3D photography

Digital Visual Effects, Spring 2007

Yung-Yu Chuang

2007/5/15

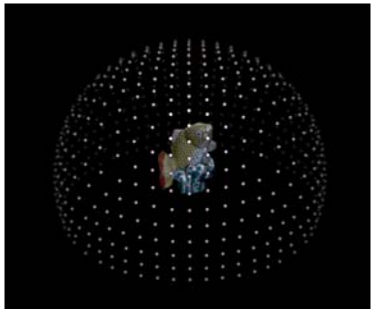
with slides by Szymon Rusinkiewicz, Richard Szeliski, Steve Seitz and Brian Curless

Announcements

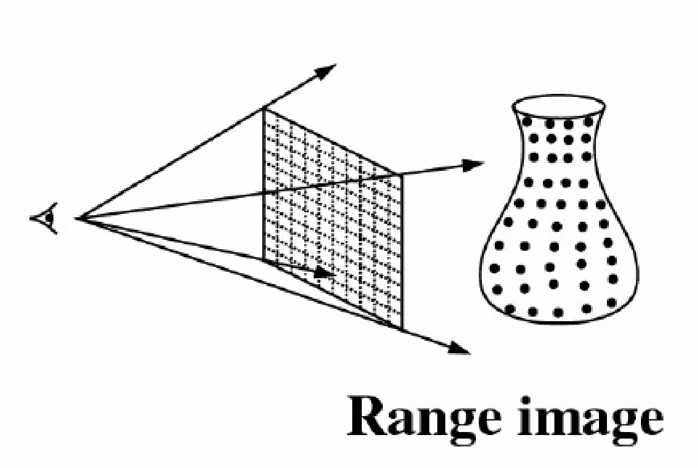
- Project #3 is due on 5/20. Same submission mechanism. Two videos. Send TA links only.
- We will have final project proposal presentation on 5/29 (or 6/5). Please send me your team members and topic by 5/27 (or 6/3).
- Final project demo day will be 1:30pm on 6/27 (Wed). Room to be announced.

3D photography

- Acquisition of geometry and material



Range acquisition



Range acquisition taxonomy

- range acquisition
  - contact
    - mechanical (CMM, jointed arm)
    - inertial (gyroscope, accelerometer)
    - ultrasonic trackers
    - magnetic trackers
  - transmissive
    - industrial CT
    - ultrasound
    - MRI
  - reflective
    - non-optical
      - radar
      - sonar
    - optical

Range acquisition taxonomy

- optical methods
  - passive
    - shape from X:
      - stereo
      - motion
      - shading
      - texture
      - focus
      - defocus
  - active
    - active variants of passive methods
      - Stereo w. projected texture
      - Active depth from defocus
      - Photometric stereo
    - time of flight
    - triangulation

