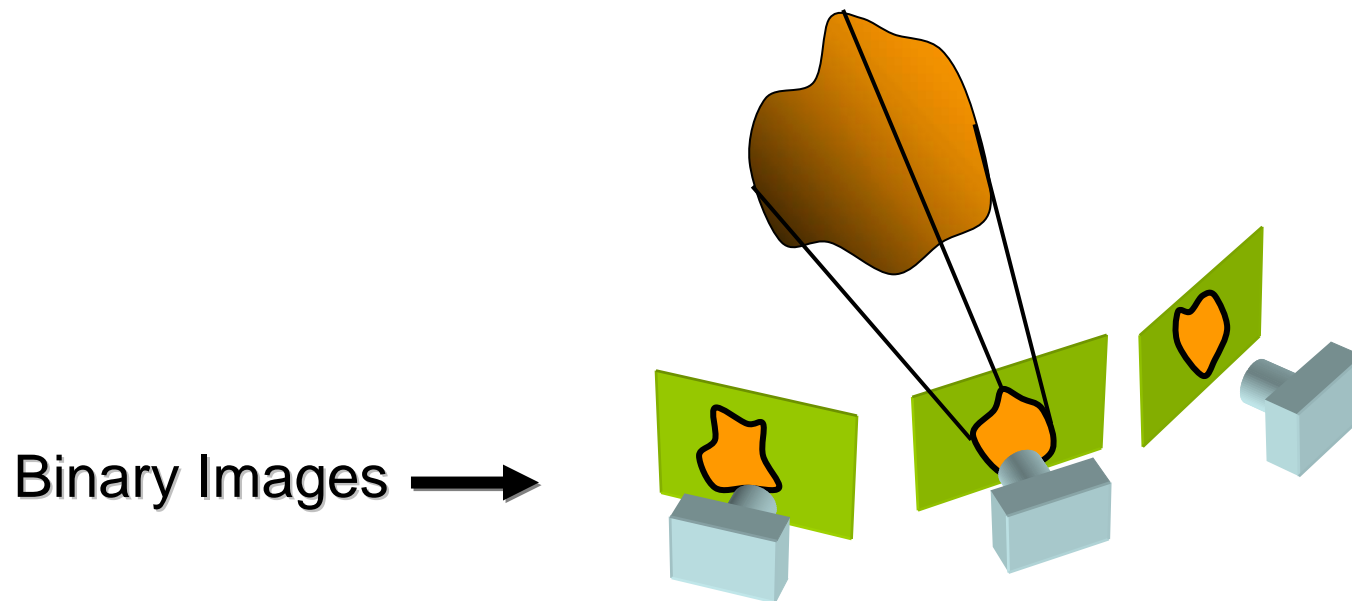
Photo consistency

- Result: not necessarily the correct scene
- Many scenes produce the same images



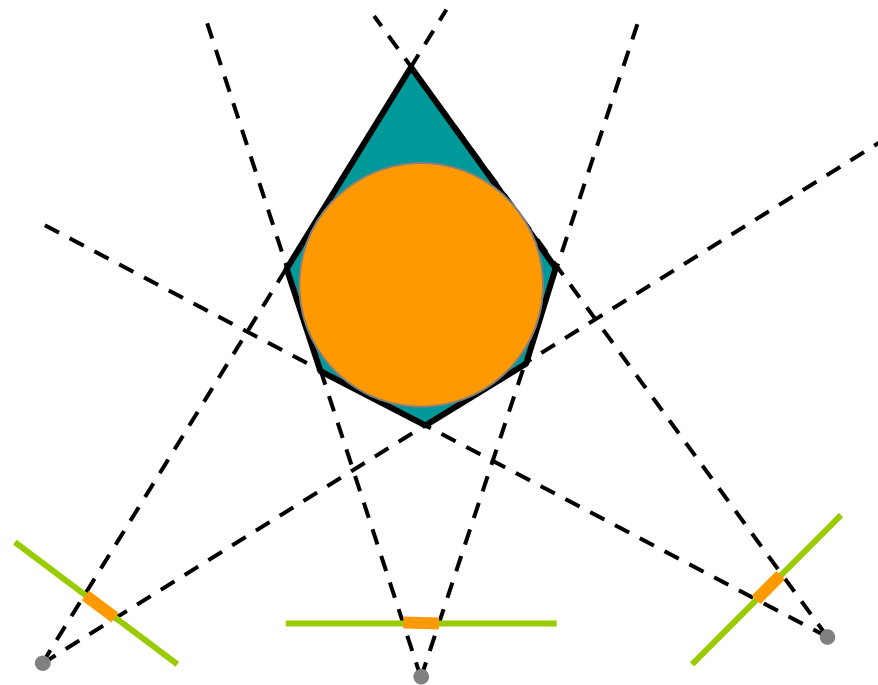
Silhouette carving

- Find silhouettes in all images
- Exact version:
 - Back-project all silhouettes, find intersection



Silhouette carving

- Find silhouettes in all images
- Exact version:
 - Back-project all silhouettes, find intersection



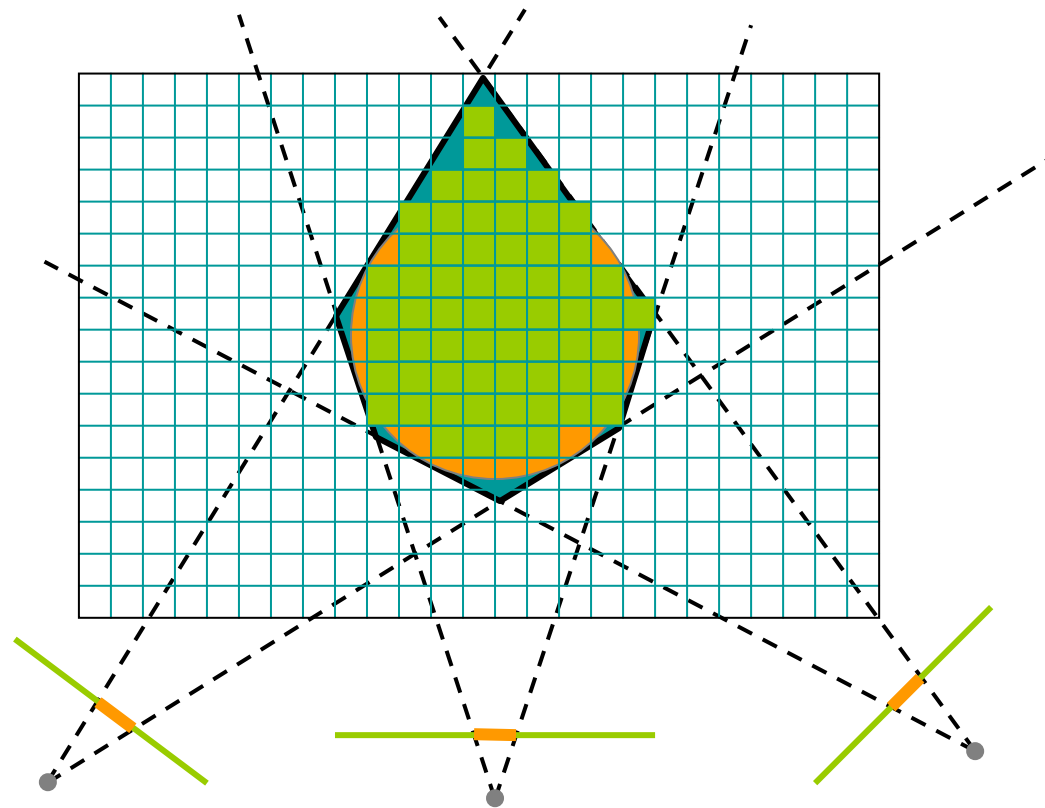
Silhouette carving

- Limit of silhouette carving is *visual hull* or *line hull*
- Complement of lines that don't intersect object
- In general not the same as object
 - Can't recover "pits" in object
- Not the same as convex hull

Silhouette carving

- Discrete version:
 - Loop over all voxels in some volume
 - If projection into images lies inside all silhouettes, mark as occupied
 - Else mark as free

Silhouette carving



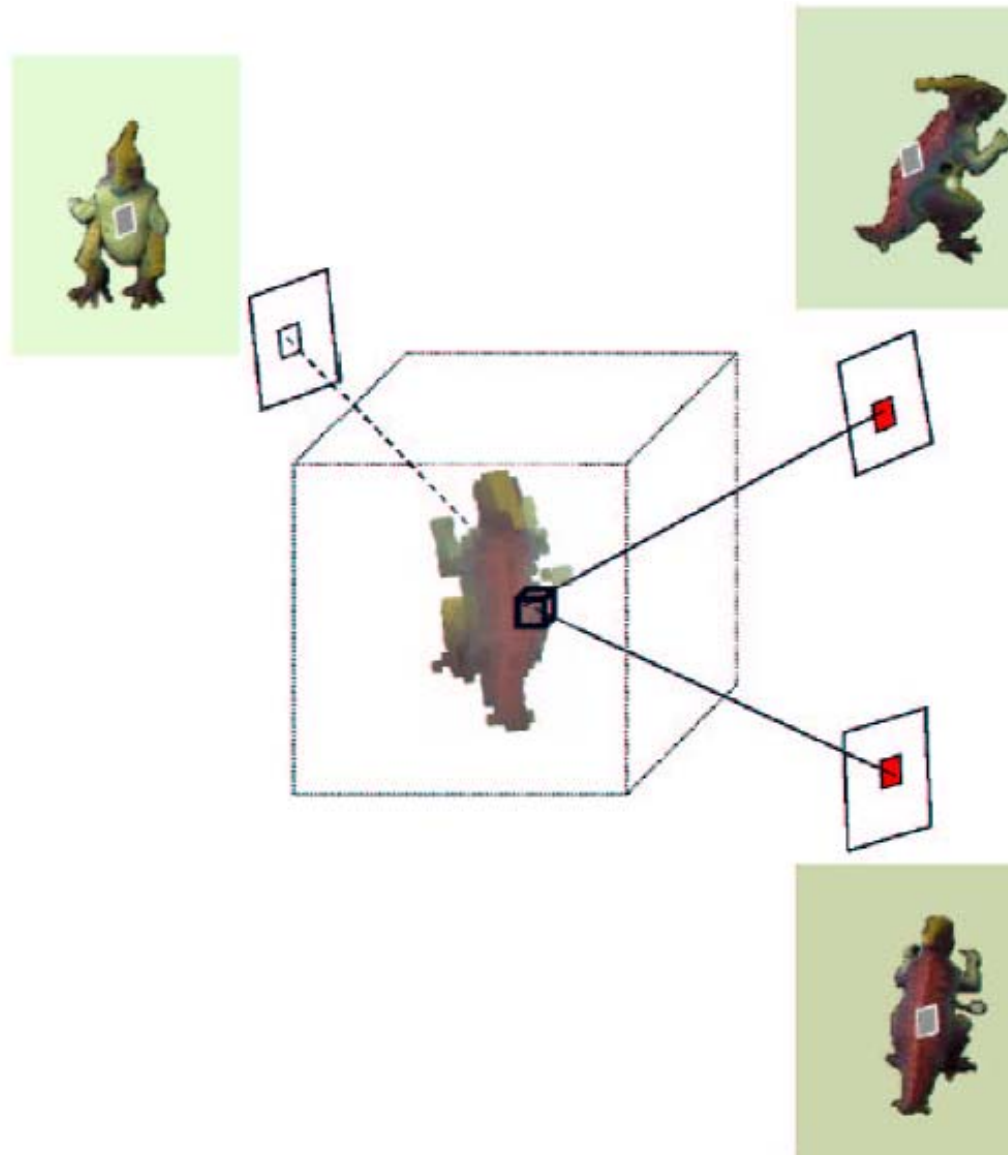
Voxel coloring

- Seitz and Dyer, 1997
- In addition to free / occupied, store color at each voxel
- Explicitly accounts for occlusion

Voxel coloring

- Basic idea: sweep through a voxel grid
 - Project each voxel into each image in which it is visible
 - If colors in images agree, mark voxel with color
 - Else, mark voxel as empty
- Agreement of colors based on comparing standard deviation of colors to threshold

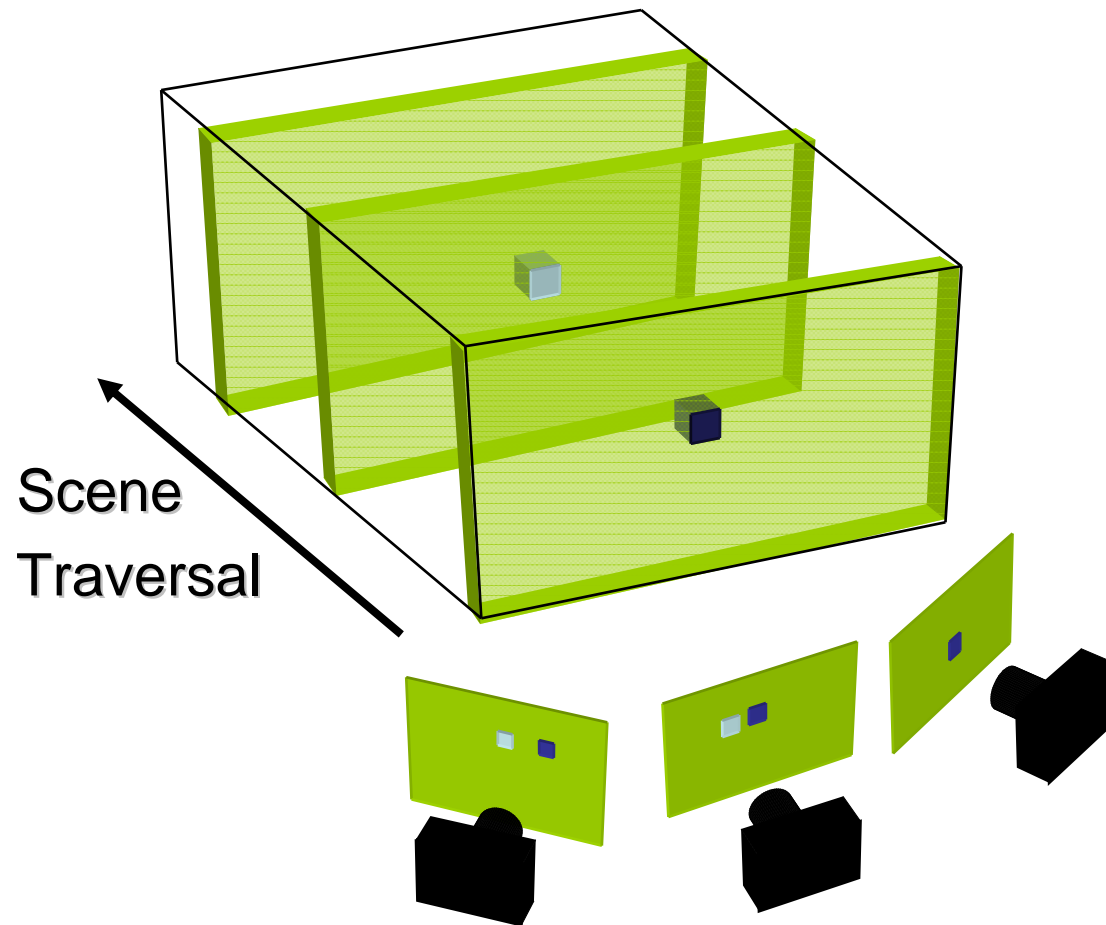
Voxel coloring and occlusion



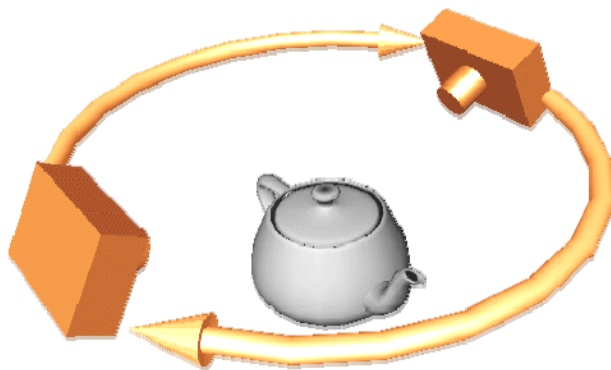
Voxel coloring and occlusion

- Problem: which voxels are visible?
- Solution: constrain camera views
 - When a voxel is considered, necessary occlusion information must be available
 - Sweep occluders before occludees
 - Constrain camera positions to allow this sweep

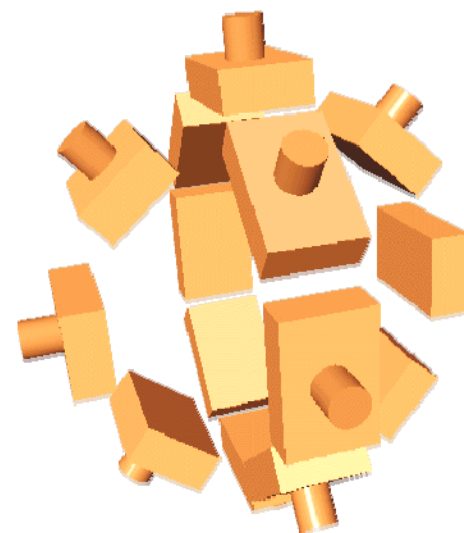
Voxel coloring sweep order



Voxel coloring camera positions

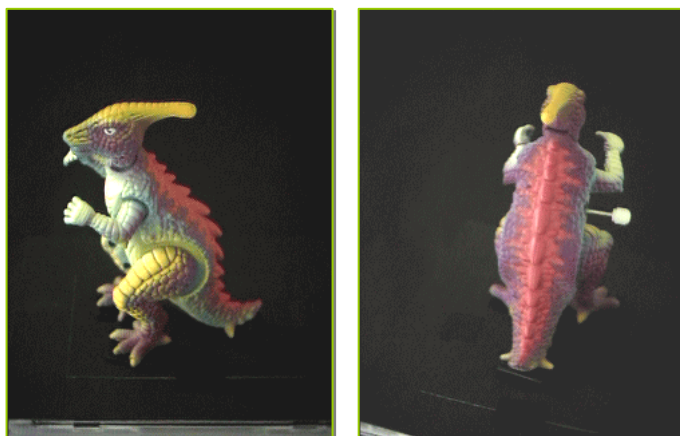


Inward-looking
Cameras above scene



Outward-looking
Cameras inside scene

Image acquisition



Selected Dinosaur Images



Selected Flower Images



- Calibrated Turntable
- 360° rotation (21 images)

Voxel coloring results



Dinosaur Reconstruction

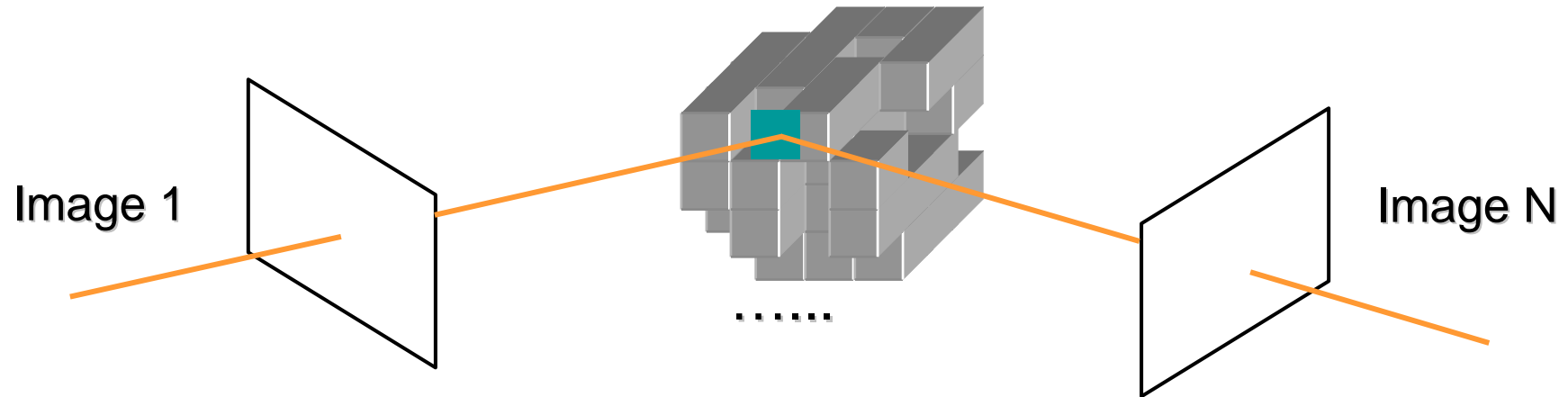
72 K voxels colored
7.6 M voxels tested
7 min. to compute
on a 250MHz SGI



Flower Reconstruction

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

Space carving

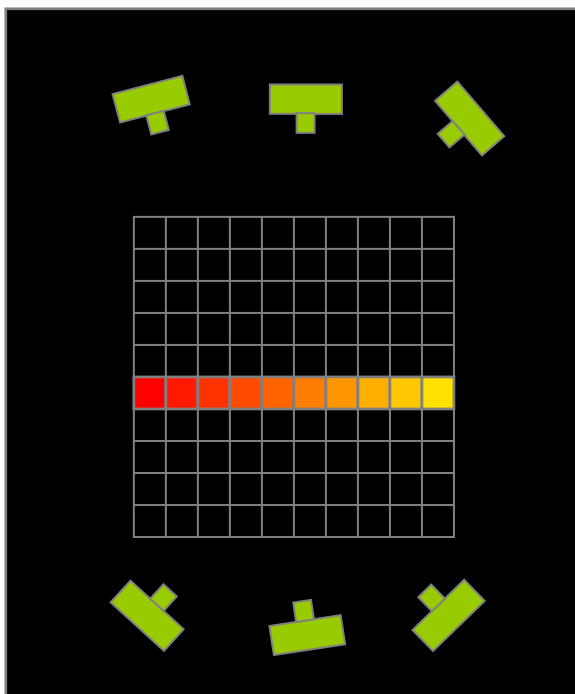


Initialize to a volume V containing the true scene
Choose a voxel on the current surface
Project to visible input images
Carve if not photo-consistent
Repeat until convergence