Outline

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

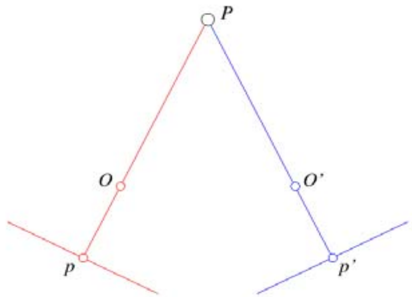
Passive approaches

Stereo

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

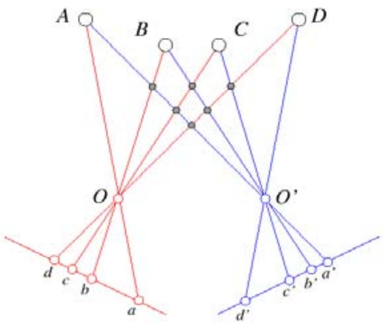
Stereo

- 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

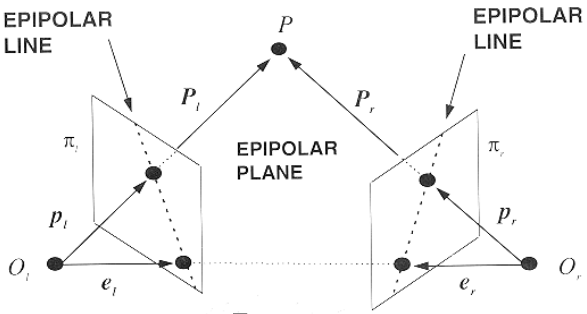


Components of stereo vision systems

- 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

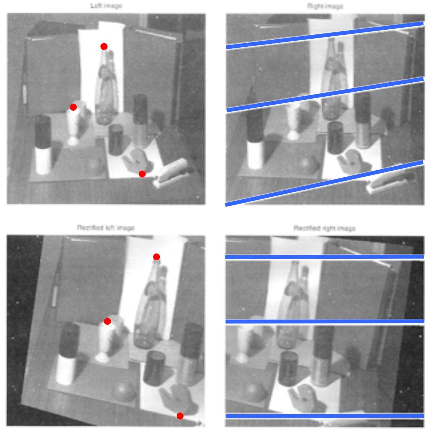
- Epipolar constraint: corresponding points must lie on conjugate epipolar lines
  - Search for correspondences becomes a 1-D problem



$$\mathbf{x}'^{\mathrm{T}}\mathbf{F}\mathbf{x}=0$$

Image rectification

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

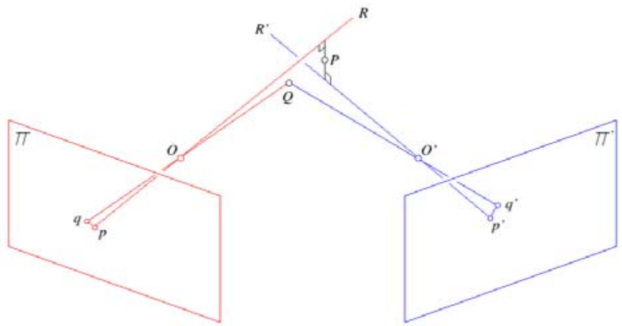


Disparity

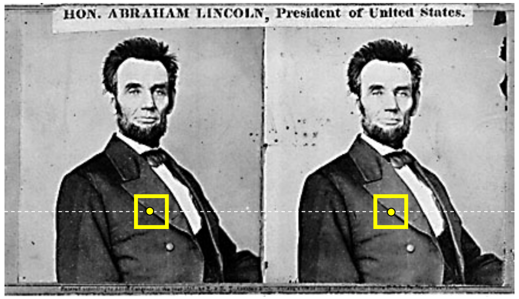
- With rectified images, disparity is just (horizontal) displacement of corresponding features in the two images
  - Disparity = 0 for distant points
  - Larger disparity for closer points
  - Depth of point proportional to 1/disparity

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

Image 1

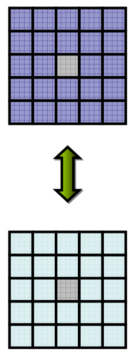
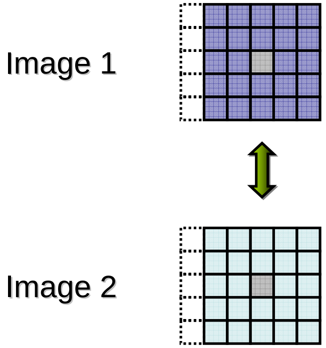


Image 2

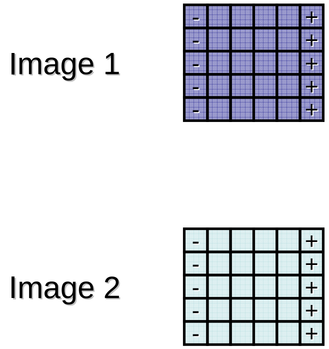
Incremental computation

- Want: SSD at next location



Incremental computation

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

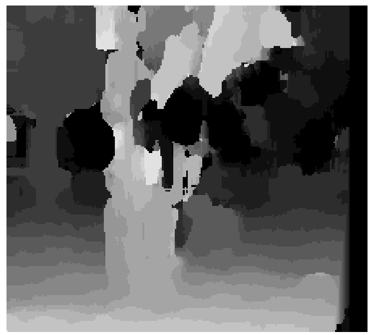


Selecting window size

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

Selecting window size

3 pixel window



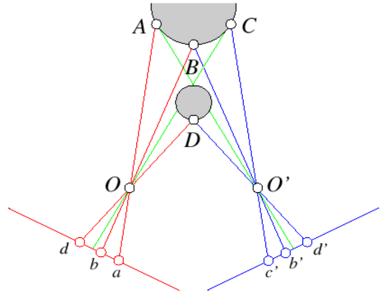
20 pixel window

Non-square windows

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

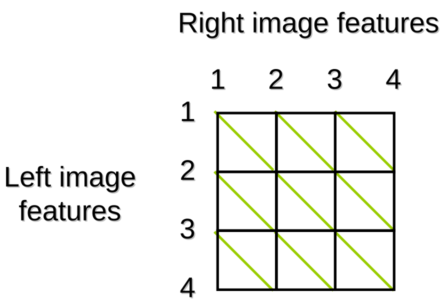
Ordering constraint

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



Dynamic programming

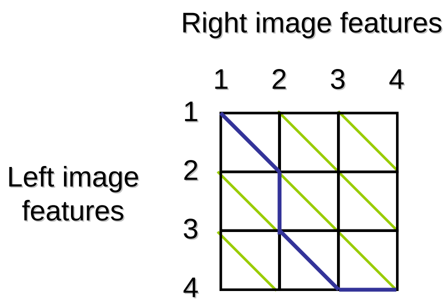
- Treat feature correspondence as graph problem

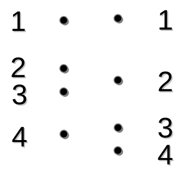


Cost of edges = similarity of regions between image features

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:

  $E_{data}+\lambda E_{smoothness}$
- E_{data}: how well does disparity match data
- $E_{smoothness}$: 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

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

  - "neighborhood term" encouraging spatial smoothness

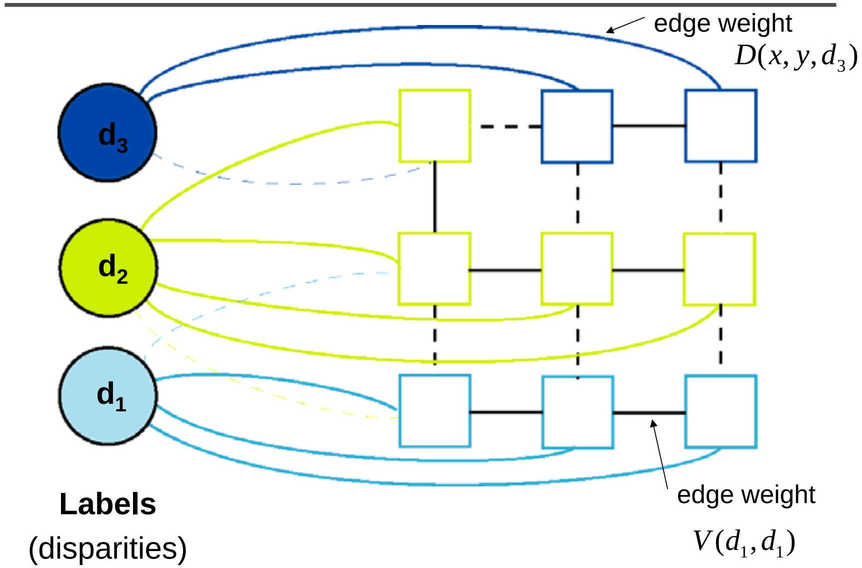
$$V(d_1,d_2)=\text{cost of adjacent pixels with labels d1 and d2}$$
$$=\left|d_1-d_2\right| \quad \text{(or something similar)}$$

$$E=\sum_{(x,y)} D(x,y,d_{x,y})+\sum_{neighbors\ (x1,y1),(x2,y2)} V(d_{x1,y1},d_{x2,y2})$$