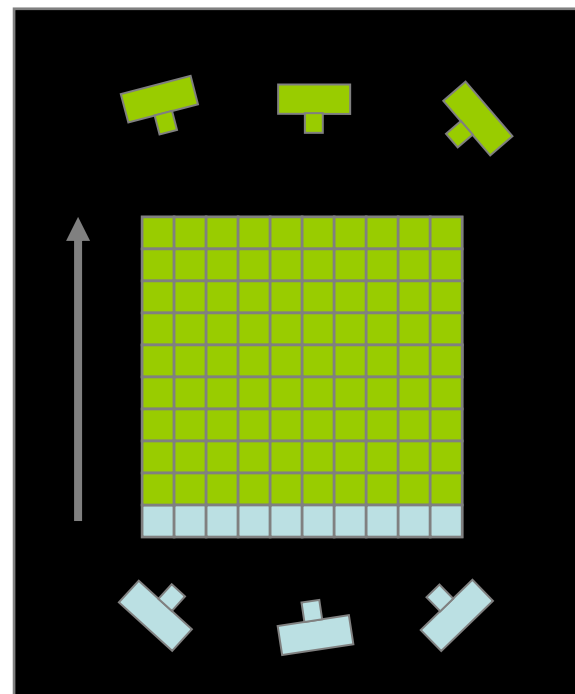
Multi-pass plane sweep

- Faster alternative:
 - Sweep plane in each of 6 principal directions
 - Consider cameras on only one side of plane
 - Repeat until convergence

Multi-pass plane sweep

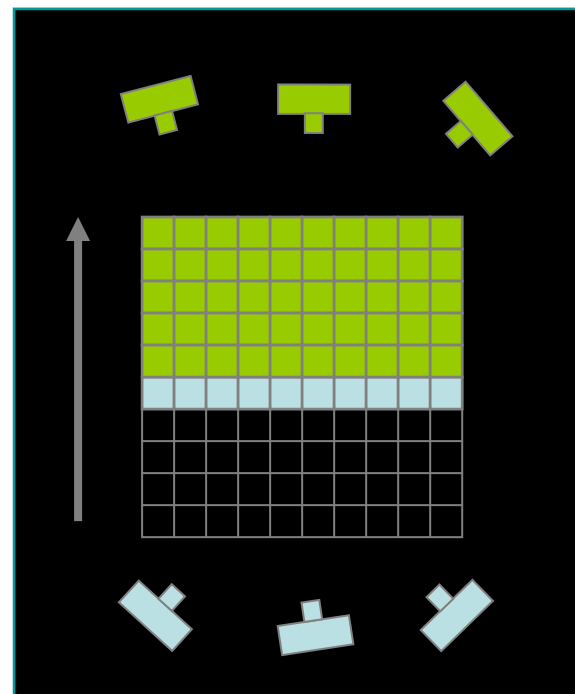
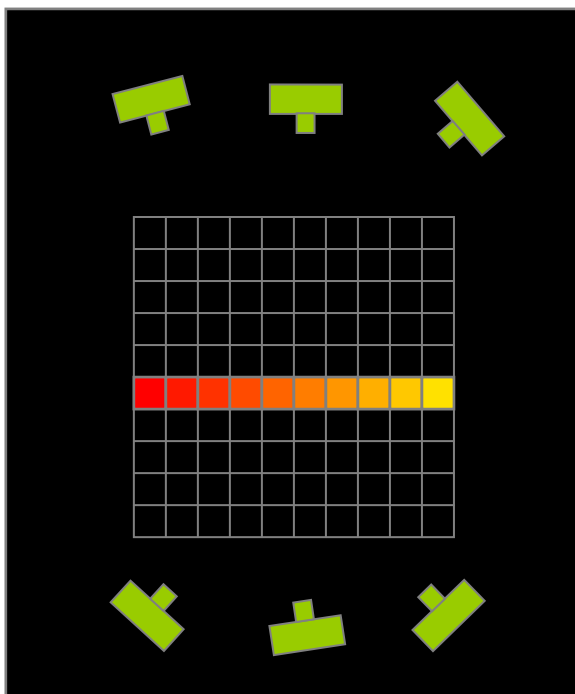


True Scene

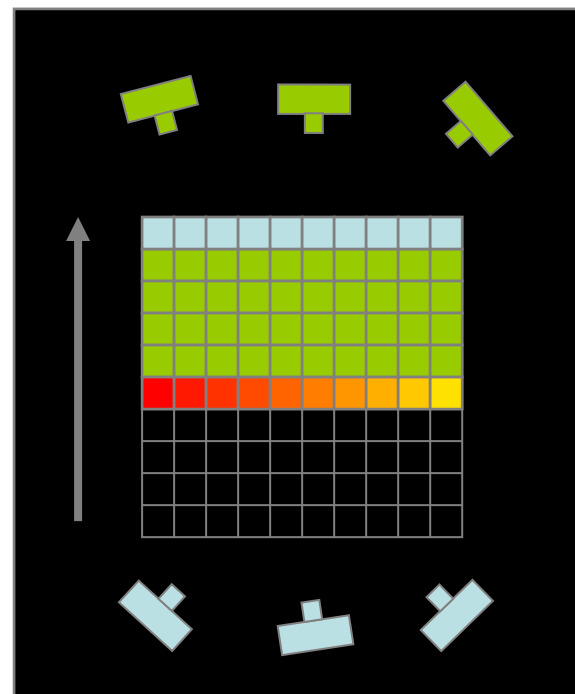
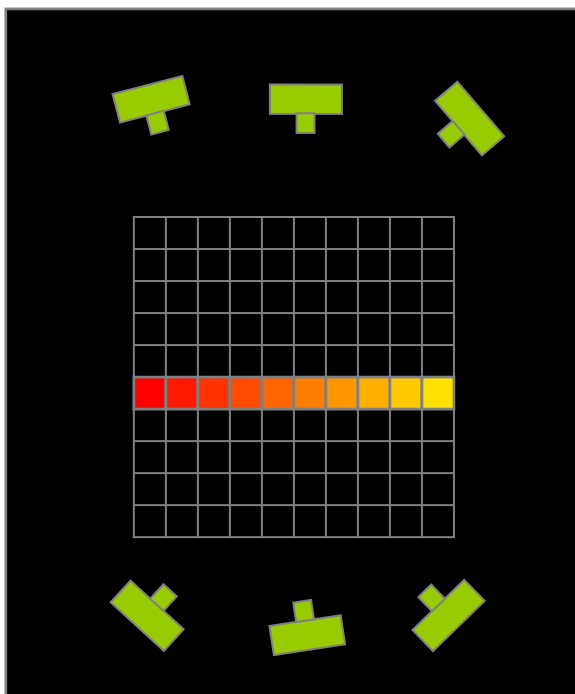


Reconstruction

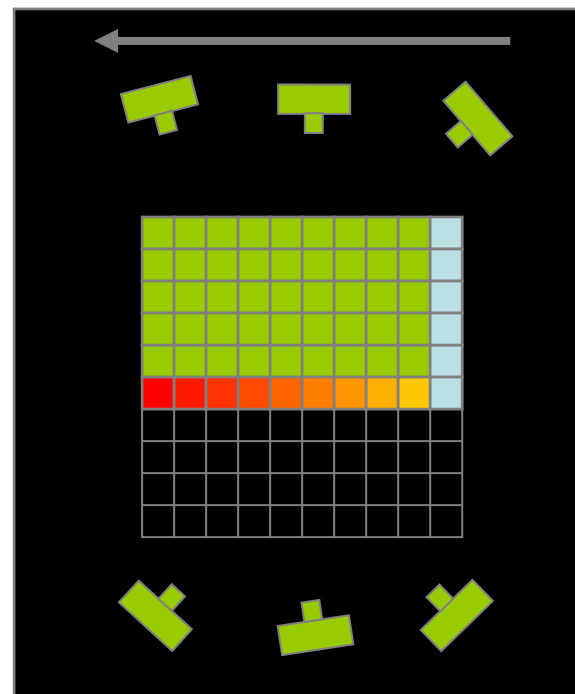
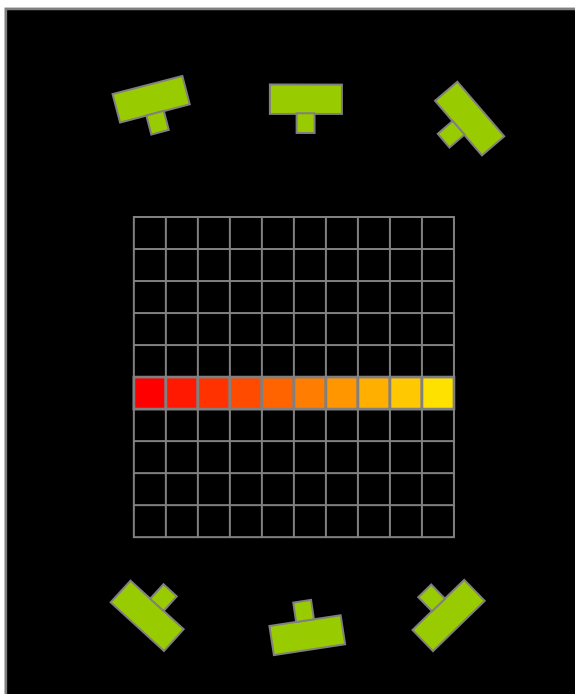
Multi-pass plane sweep



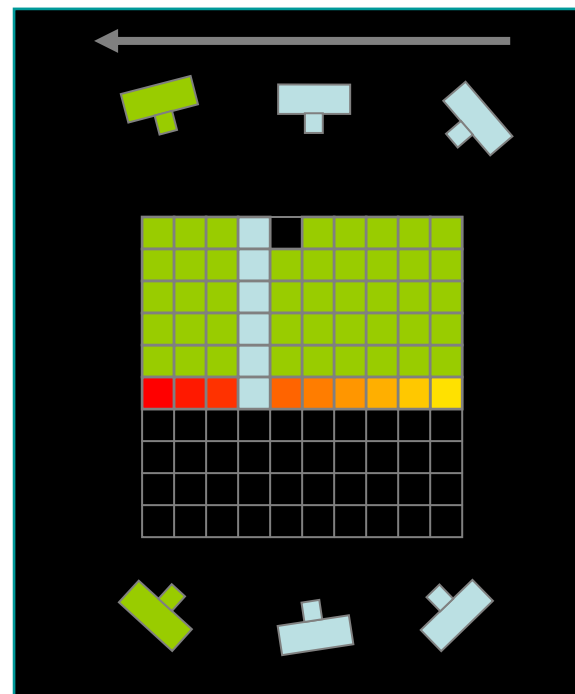
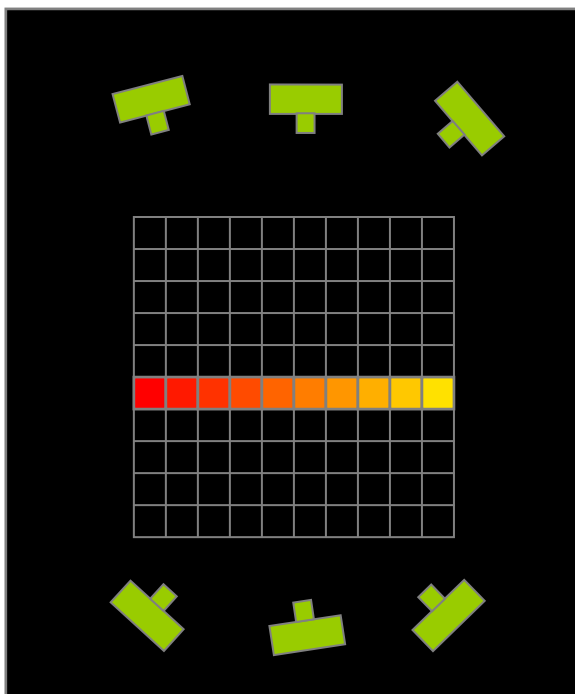
Multi-pass plane sweep



Multi-pass plane sweep



Multi-pass plane sweep



Space carving results: African violet



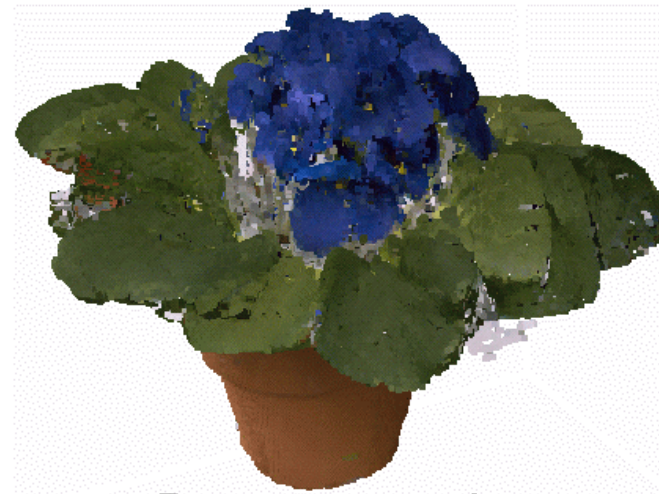
Input image (1 of 45)



Reconstruction



Reconstruction



Reconstruction

Space carving results: hand



Input image
(1 of 100)



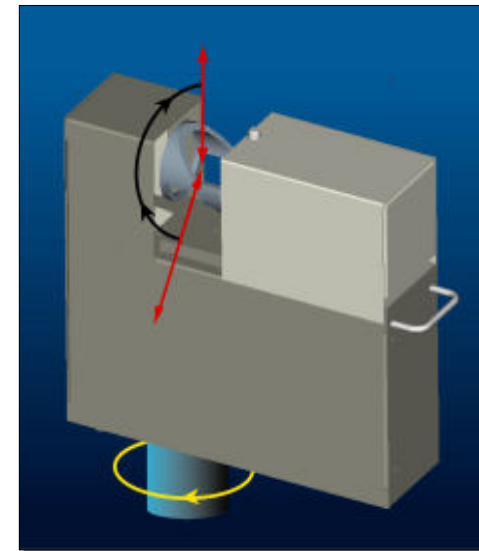
Reconstruction

Active approaches

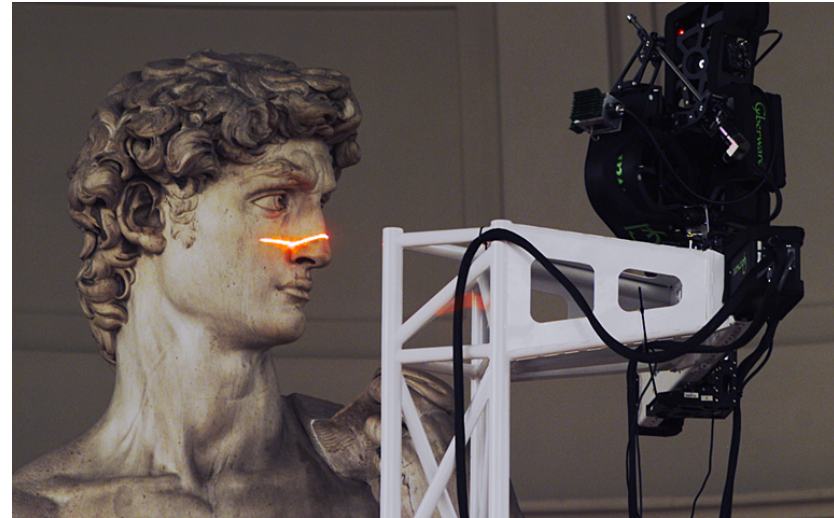
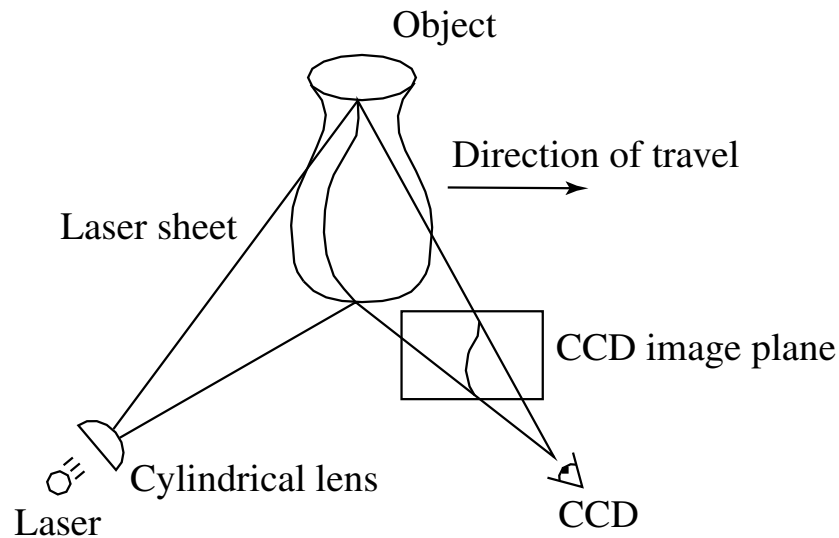
Time of flight

- Basic idea: send out pulse of light (usually laser), time how long it takes to return

$$r = \frac{1}{2} c \Delta t$$



Laser scanning (triangulation)



Digital Michelangelo Project
<http://graphics.stanford.edu/projects/mich/>

- Optical triangulation
 - Project a single stripe of laser light
 - Scan it across the surface of the object
 - This is a very precise version of structured light scanning
- Other patterns are possible

Cyberware



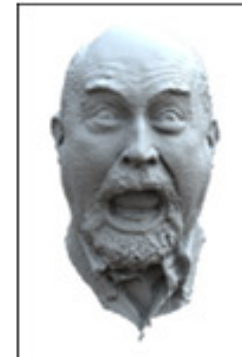
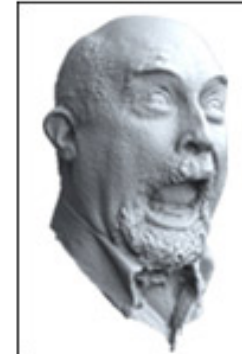
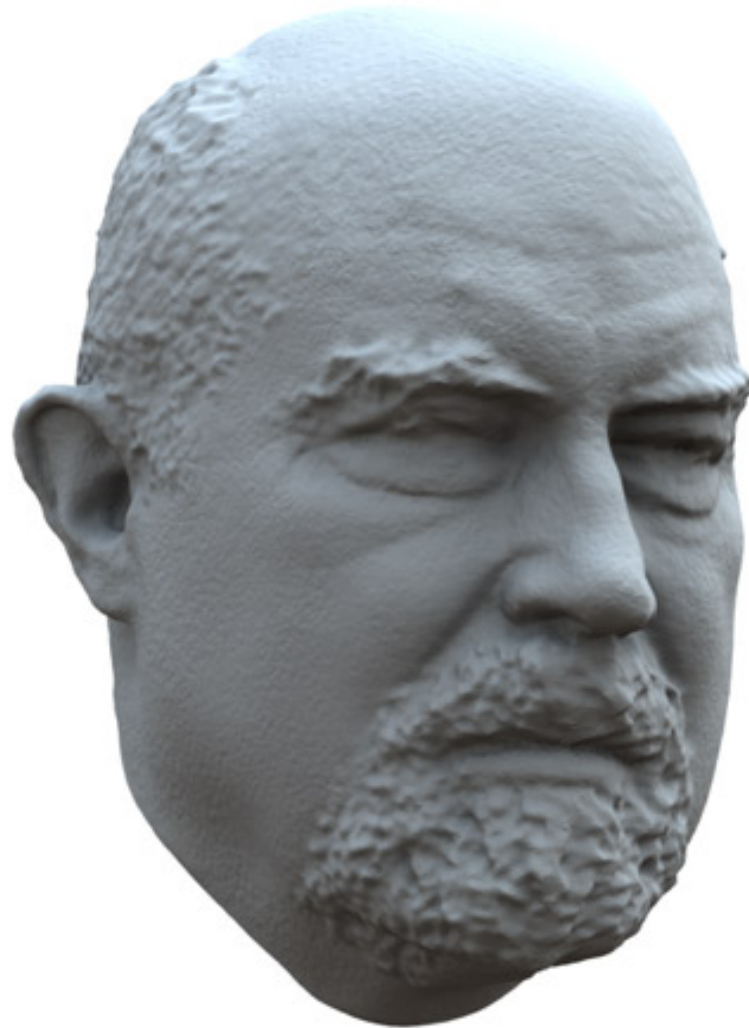
face and hand



full body

XYZRGB

Data acquired in 0.01 seconds



© 2007 XYZ RGB Inc.