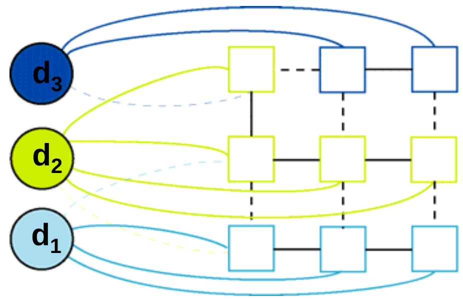
Energy minimization

- 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

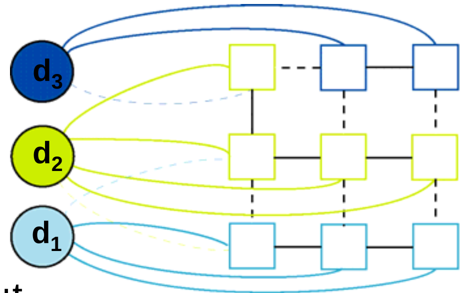


Energy minimization via graph cuts



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

Energy minimization via graph cuts

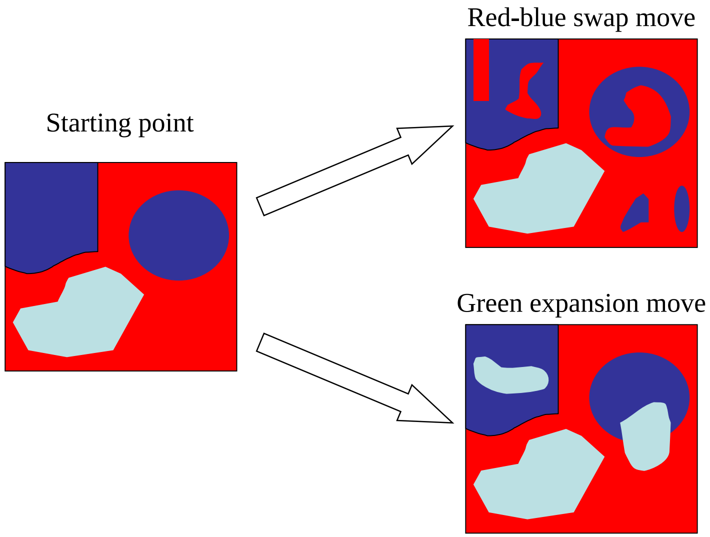


- Graph Cut
  - Delete enough edges so that
    - each pixel is (transitively) connected to exactly one label node
  - Cost of a cut: sum of deleted edge weights
  - Finding min cost cut equivalent to finding global minimum of energy function

Computing a multiway cut

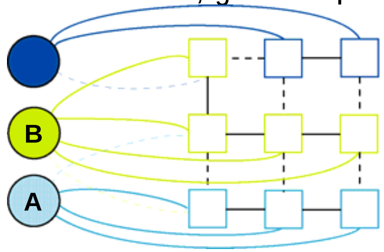
- With 2 labels: classical min-cut problem
  - Solvable by standard flow algorithms
    - polynomial time in theory, nearly linear in practice
  - More than 2 terminals: NP-hard
    [Dahlhaus *et al.*, STOC '92]
- Efficient approximation algorithms exist
  - Within a factor of 2 of optimal
  - Computes 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

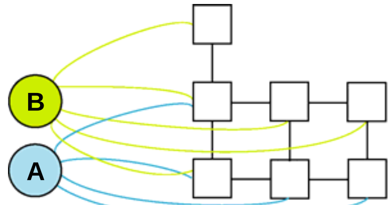


The swap move algorithm

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 it's lower *E* than the current labeling
3. If *E* did not decrease in the cycle, we're done
   Otherwise, go to step 2



Original graph



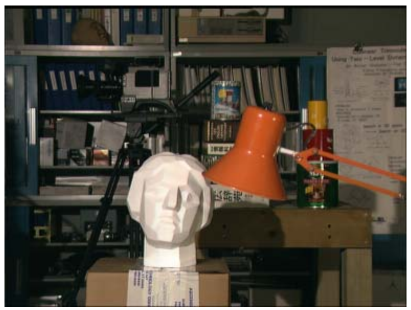
AB subgraph
(run min-cut on this graph)

The expansion move algorithm

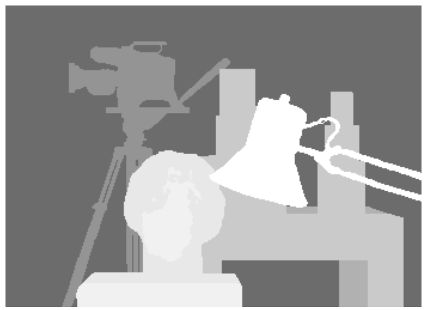
1. Start with an arbitrary labeling
2. Cycle through every label *A* in some order
   - 2.1 Find the lowest *E* labeling within a single *A*-expansion
   - 2.2 Go there if it's lower *E* than the current labeling
3. If *E* did not decrease in the cycle, we're done
   Otherwise, go to step 2

Stereo results

– Data from University of Tsukuba



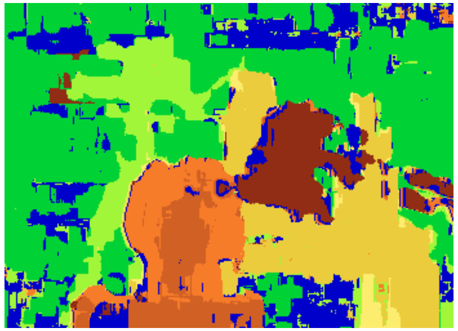
scene



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)

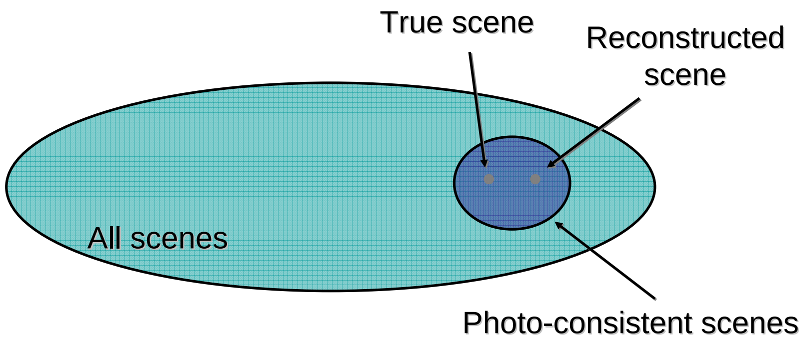
ground truth

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.)