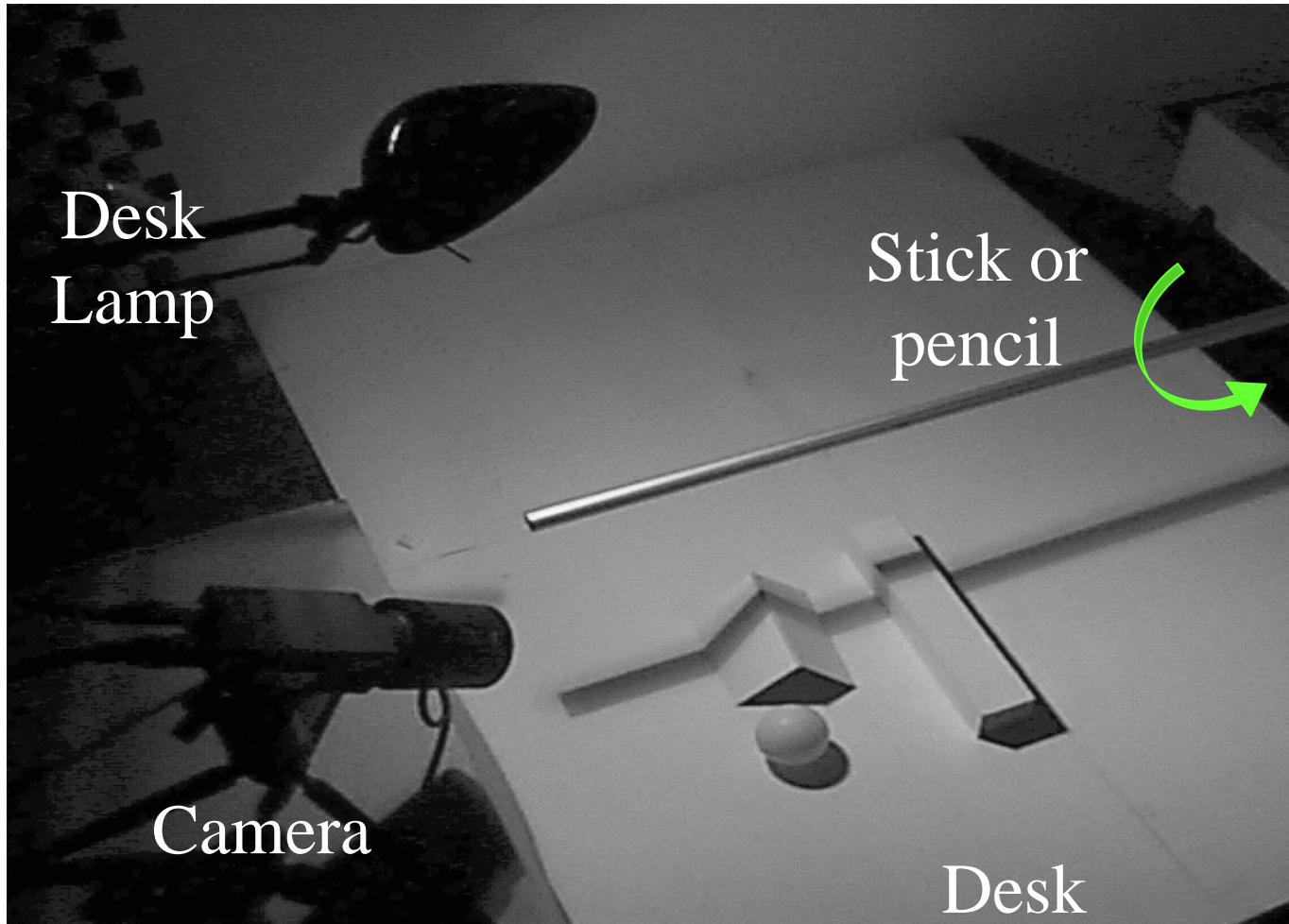
XYZRGB

Live Body Scan
Data acquired in 0.01 seconds



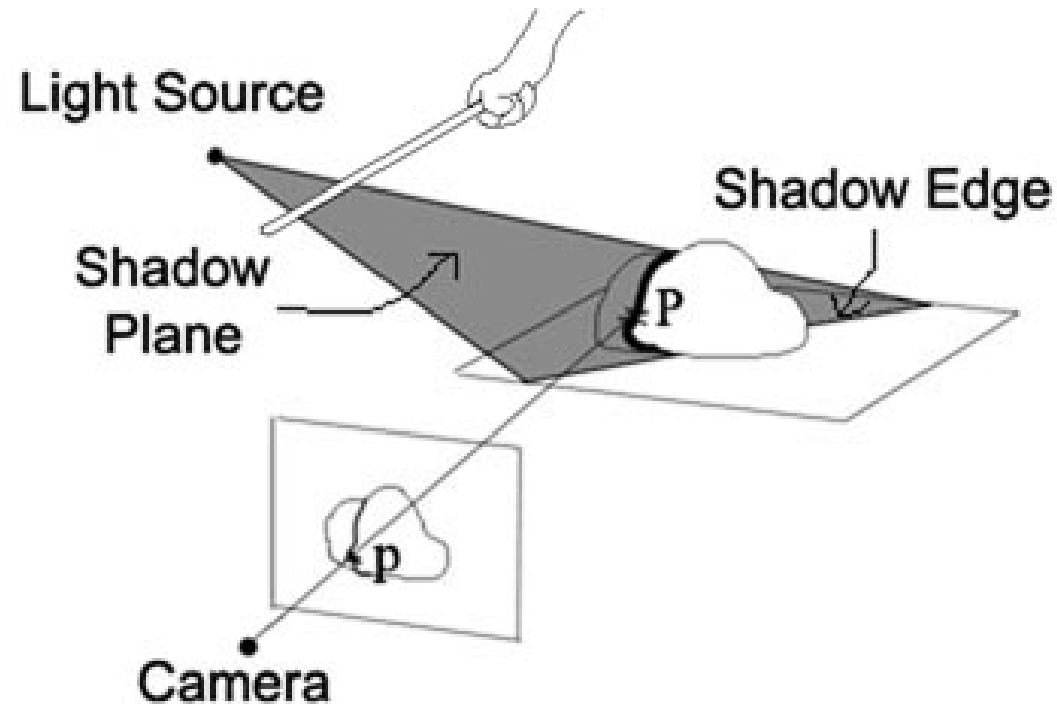
© 2007 XYZ RGB Inc.

Shadow scanning



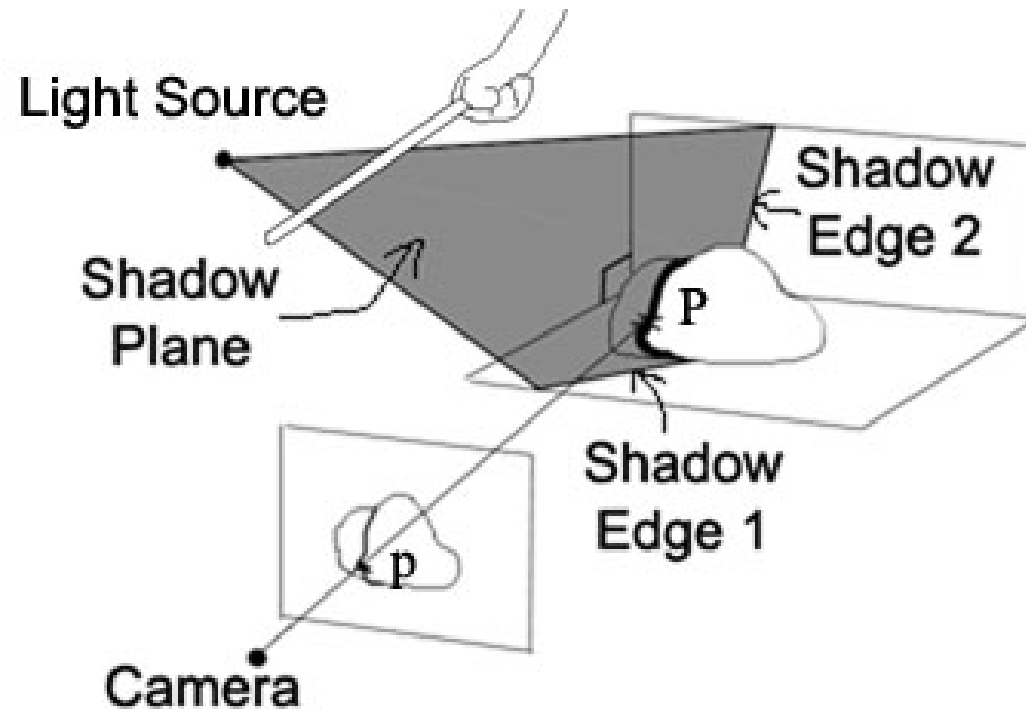
<http://www.vision.caltech.edu/bouguetj/ICCV98/>

Basic idea



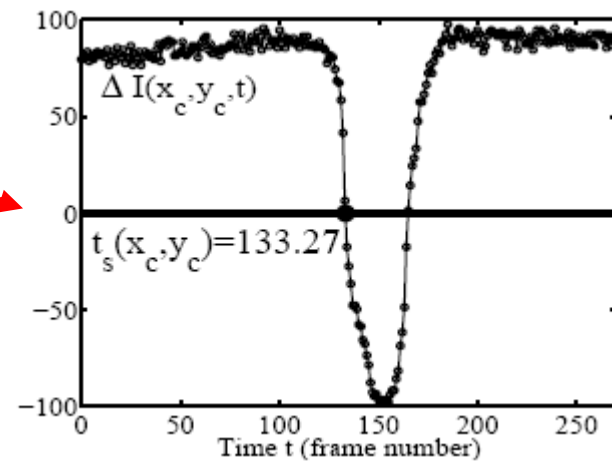
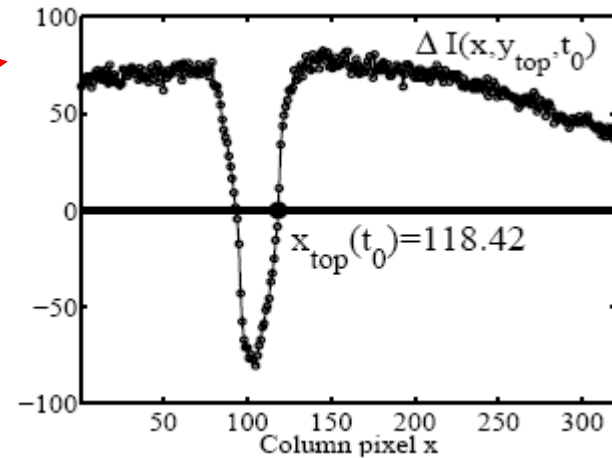
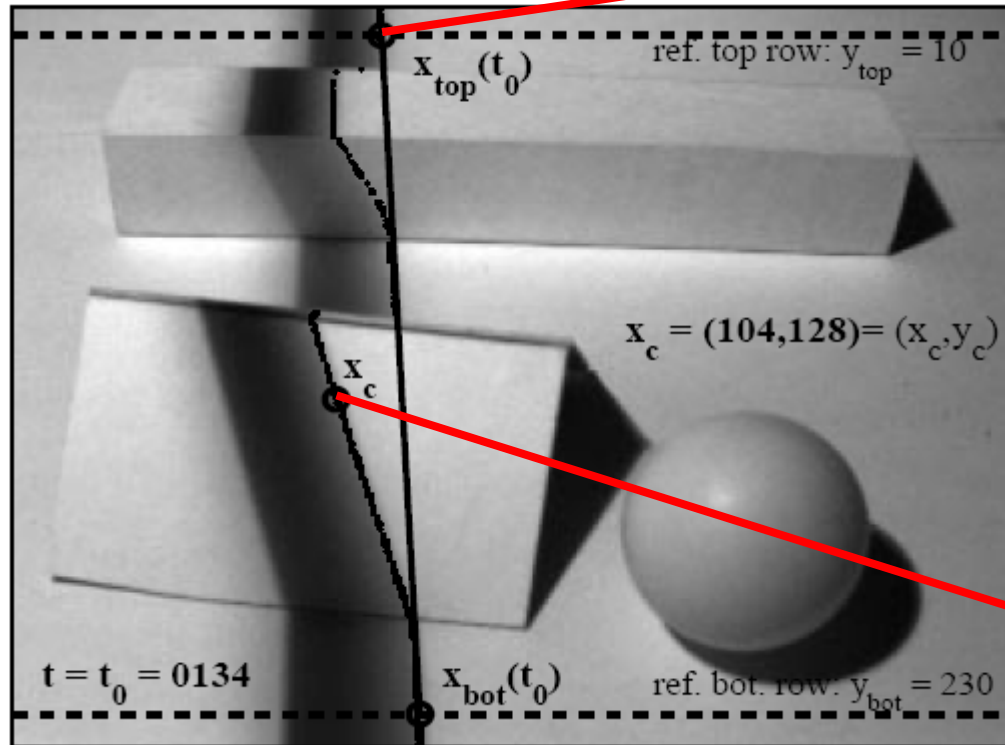
- Calibration issues:
 - where's the camera wrt. ground plane?
 - where's the shadow plane?
 - depends on light source position, shadow edge

Two Plane Version



- Advantages
 - don't need to pre-calibrate the light source
 - shadow plane determined from two shadow edges

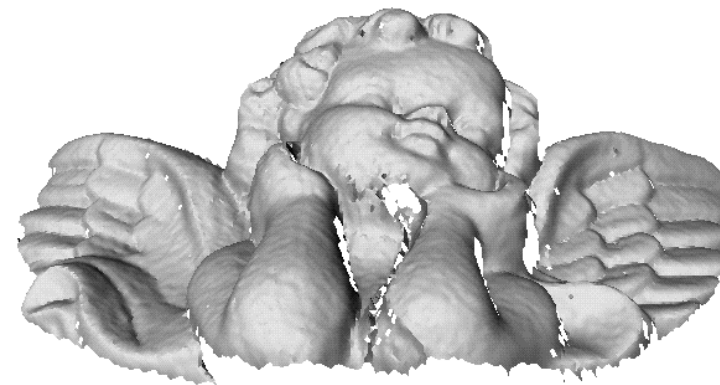
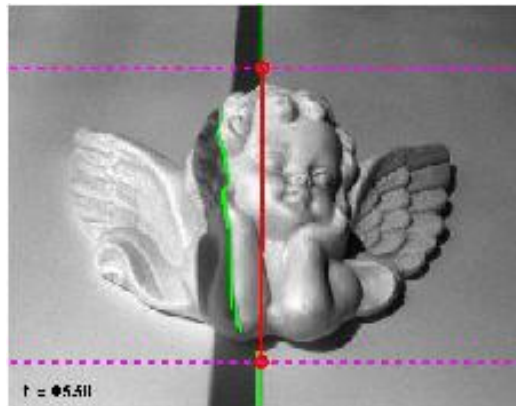
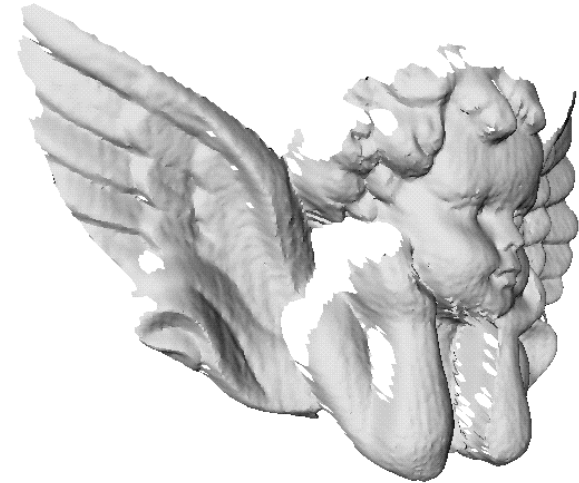
Estimating shadow lines



Shadow scanning in action

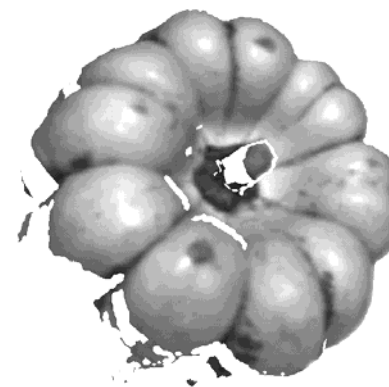
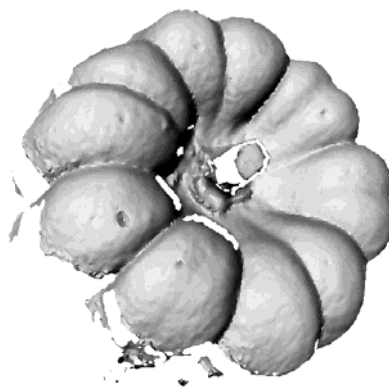


Results

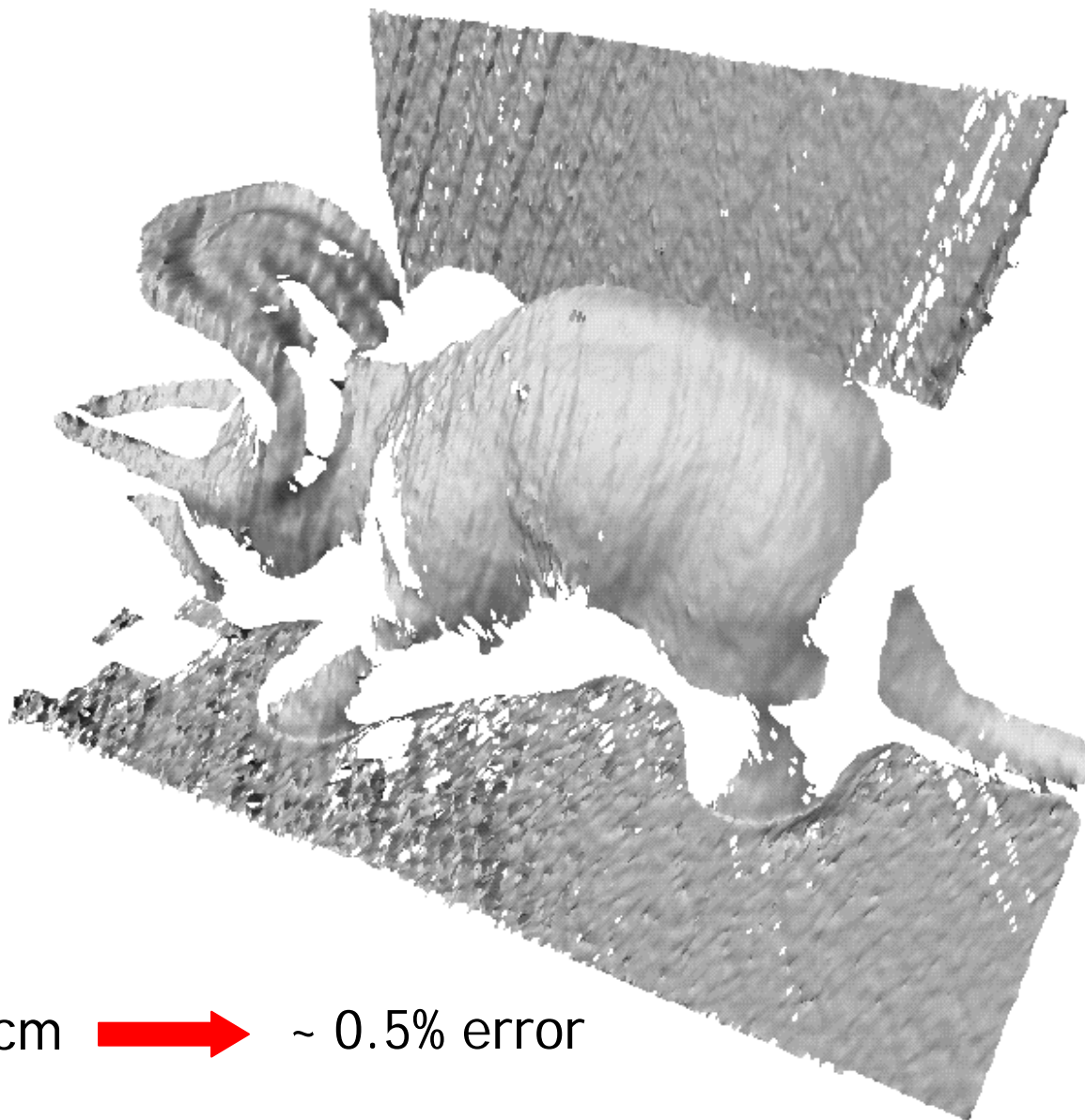
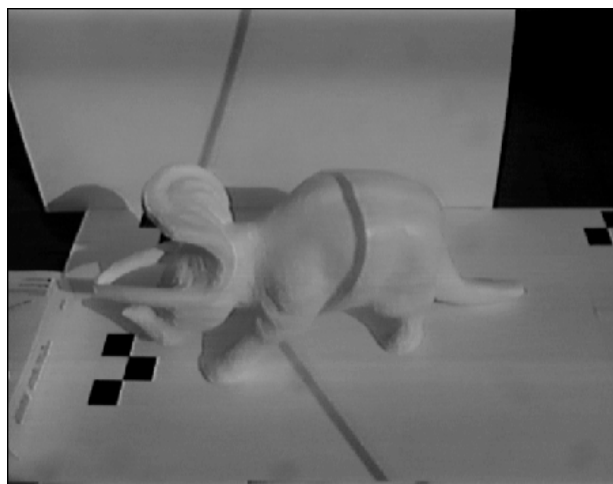
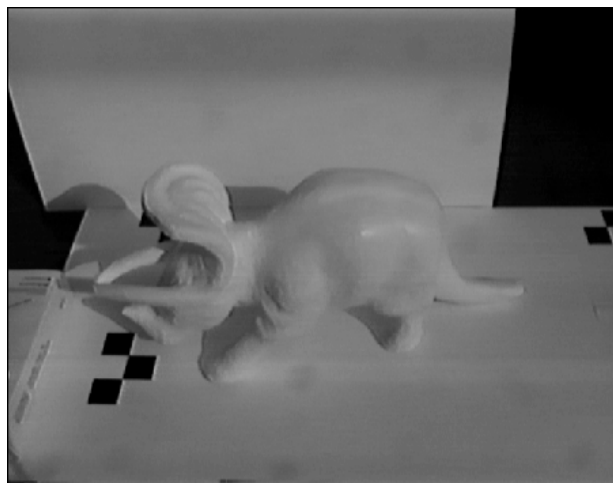