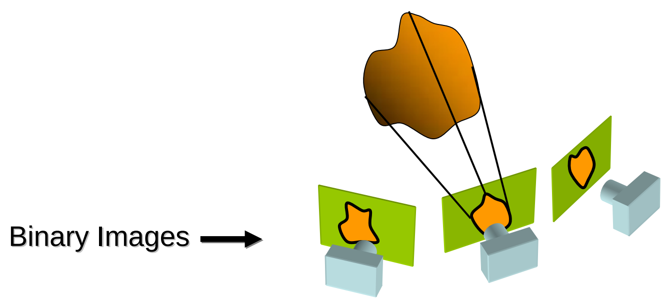
Photo consistency

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



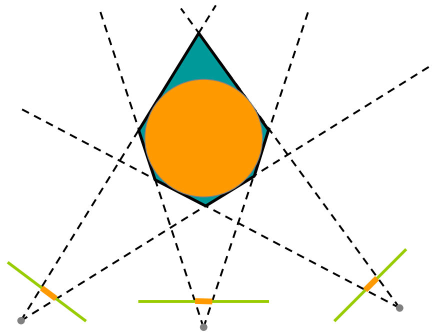
Silhouette carving

- 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



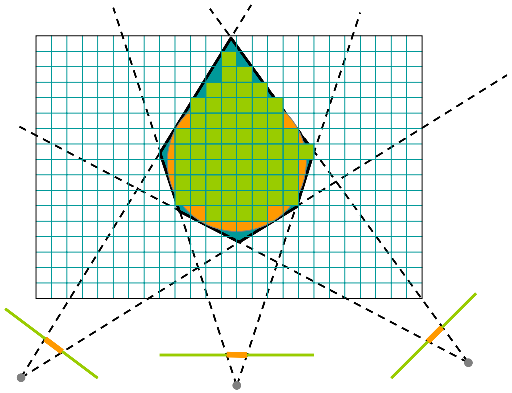
Silhouette carving

- 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

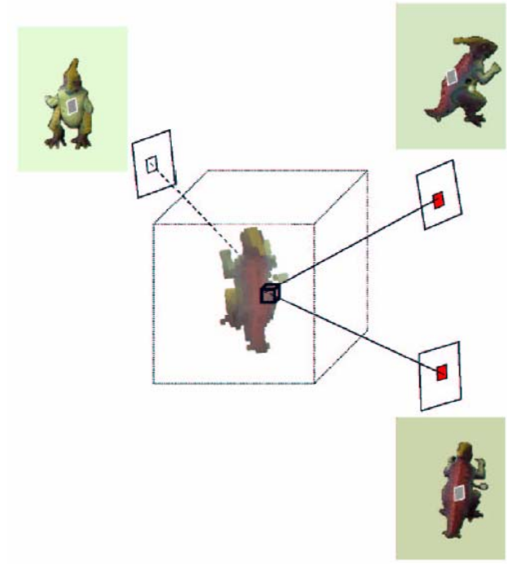
Voxel coloring

- 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

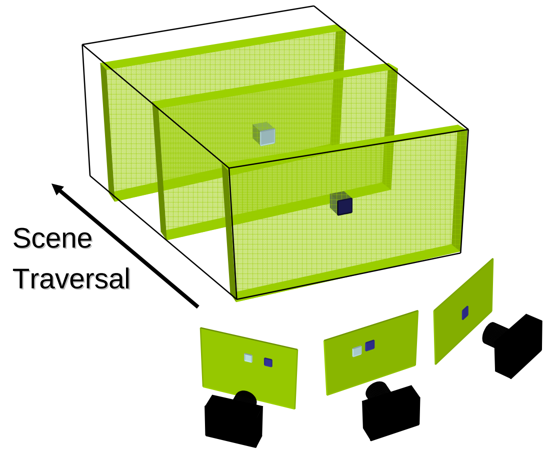
Voxel coloring and occlusion

- 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

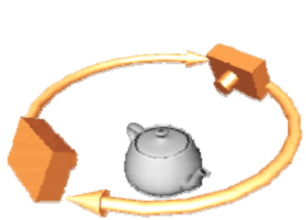
Occlusion handling



Voxel coloring sweep order



Voxel coloring camera positions



Inward-looking
Cameras above scene

Outward-looking
Cameras inside scene

Seitz

Image acquisition

Selected Dinosaur Images

Selected Flower Images

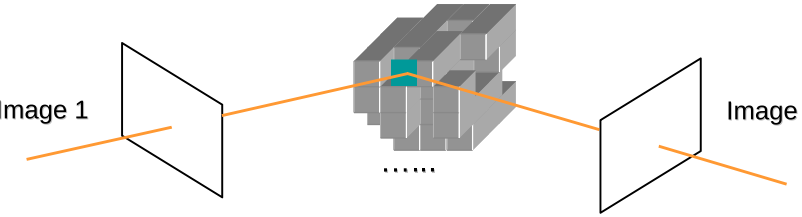
- Calibrated Turntable
- 360° rotation (21 images)

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

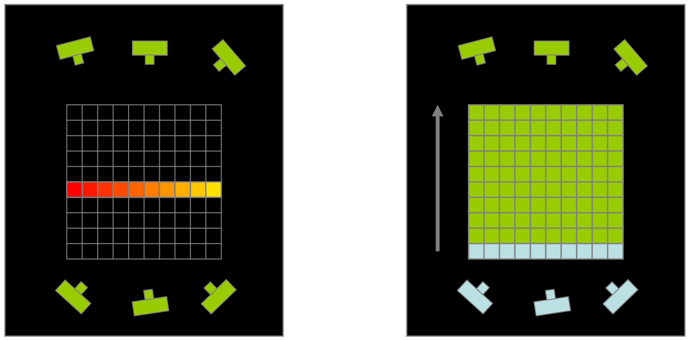


Initialize to a volume V containing the true scene
Choose a voxel on the current surface
Project to visible input images
Carve if not photo-consistent
Repeat until convergence

Multi-pass plane sweep

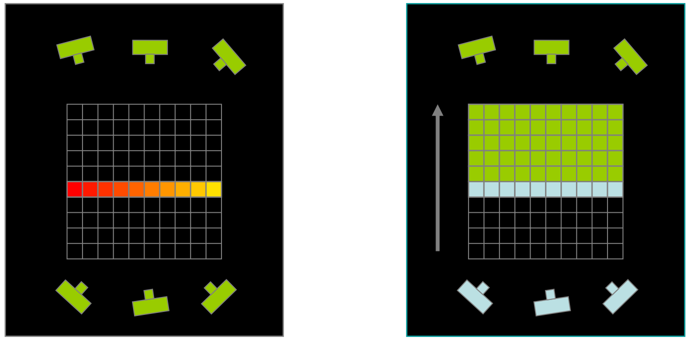
- Faster alternative:
  - Sweep plane in each of 6 principal directions
  - Consider cameras on only one side of plane
  - Repeat until convergence