accuracy: 0.1mm over 10cm ➔ ~ 0.1% error

Textured objects



Scanning with the sun



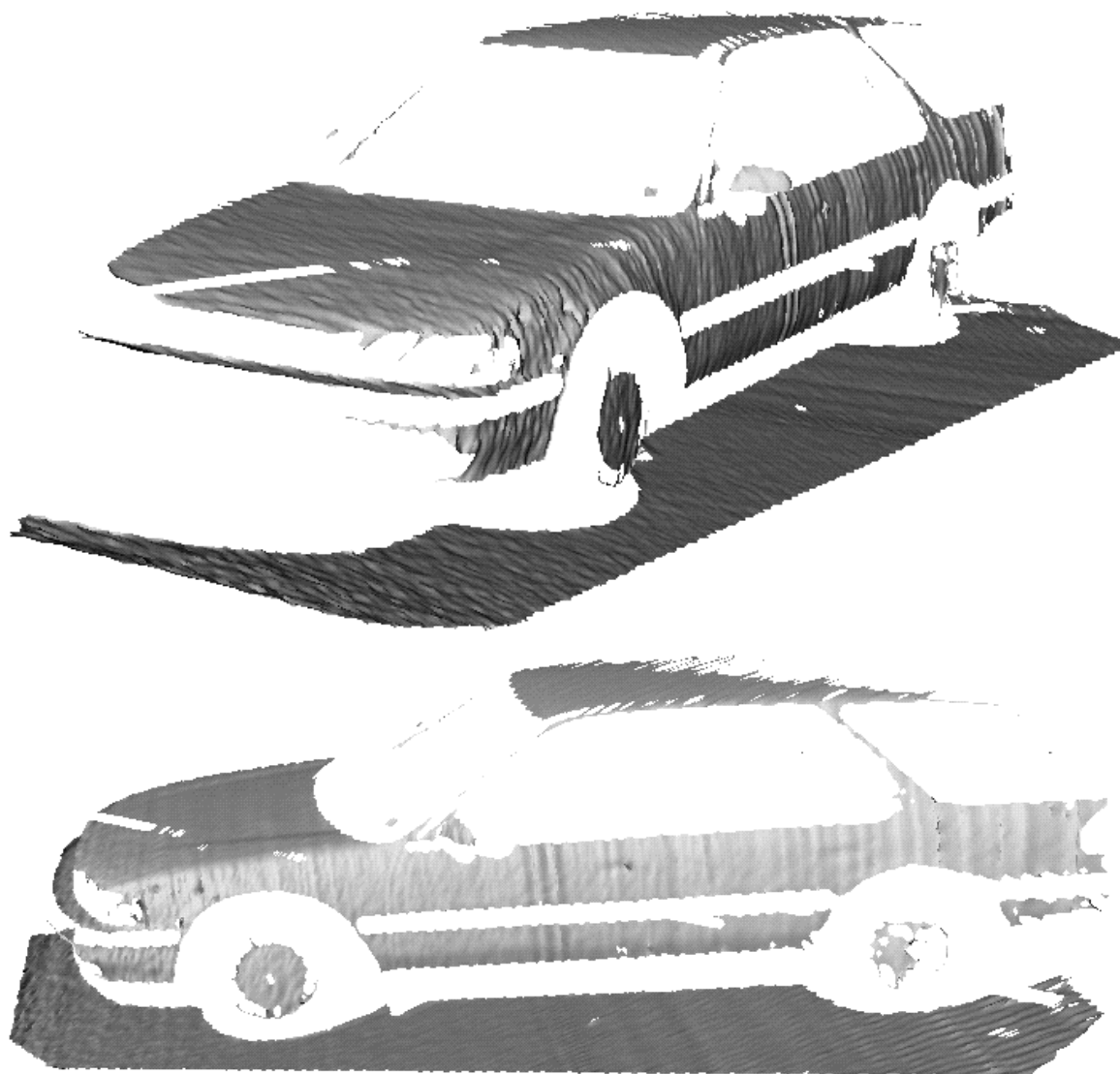
accuracy: 1mm over 50cm  ~ 0.5% error

Scanning with the sun



accuracy: 1cm over 2m

→ ~ 0.5% error



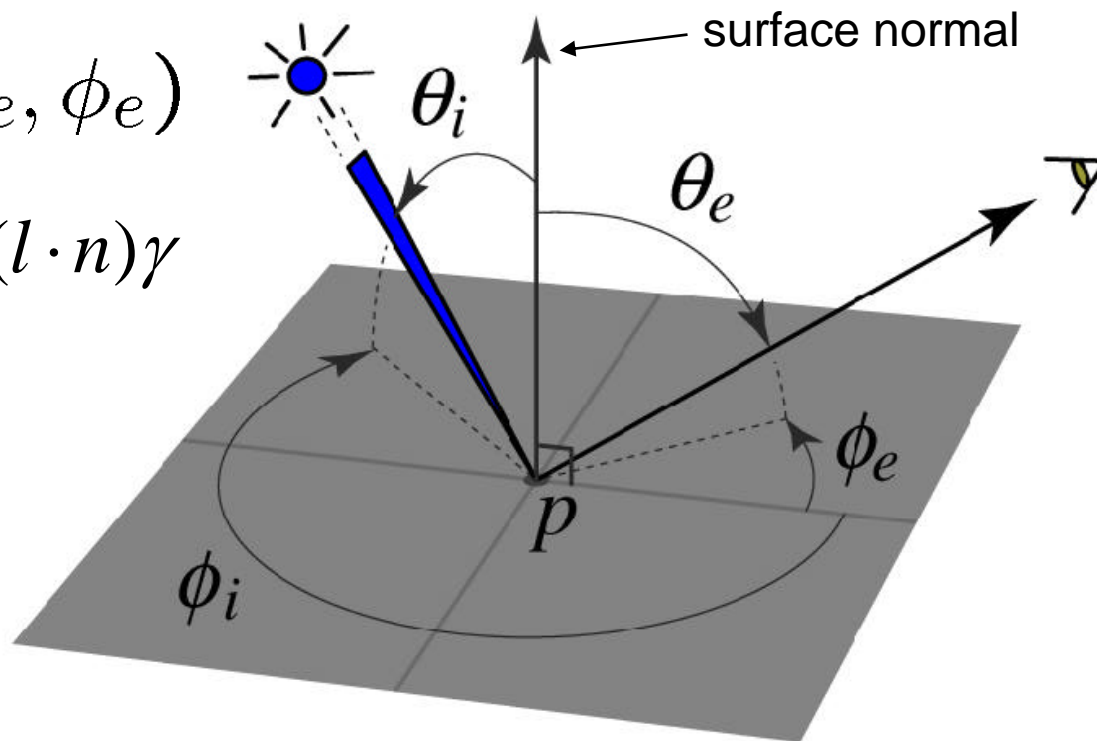
**Active variants of
passive approaches**

The BRDF

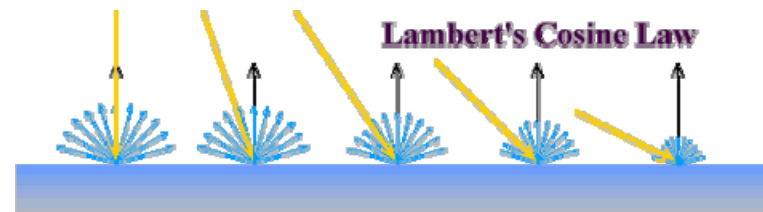
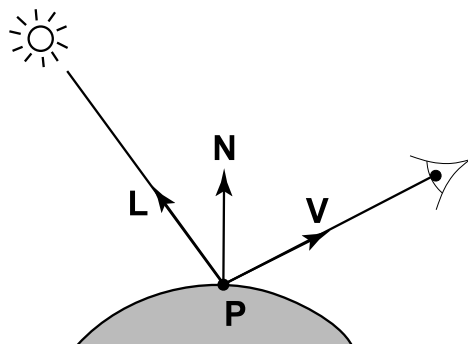
- The Bidirectional Reflection Distribution Function
 - Given an incoming ray (θ_i, ϕ_i) and outgoing ray (θ_e, ϕ_e) what proportion of the incoming light is reflected along outgoing ray?

$$\rho(\theta_i, \phi_i, \theta_e, \phi_e)$$

$$I = \rho(l, v) \times (l \cdot n) \gamma$$



Diffuse reflection (Lambertian)

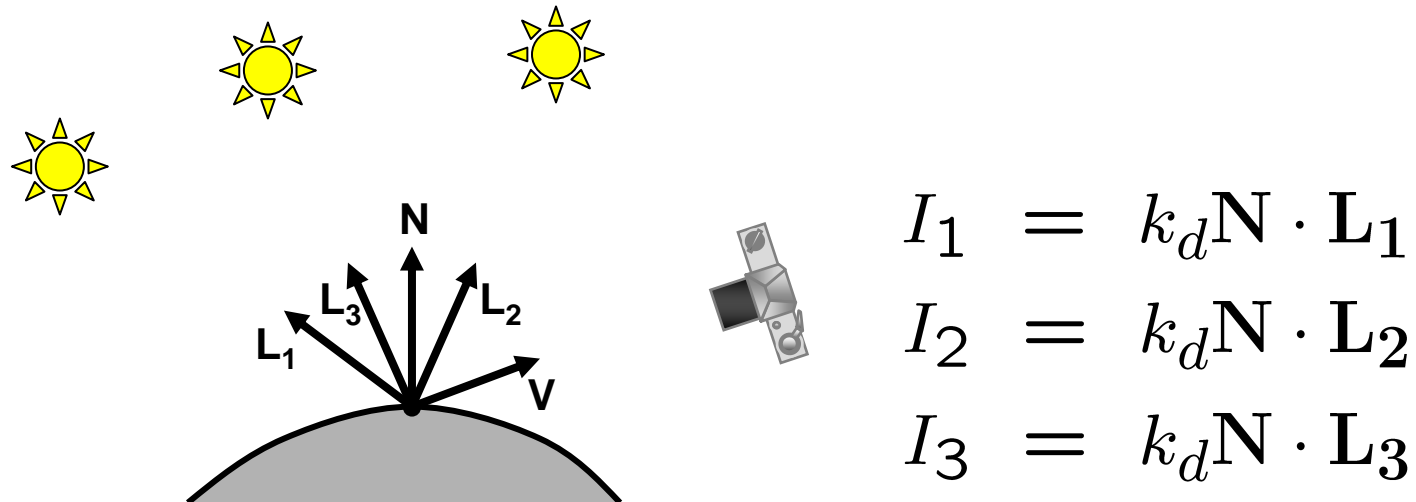


$$\rho(l, v) = k_d \leftarrow \text{albedo}$$

$$I = k_d \mathbf{N} \cdot \mathbf{L}$$

Assuming that light strength is 1.

Photometric stereo



- Can write this as a matrix equation:

$$\begin{bmatrix} I_1 \\ I_2 \\ I_3 \end{bmatrix} = k_d \begin{bmatrix} \mathbf{L}_1^T \\ \mathbf{L}_2^T \\ \mathbf{L}_3^T \end{bmatrix} \mathbf{N}$$

Solving the equations

$$\underbrace{\begin{bmatrix} I_1 \\ I_2 \\ I_3 \end{bmatrix}}_{\mathbf{I}} = \underbrace{\begin{bmatrix} \mathbf{L}_1^T \\ \mathbf{L}_2^T \\ \mathbf{L}_3^T \end{bmatrix}}_{\mathbf{L}} \underbrace{k_d \mathbf{N}}_{\mathbf{G}}$$

3×1 3×3 3×1

$$\mathbf{G} = \mathbf{L}^{-1} \mathbf{I}$$

$$k_d = \|\mathbf{G}\|$$

$$\mathbf{N} = \frac{1}{k_d} \mathbf{G}$$

More than three lights

- Get better results by using more lights

$$\begin{bmatrix} I_1 \\ \vdots \\ I_n \end{bmatrix} = \begin{bmatrix} \mathbf{L}_1 \\ \vdots \\ \mathbf{L}_n \end{bmatrix} k_d \mathbf{N}$$

- Least squares solution:

$$\begin{aligned} \mathbf{I} &= \mathbf{L}\mathbf{G} \\ \mathbf{L}^T \mathbf{I} &= \mathbf{L}^T \mathbf{L}\mathbf{G} \\ \mathbf{G} &= (\mathbf{L}^T \mathbf{L})^{-1} (\mathbf{L}^T \mathbf{I}) \end{aligned}$$

- Solve for \mathbf{N} , k_d as before

Trick for handling shadows

- Weight each equation by the pixel brightness:

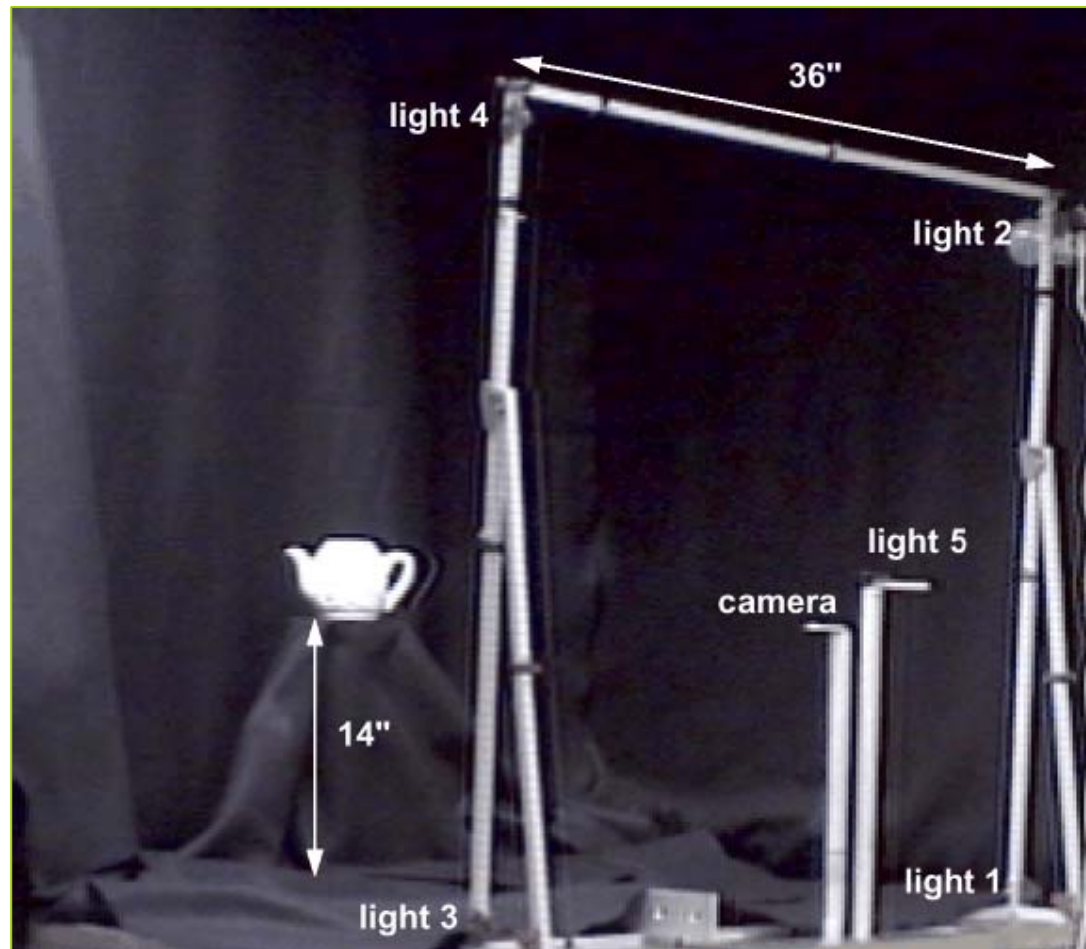
$$I_i(I_i) = I_i[k_d \mathbf{N} \cdot \mathbf{L}_i]$$

- Gives weighted least-squares matrix equation:

$$\begin{bmatrix} I_1^2 \\ \vdots \\ I_n^2 \end{bmatrix} = \begin{bmatrix} I_1 \mathbf{L}_1^T \\ \vdots \\ I_n \mathbf{L}_n^T \end{bmatrix} k_d \mathbf{N}$$

- Solve for \mathbf{N} , k_d as before

Photometric Stereo Setup

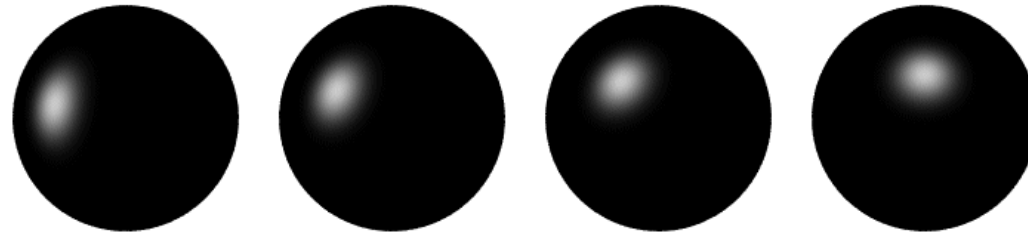


Procedure

- Calibrate camera
- Calibrate light directions/intensities
- Photographing objects (HDR recommended)
- Estimate normals
- Estimate depth

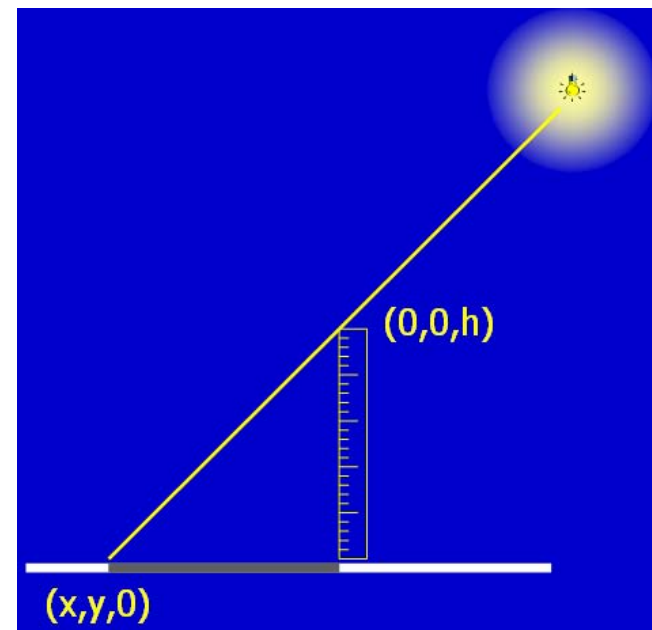
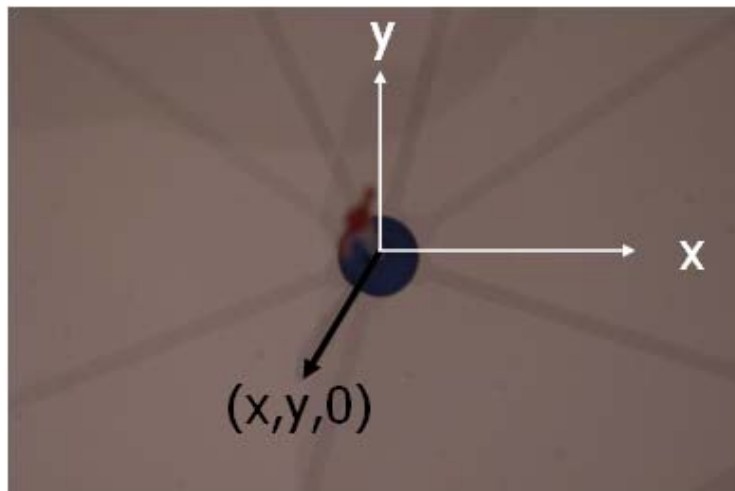
Estimating light directions

- Trick: place a chrome sphere in the scene



- the location of the highlight tells you where the light source is

- Use a ruler



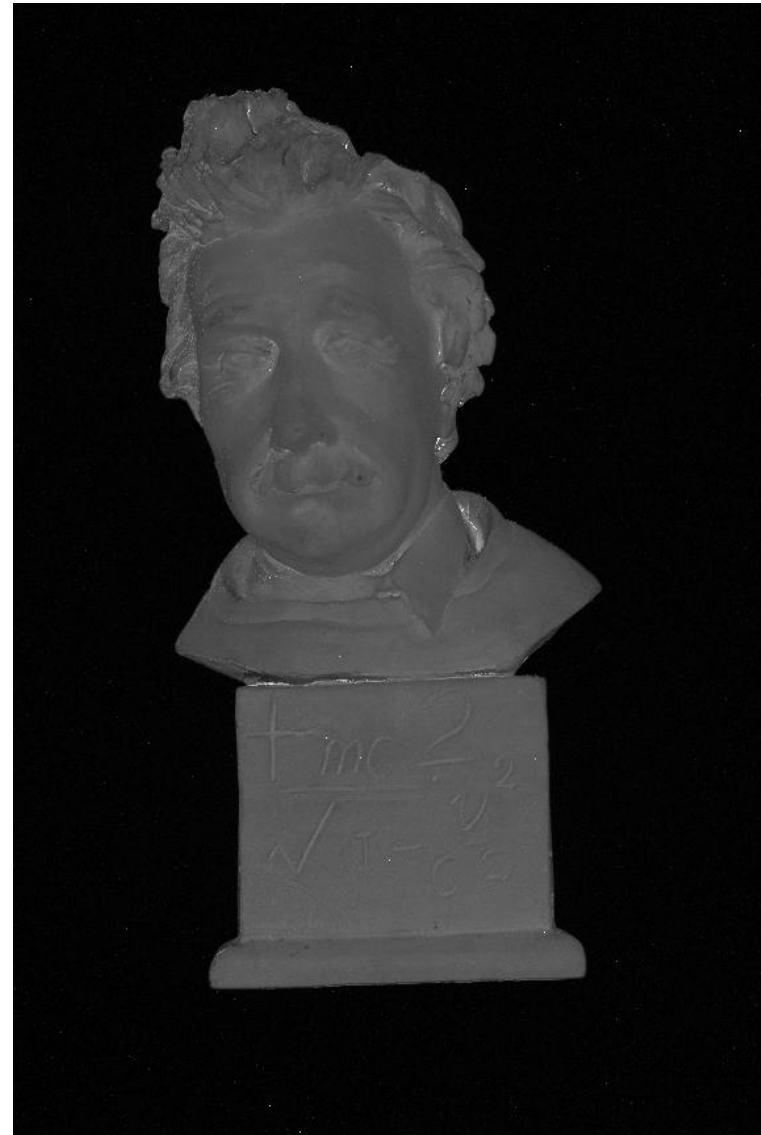
Photographing objects



Normalize light intensities



Estimate normals



Depth from normals

$$(n_x, n_y, n_z) = \left(\frac{\partial z}{\partial x}, \frac{\partial z}{\partial y}, -1 \right) = (p, q, -1)$$

$$E = E_{data} + E_{smooth} + E_{cons}$$

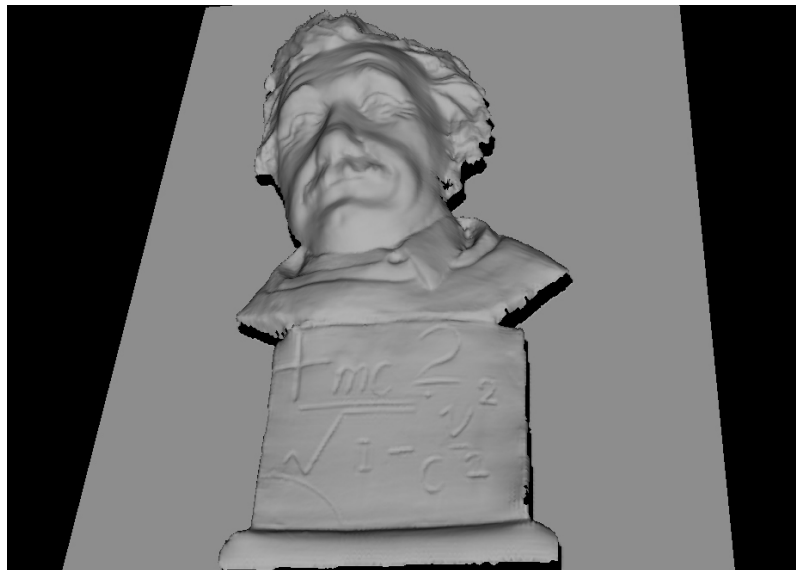
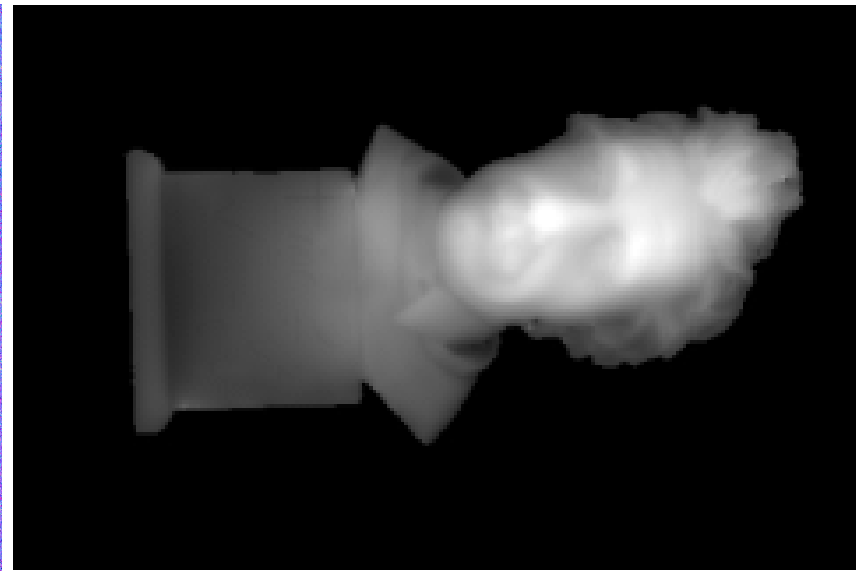
$$= \sum_{i,j} w_{data} * \left[\left(\frac{\partial z(i,j)}{\partial x} - p_{ij} \right)^2 + \left(\frac{\partial z(i,j)}{\partial y} - q_{ij} \right)^2 \right]$$

$$+ \sum_{i,j} w_{smooth} * \left[\left(\frac{\partial^2 z(i,j)}{\partial x^2} \right)^2 + 2 \left(\frac{\partial^2 z(i,j)}{\partial x \partial y} \right)^2 + \left(\frac{\partial^2 z(i,j)}{\partial y^2} \right)^2 \right]$$

$$+ \sum_{(i,j) \in C_{cons}} w_{cons} * (z(i,j) - c_{ij})^2$$

$$E = \frac{1}{2} \mathbf{z}^T \mathbf{A} \mathbf{z} - \mathbf{b}^T \mathbf{z} + c \quad \equiv \quad \mathbf{A} \mathbf{z} = \mathbf{b}$$

Results



Limitations

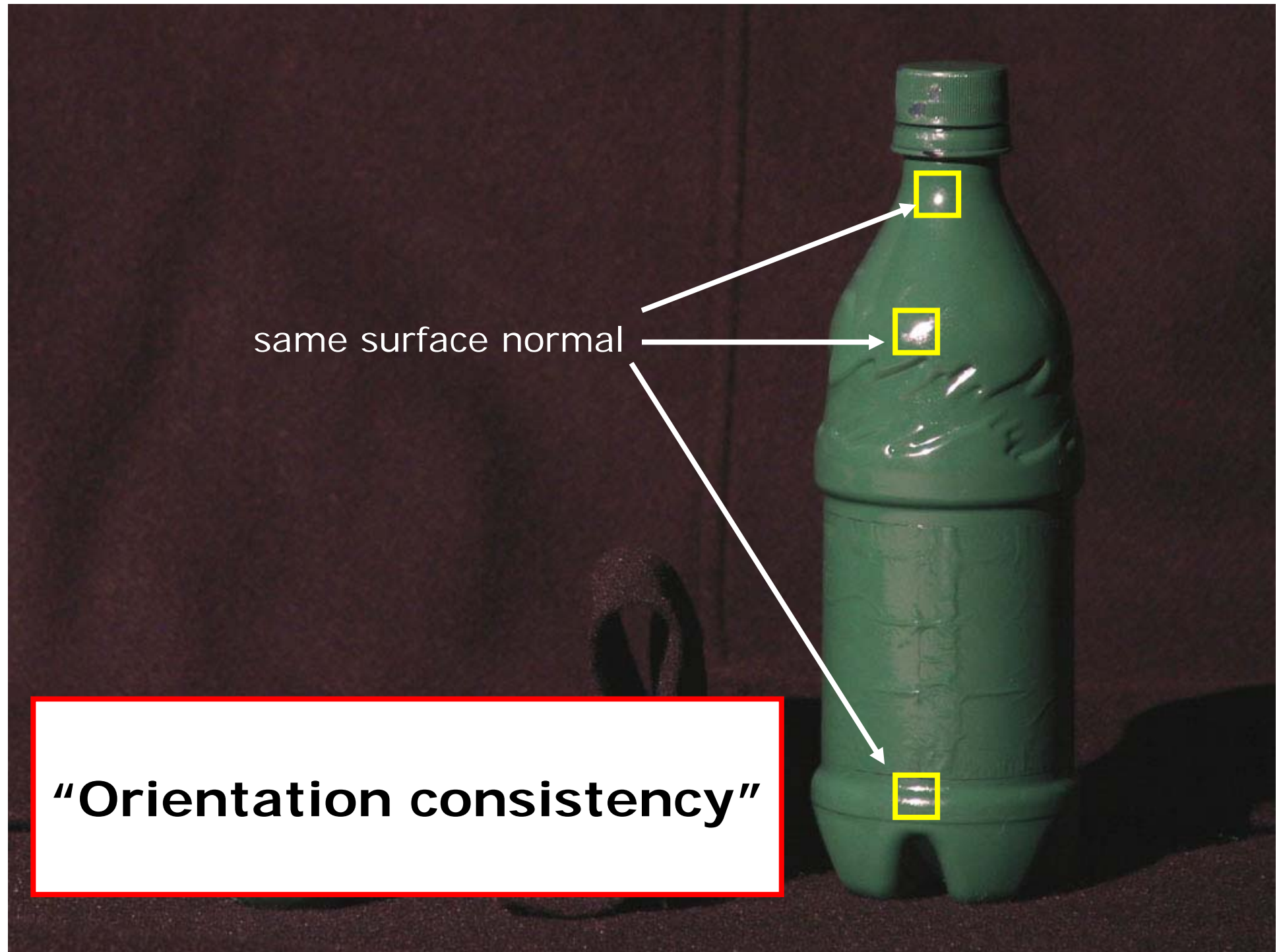
- Big problems
 - doesn't work for shiny things, semi-translucent things
 - shadows, inter-reflections
- Smaller problems
 - calibration requirements
 - measure light source directions, intensities
 - camera response function

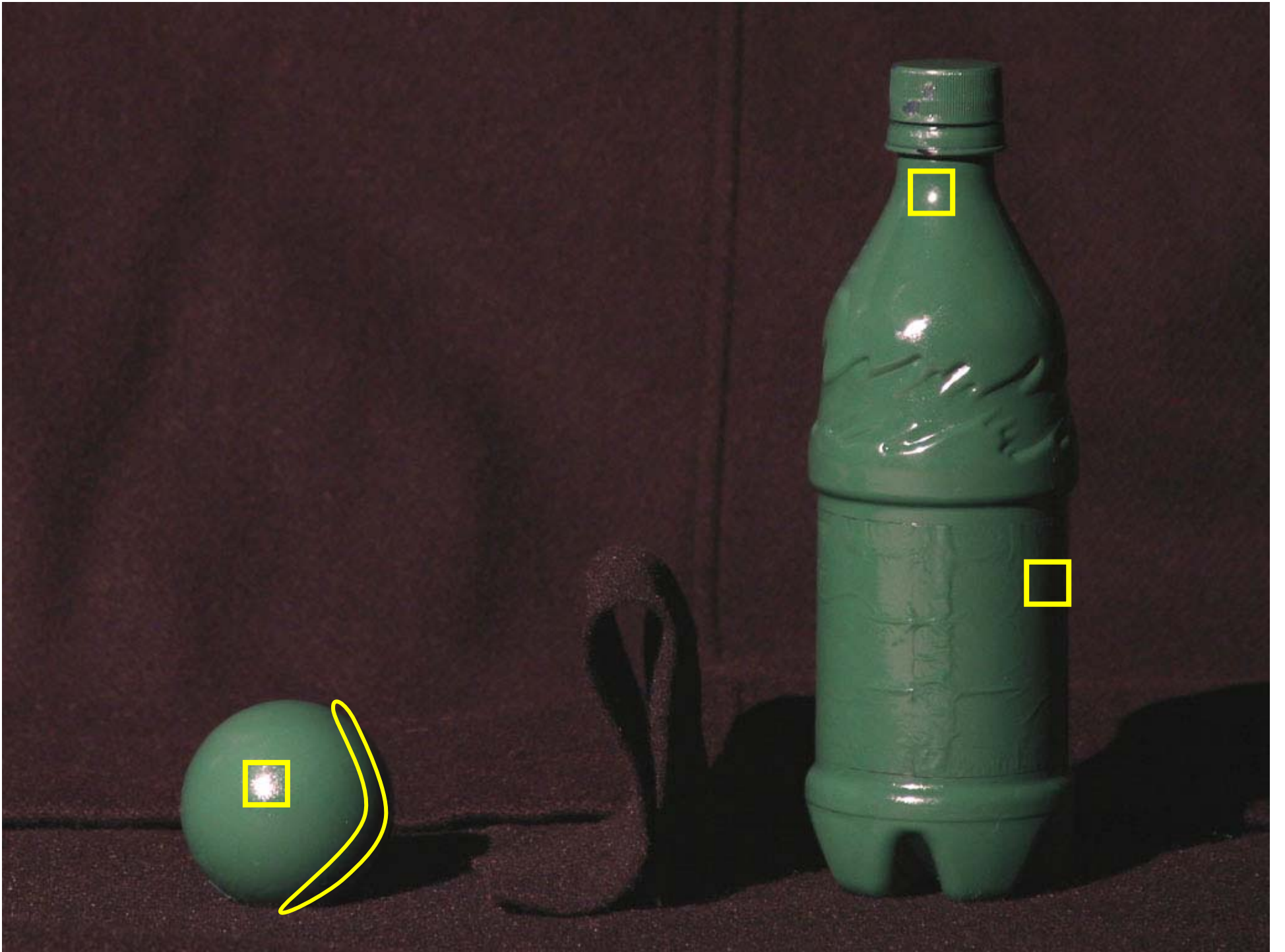
Example-based photometric stereo

- Estimate 3D shape by varying illumination, fixed camera
- Operating conditions
 - any opaque material
 - distant camera, lighting
 - reference object available
 - no shadows, interreflections, transparency

same surface normal

“Orientation consistency”



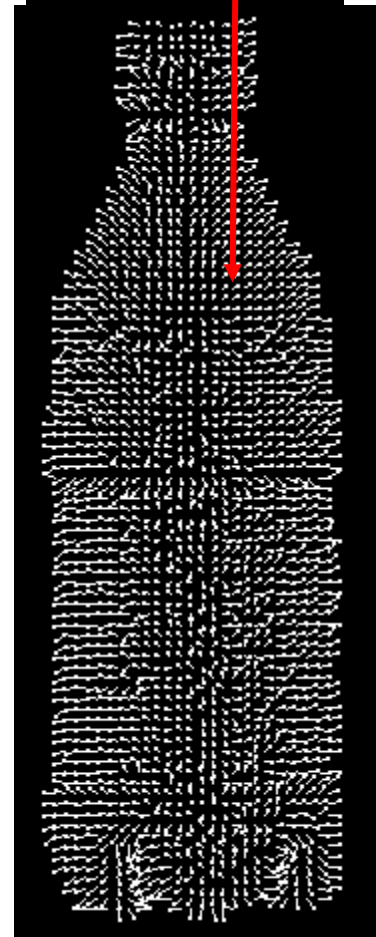
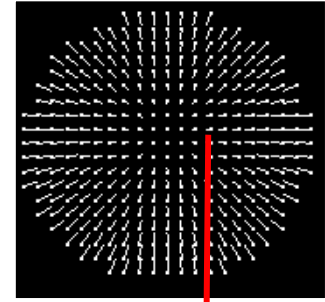




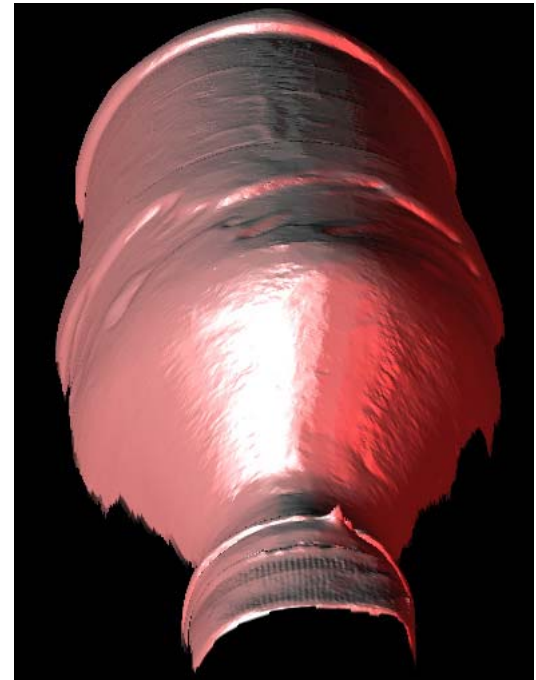








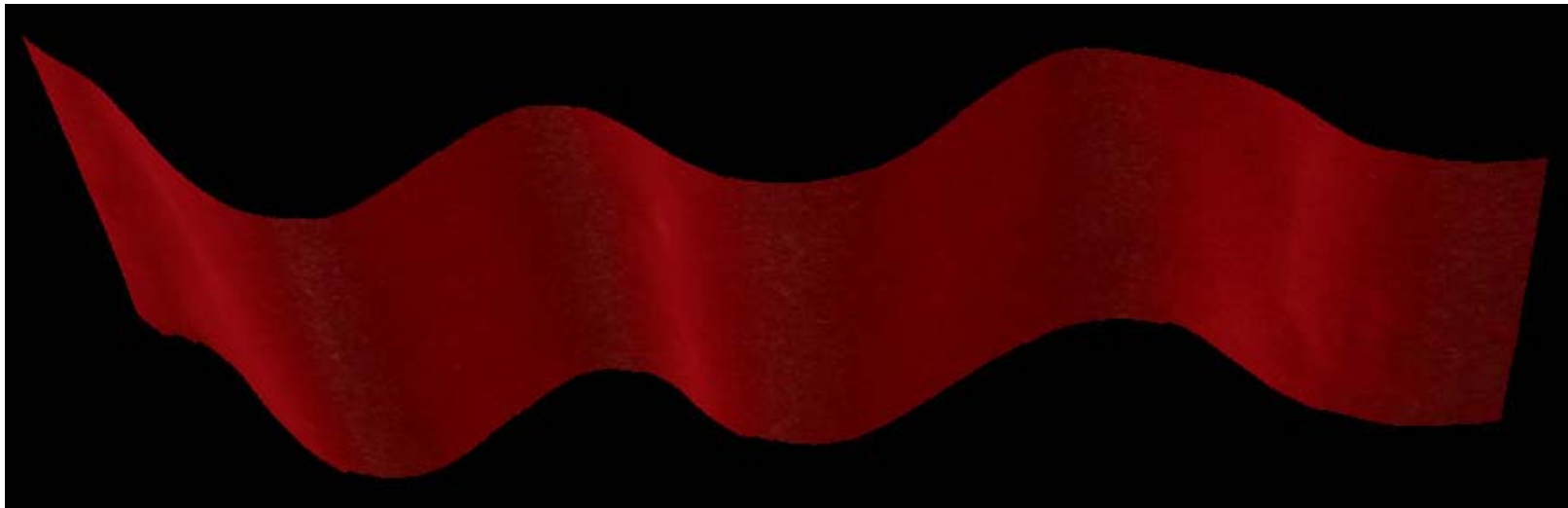
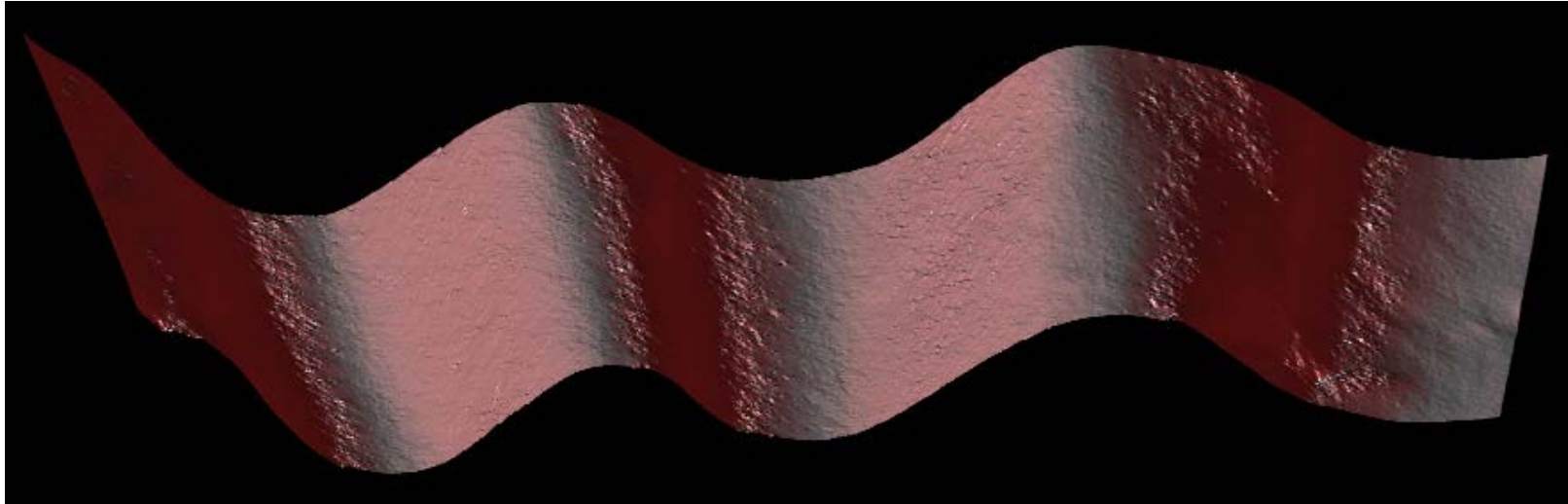
Virtual views



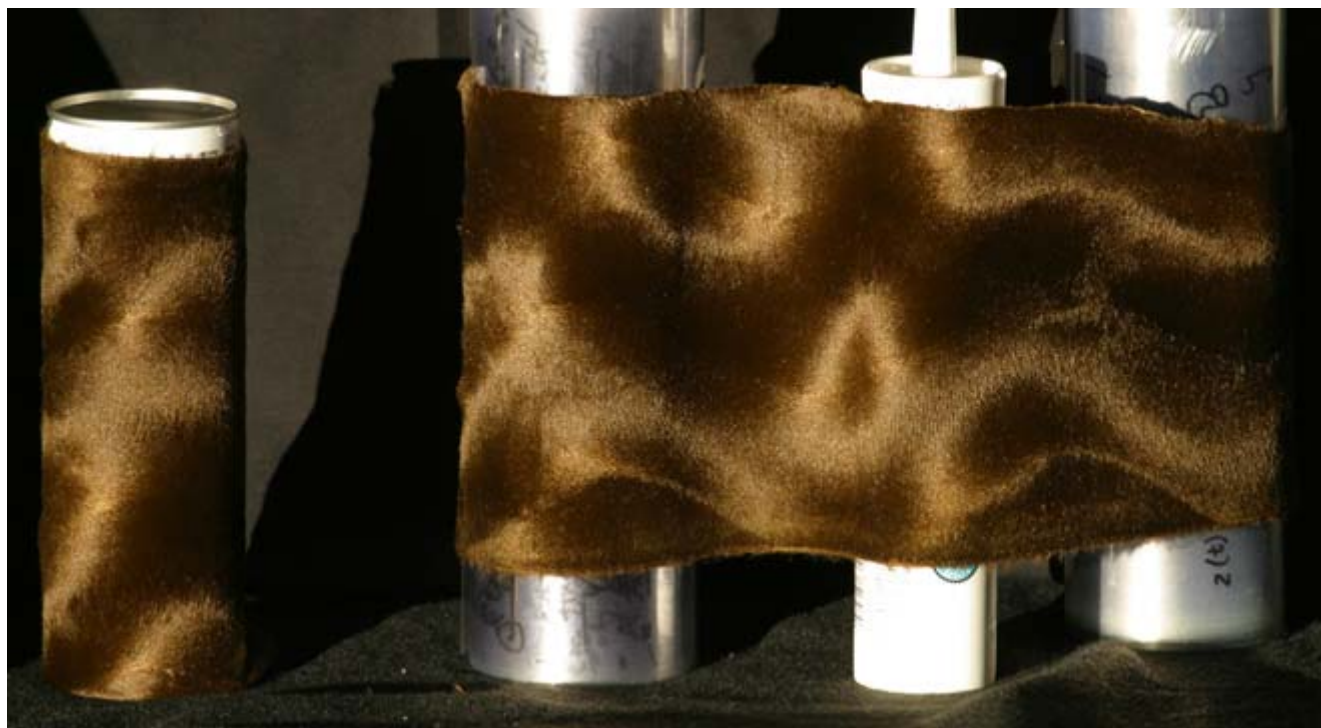
Velvet



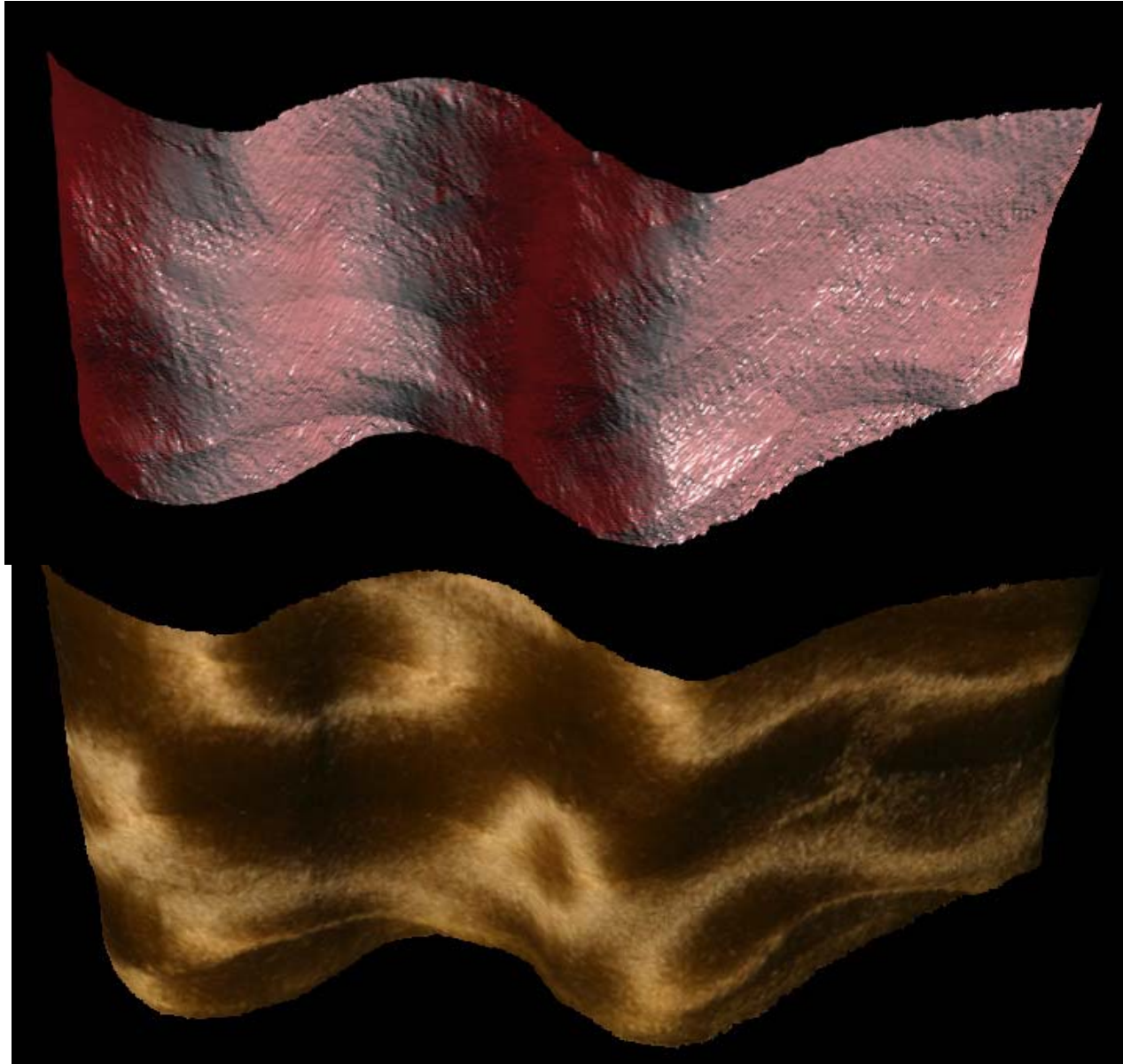
Virtual Views



Brushed Fur



Virtual Views



Salem Specialty Ball Company

[Home](#) | [Materials](#) | [Production](#) | [Inventory](#) | [Charts](#) | [Tools](#) | [Company](#) | [Contact](#)

[Quality Control](#) | [Phone & Fax](#) | [Addresses](#) | [E-mail Directory](#) | [Methods of Payment](#)

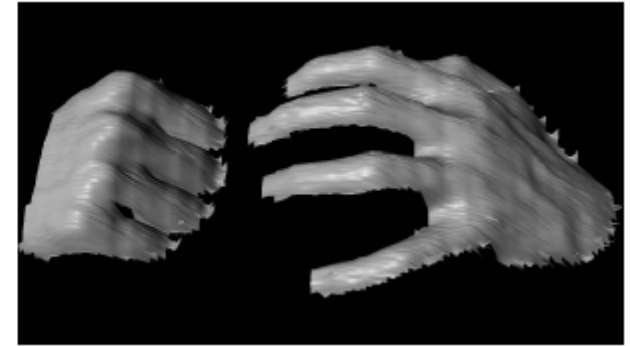
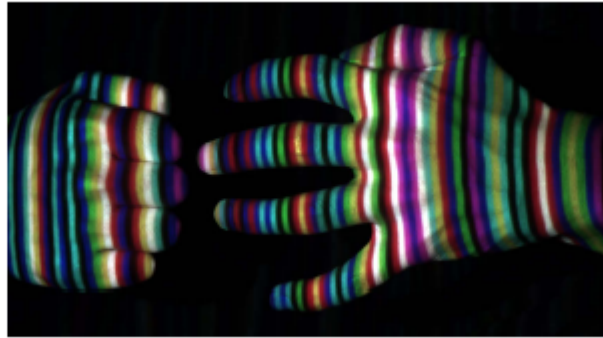
Salem Specialty Ball supplies industrial grade balls that are used in bearings, pumps, valves and other commercial applications. We can supply balls in just about any size that is machineable. We have produced precision balls from .002" all the way up to 12.0" and beyond. We can also produce these balls in any material. Almost without exception, if the material exists, we can make it into a ball. Not only do we specialize in hard to find materials, we also carry standard materials such as [chrome steel](#) and the [stainless steels](#). We stock an extensive [inventory](#) of ready to ship balls. Most orders are shipped the same day. And if it isn't in stock, we can make it for you in matter of days. In addition, you will find that our prices are very competitive.



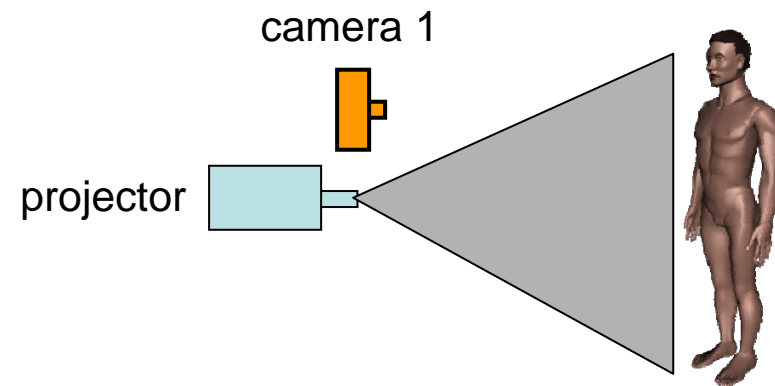
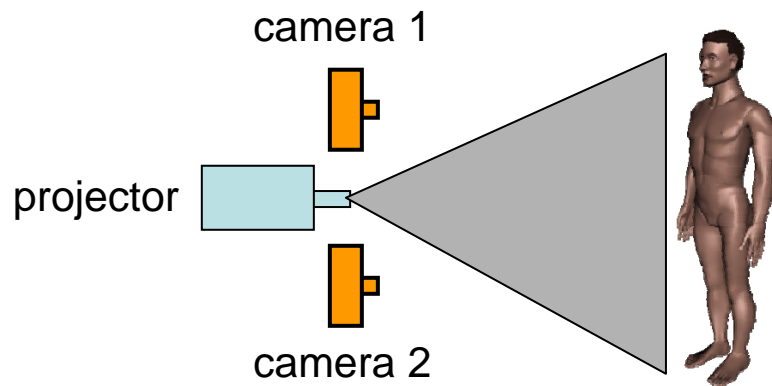
Located in the beautiful northwest corner of Connecticut, Canton has been our company's home for the last three years and we have been in complete operation for over ten years. Proud of our reputation, Salem Specialty Ball Company has over fifty years of combined experience allowing us to provide top-notch quality technical support and expert engineering consultation



Active stereo with structured light DigiVFX



Li Zhang's one-shot stereo



- Project “structured” light patterns onto the object
 - simplifies the correspondence problem

Spacetime Stereo



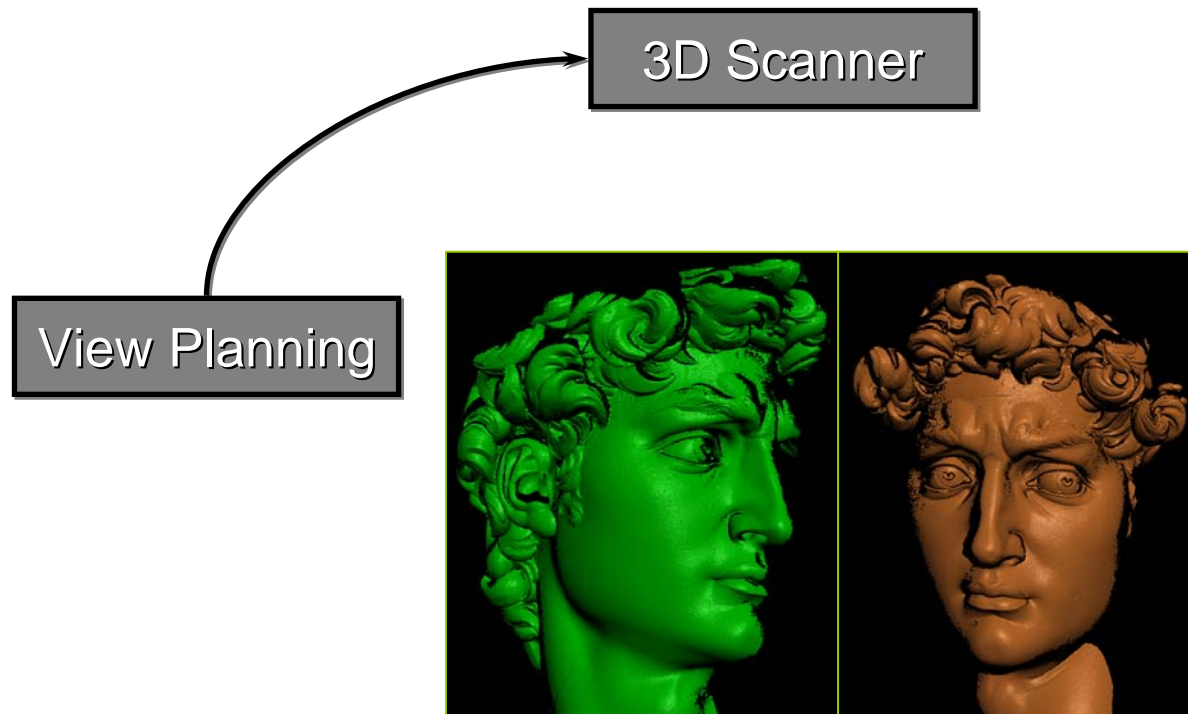
<http://grail.cs.washington.edu/projects/stfaces/>

3D Model Acquisition Pipeline

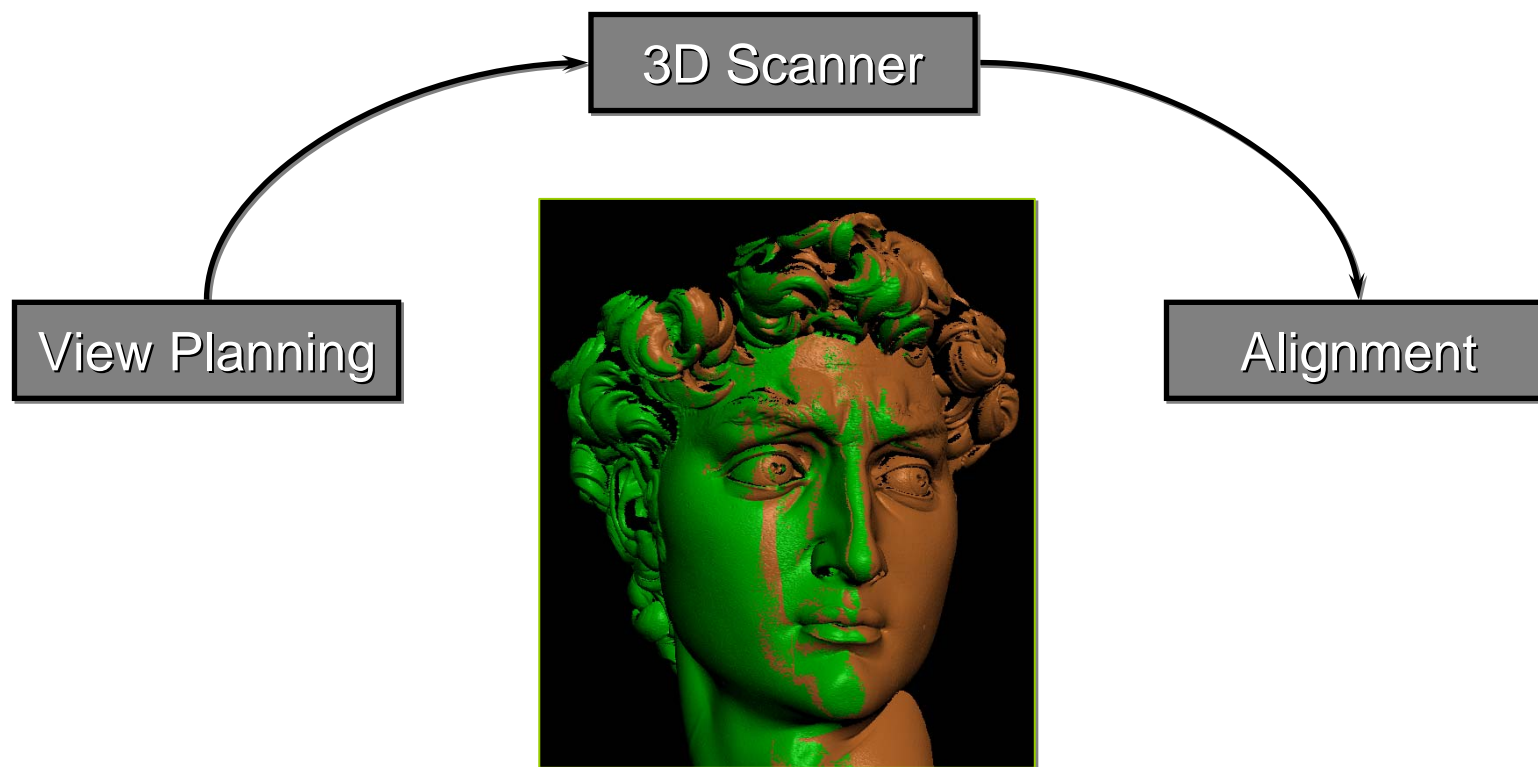
3D Scanner



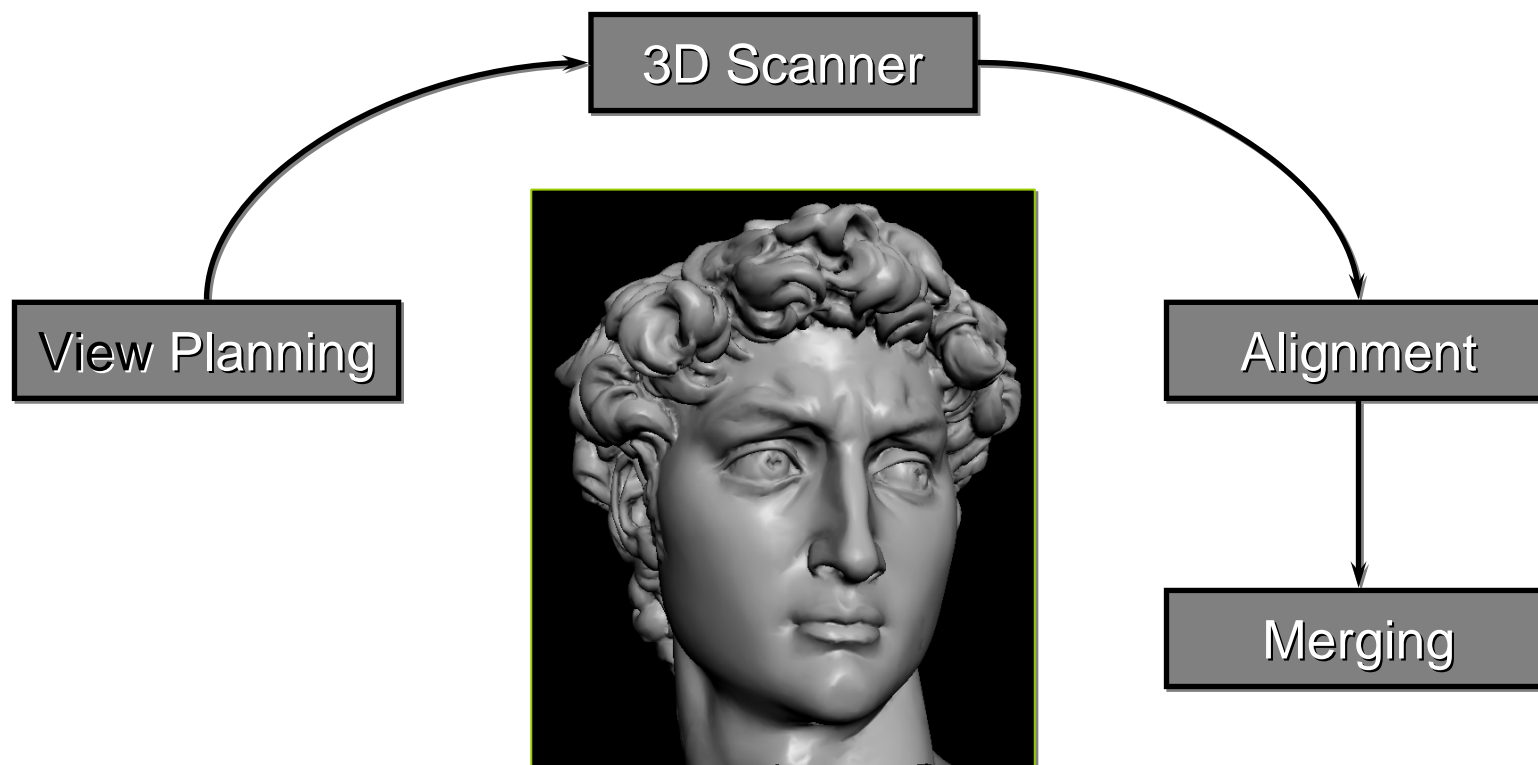
3D Model Acquisition Pipeline



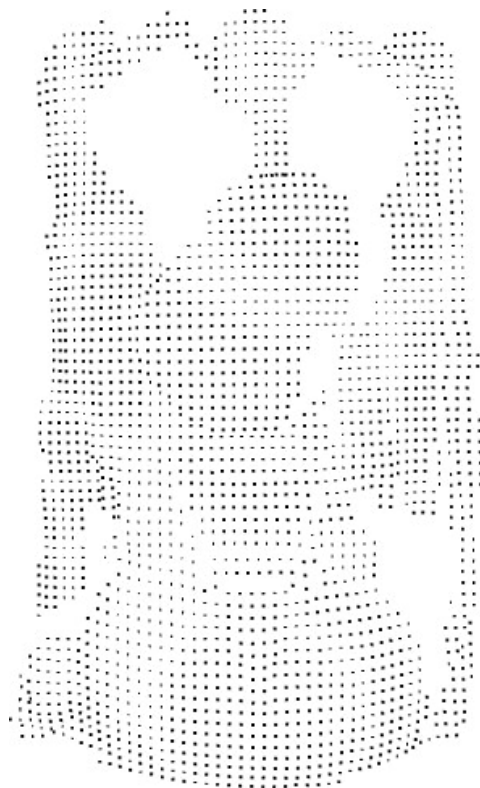
3D Model Acquisition Pipeline



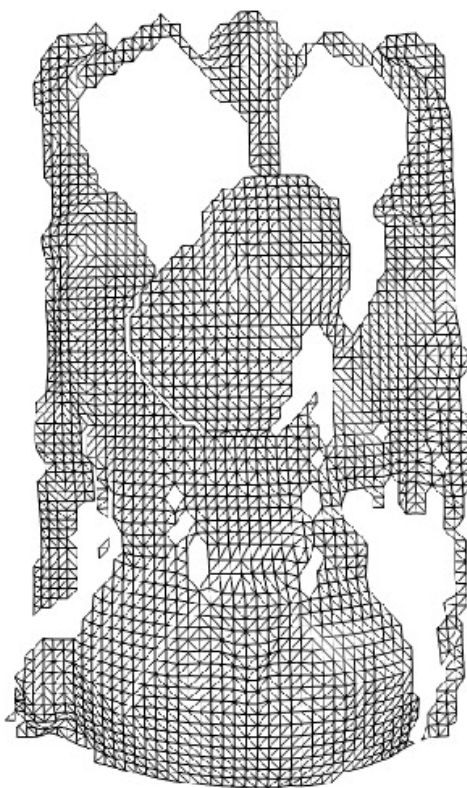
3D Model Acquisition Pipeline



Volumetric reconstruction



Range image

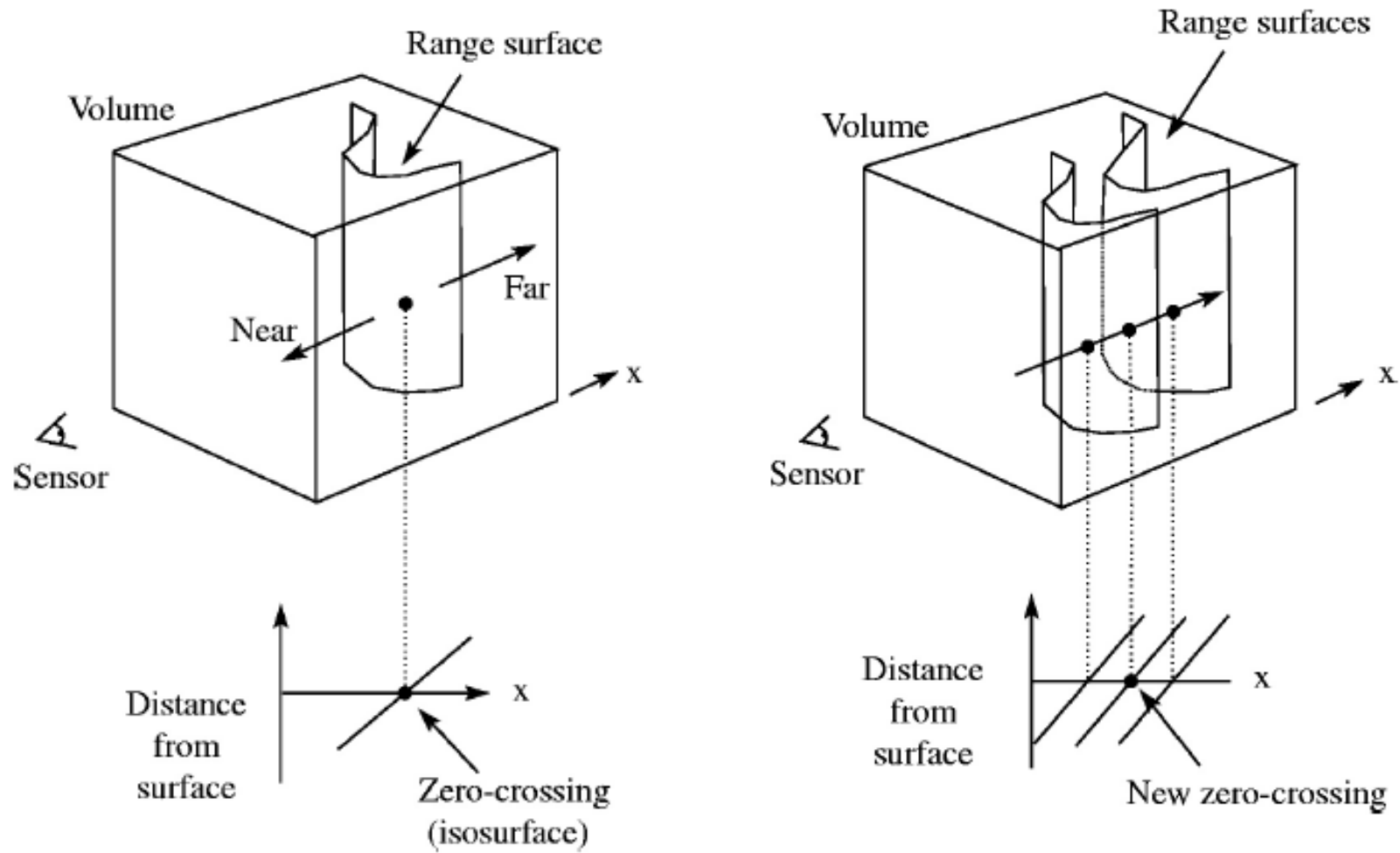


Tessellation

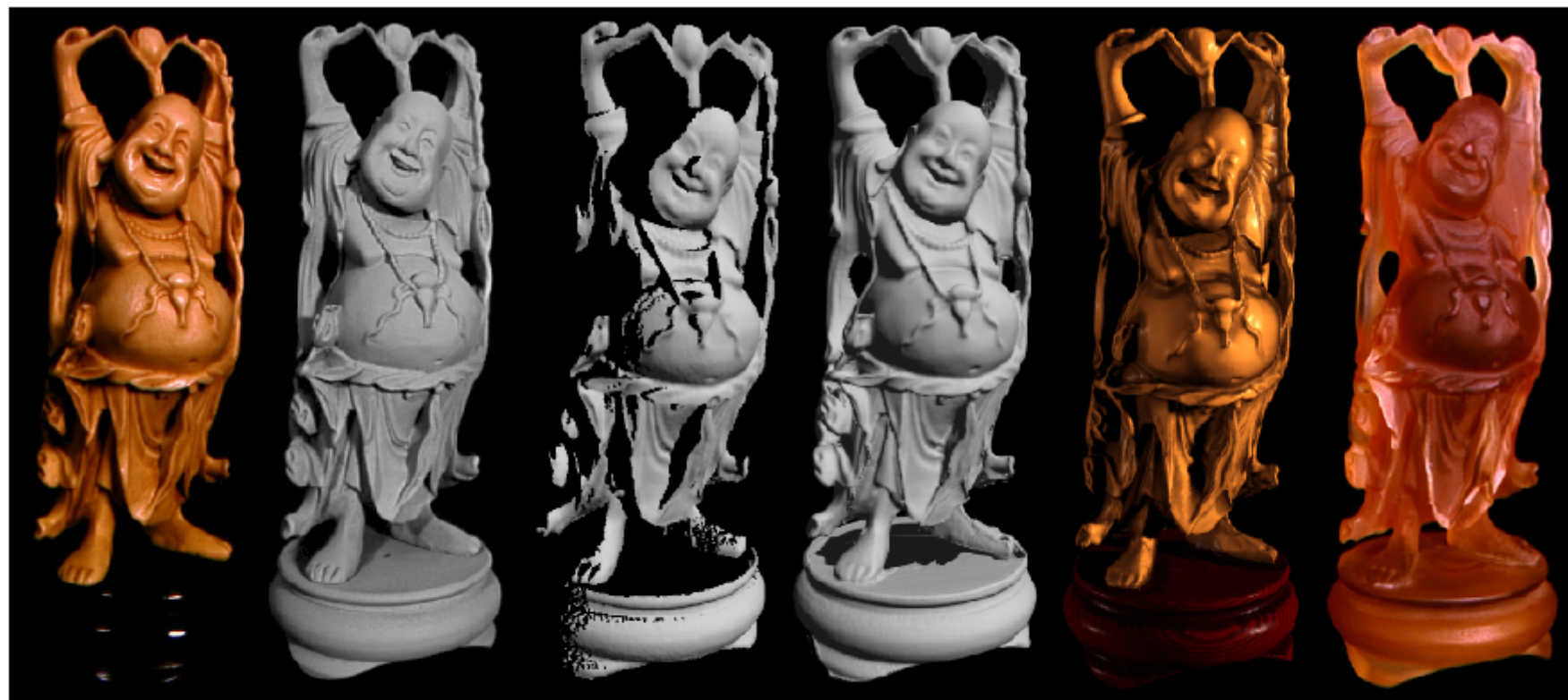


Range surface

Signed distance function



Results



Photograph of original model

Photograph of painted original

Range surface from one scan

Reconstruction before hole-filling

Reconstruction after hole-filling

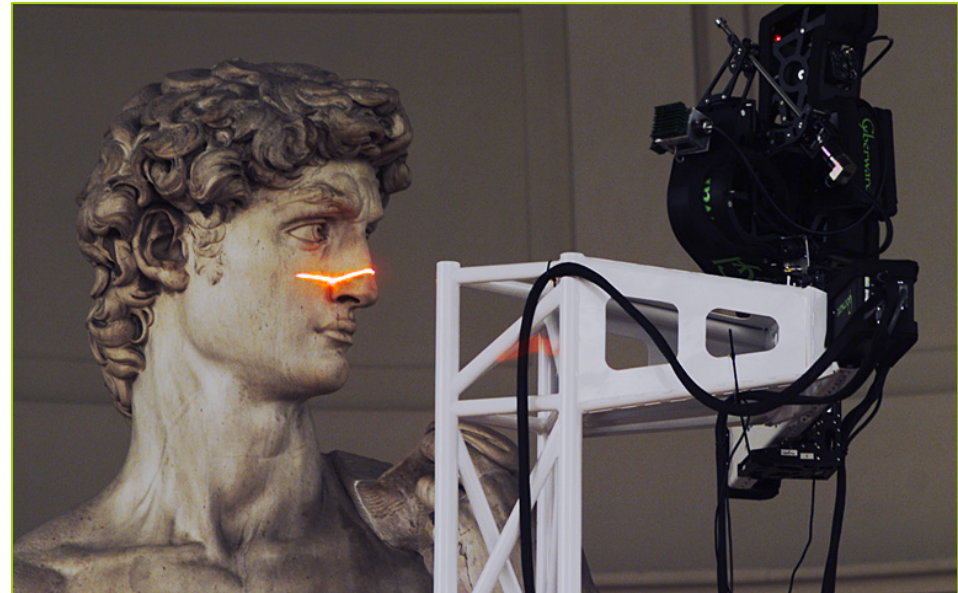
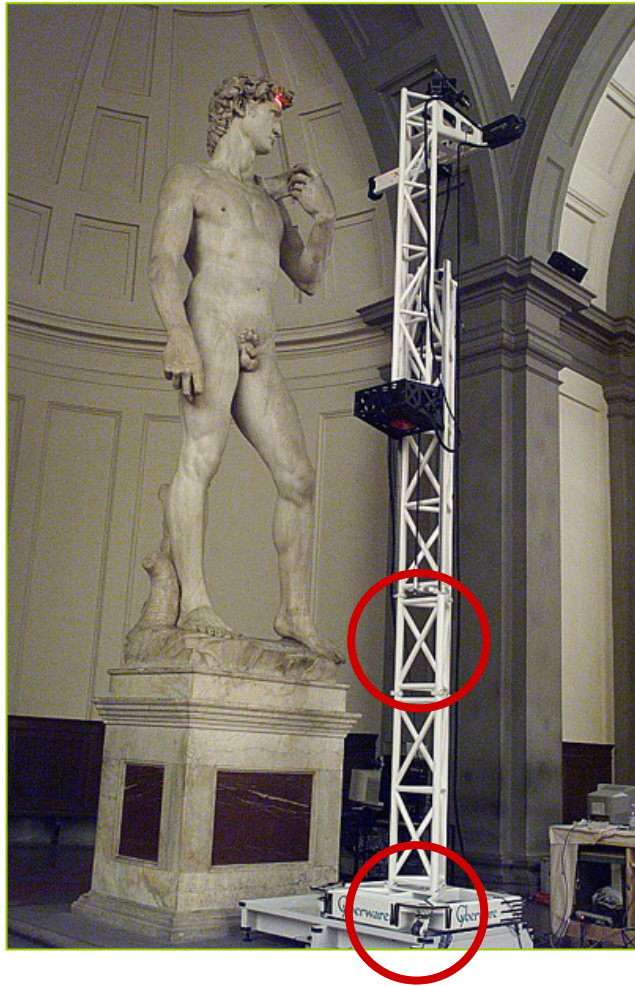
Hardcopy

The Digital Michelangelo Project

- Goal: scan 10 sculptures by Michelangelo
- High-resolution (“quarter-millimeter”) geometry
- Stanford University, led by Marc Levoy

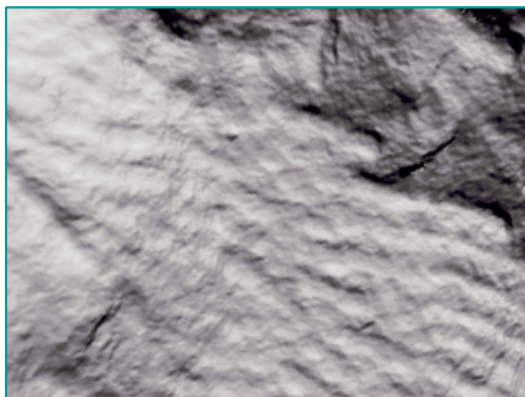
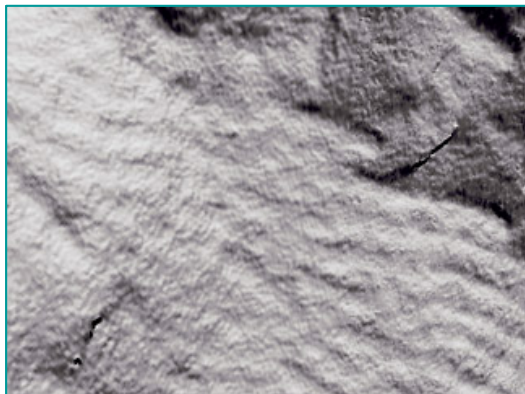
Systems, projects and applications

Scanning the David



height of gantry: 7.5 meters
weight of gantry: 800 kilograms

Range processing pipeline



- steps
 1. manual initial alignment
 2. ICP to one existing scan
 3. automatic ICP of all overlapping pairs
 4. global relaxation to spread out error
 5. merging using volumetric method

Statistics about the scan



- 480 individually aimed scans
- 2 billion polygons
- 7,000 color images
- 32 gigabytes
- 30 nights of scanning
- 22 people

Comparison



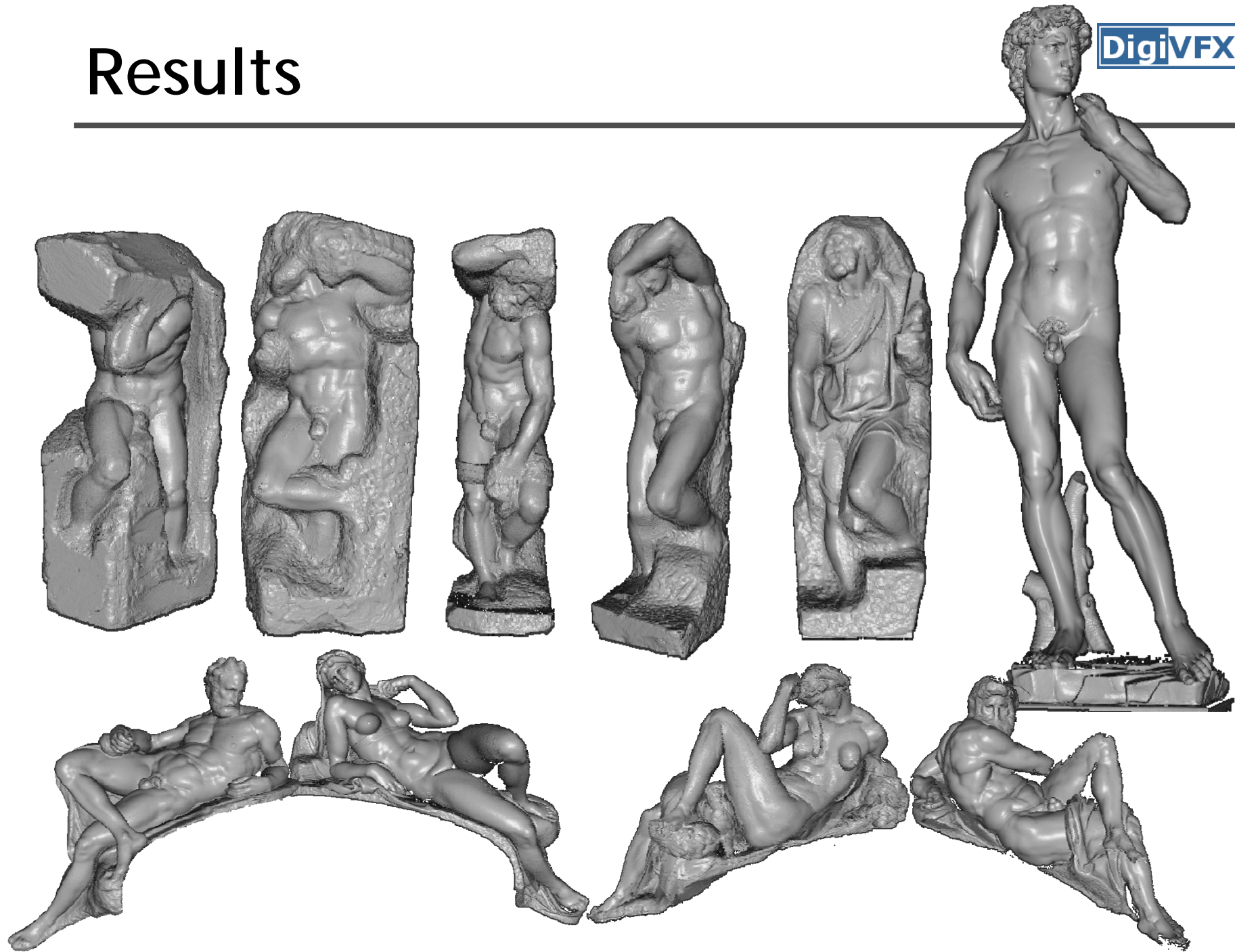
photograph



1.0 mm computer model

Results

DigiVFX



The Great Buddha Project

- Great Buddha of Kamakura
- Original made of wood, completed 1243
- Covered in bronze and gold leaf, 1267
- Approx. 15 m tall
- Goal: preservation of cultural heritage
- Institute of Industrial Science, University of Tokyo, led by Katsushi Ikeuchi



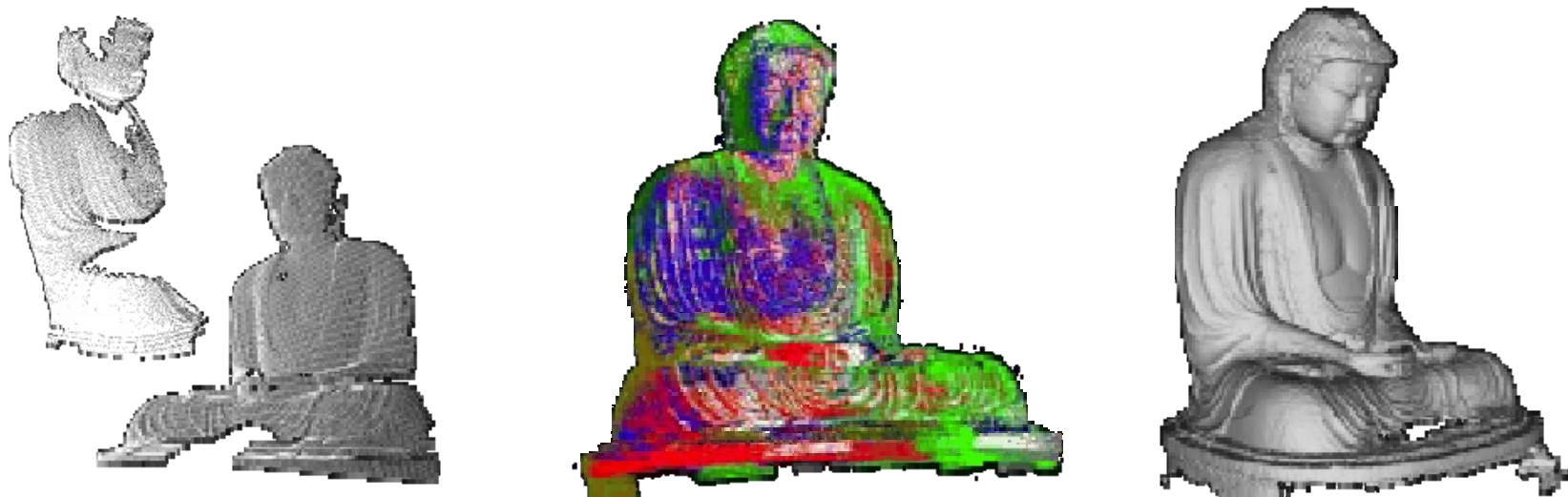
Scanner

- Cyrax range scanner by Cyra Technologies
- Laser pulse time-of-flight
- Accuracy: 4 mm
- Range: 100 m

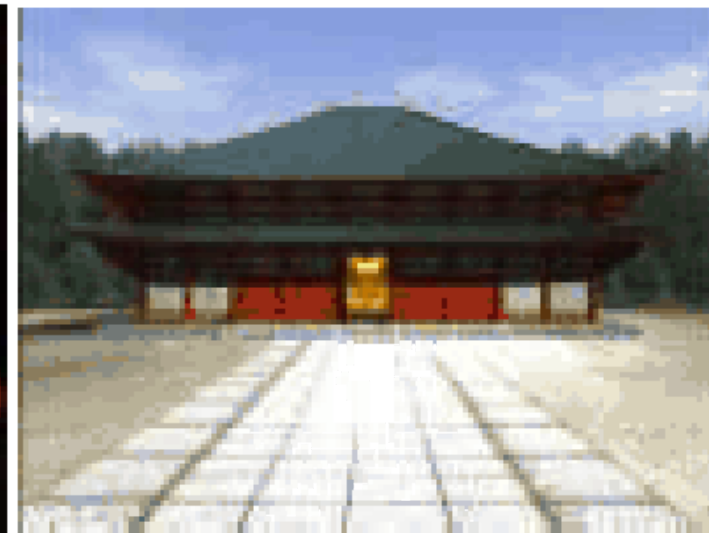
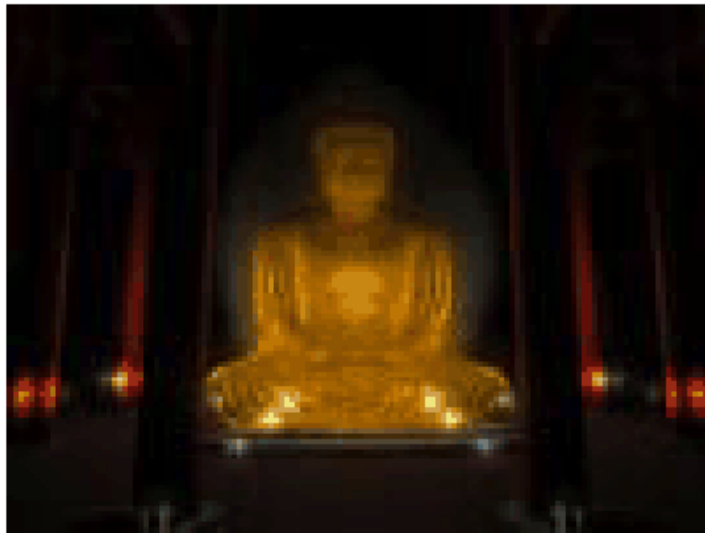
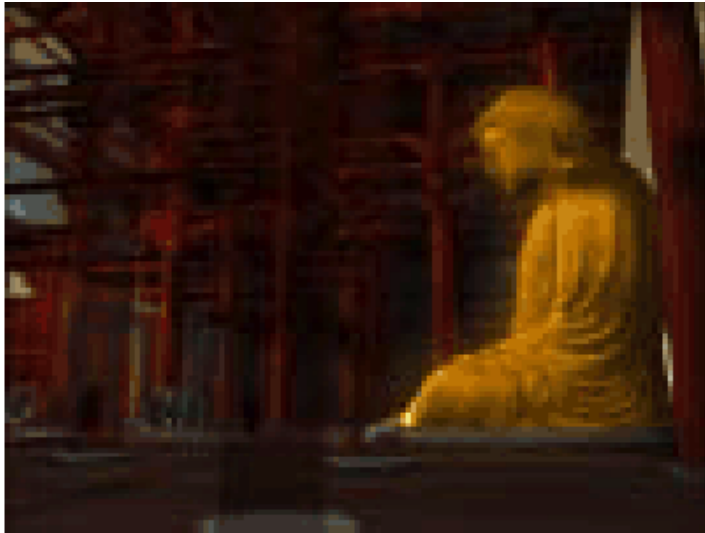


Processing

- 20 range images (a few million points)
- Simultaneous all-to-all ICP
- Variant of volumetric merging (parallelized)



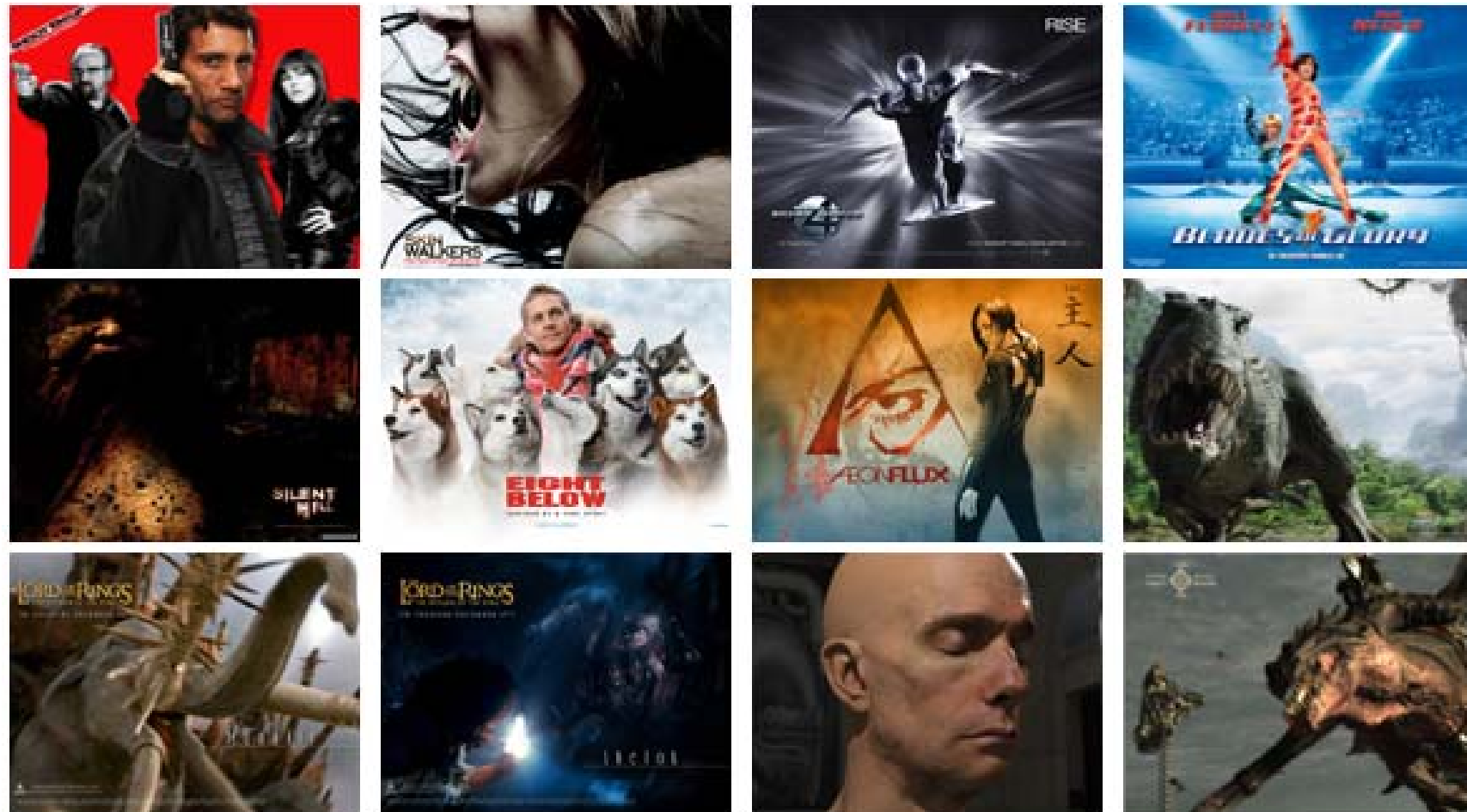
Results



Applications in VFX

- 3D scanning
- Hybrid camera for IMAX
- View interpolation

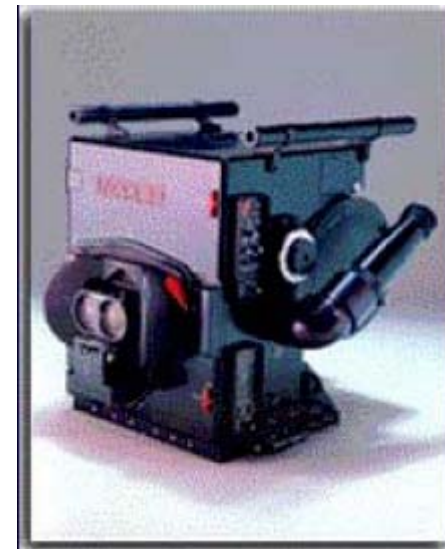
3D scanning



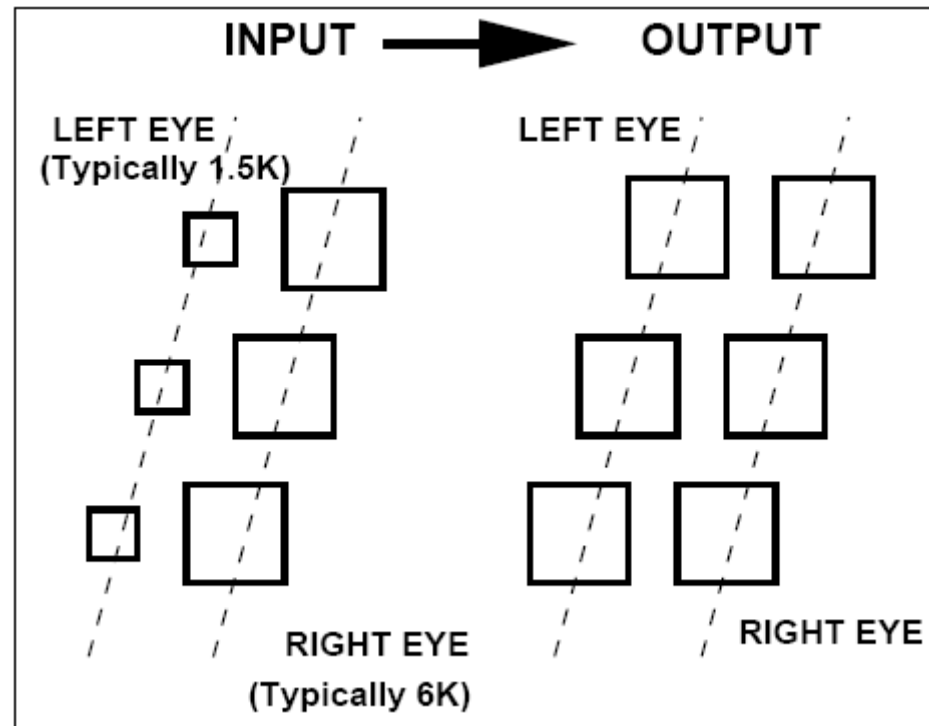
XYZRGB Inc.

IMAX 3D

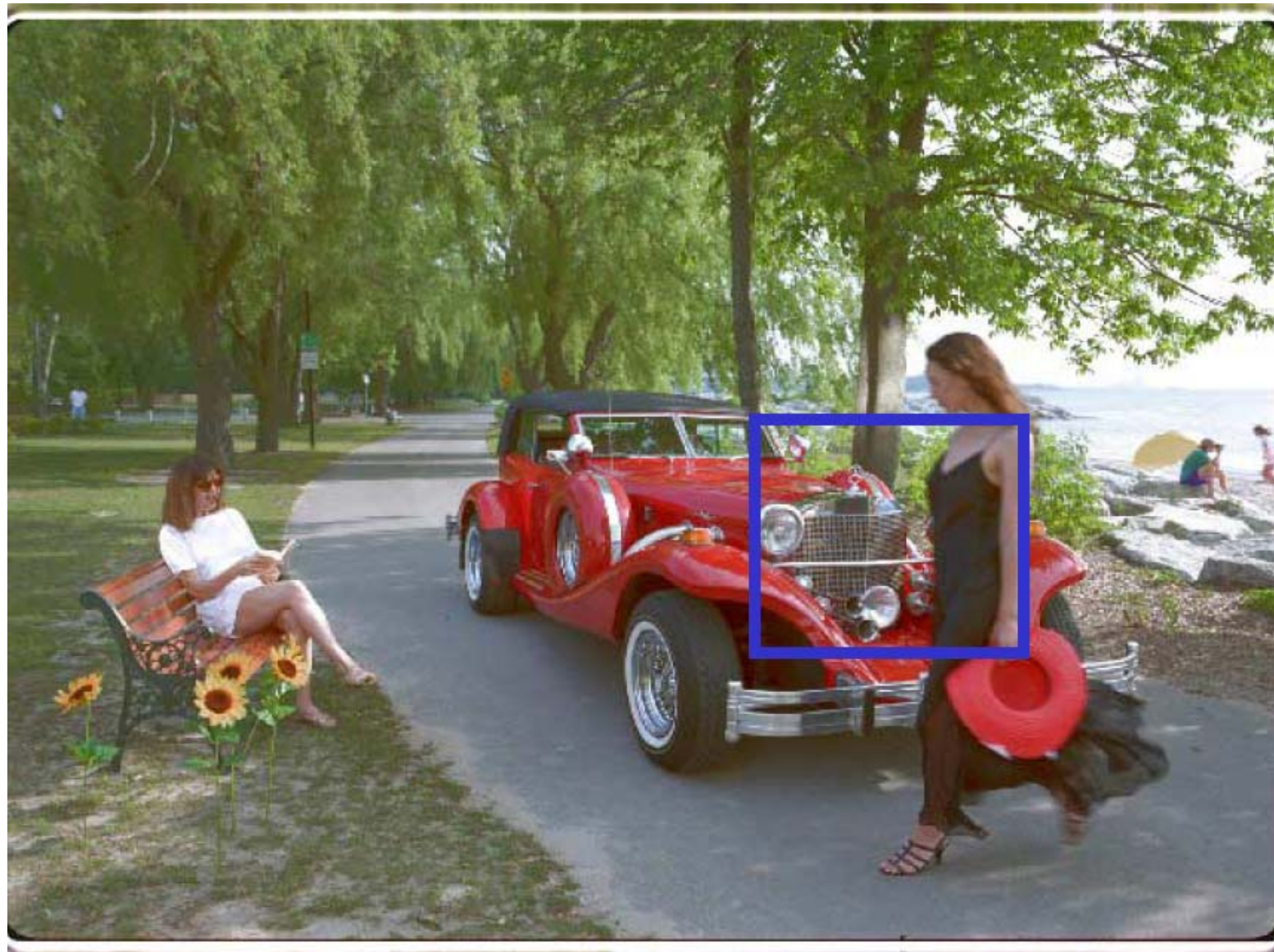
- 6K resolution, 42 linear bits per pixel
- For CG, it typically takes 6 hours for a frame
- 45-minute IMAX 3D CG film requires a 100-CPU rendering farm full-time for about a year just for rendering
- For live-action, camera is bulky (like a refrigerator)



Hybrid stereo camera



Live-action sequence



Hybrid input



left



right

Hybrid input



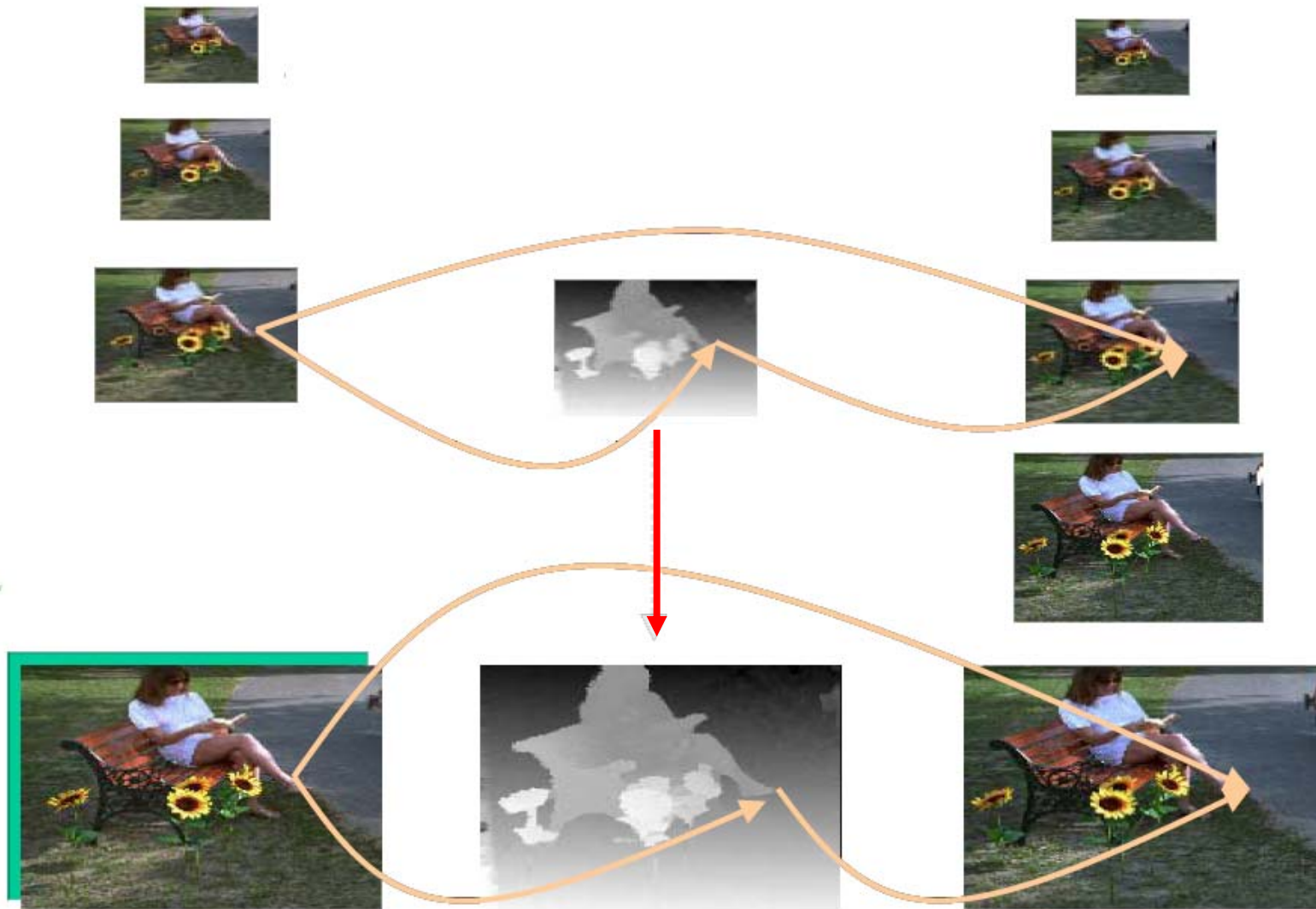
left



right



Combine multiple hires to lores



Results



View interpolation



Bullet time video

View interpolation



High-quality video view interpolation

Final project

Final project

- Assigned: 5/14
- Due: 6/25 Wednesday
- Proposal and midterm report on 5/27

Final project

- Research (1-2 people)
- System (1-3 people)
- Film (3-4 people)

Research

- Define a problem and try to solve it
- You don't need to solve it all, but have to make a reasonable progress, for example, solve a simplified version.
- Find inspirations from SIGGRAPH/CVPR/ICCV papers

System

- Implement existing algorithm into a useful system such as implementing SIGGRAPH 2006/2007/2008 or CVPR/ICCV 2006/2007/2008 papers

Film

- It must be an “effect” film.
- You can use any tools as you want. But, I assume that you have to write some on your own.
- Find inspirations from
Gatech’s vfx course
http://www.cc.gatech.edu/classes/AY2004/cs4480_spring/
independent film makers
<http://www.peerlessproductions.com/>
ADs/films/YouTube
- Submit two videos, final and making-of.