Multi-pass plane sweep

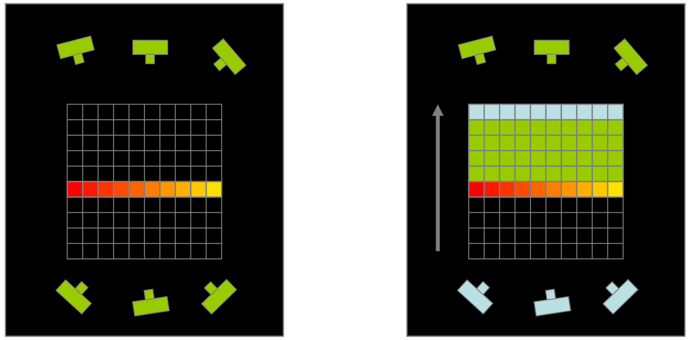


True Scene Reconstruction

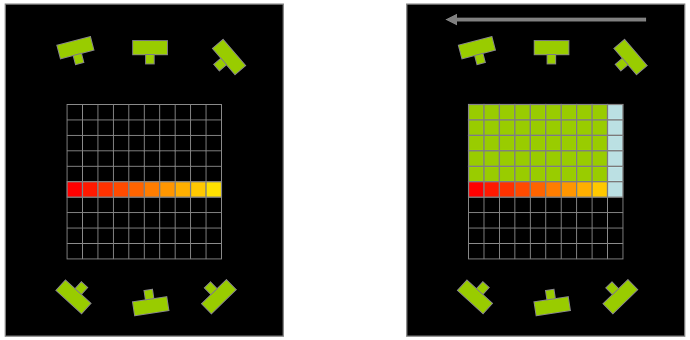
Multi-pass plane sweep



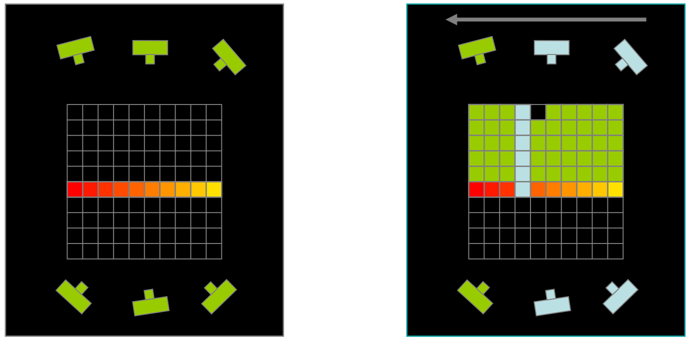
Multi-pass plane sweep



Multi-pass plane sweep



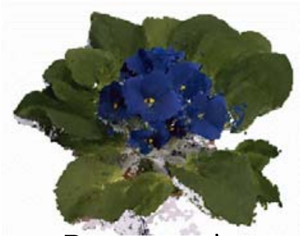
Multi-pass plane sweep



Space carving results: African violet

Input image (1 of 45)

Reconstruction



Reconstruction

Reconstruction

Space carving results: hand

Input image
(1 of 100)

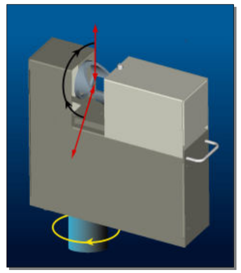
Reconstruction

Active approaches

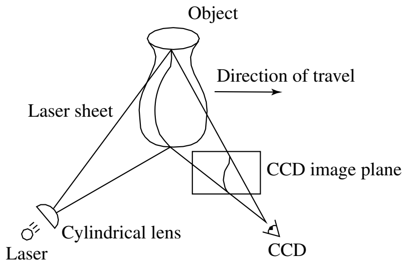
Time of flight

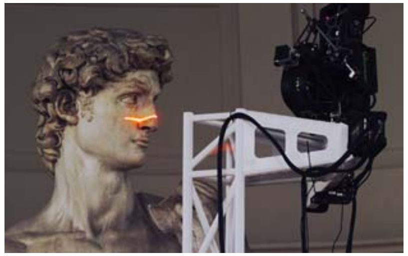
- Basic idea: send out pulse of light (usually laser), time how long it takes to return

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



Laser scanning (triangulation)

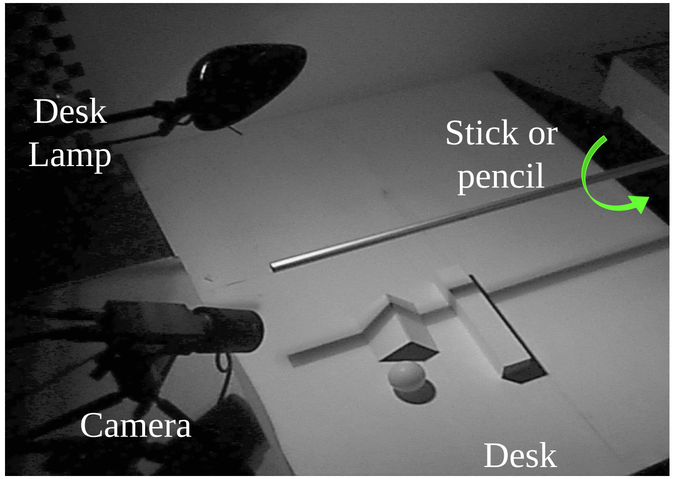




Digital Michelangelo Project
http://graphics.stanford.edu/projects/mich/

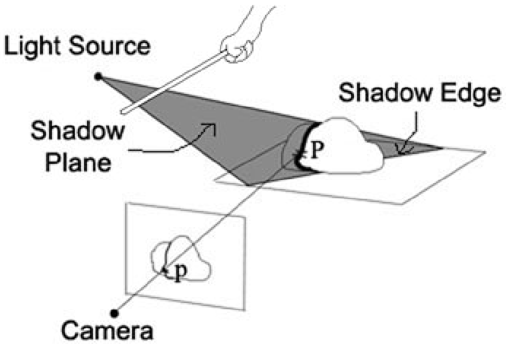
- Optical triangulation
  - Project a single stripe of laser light
  - Scan it across the surface of the object
  - This is a very precise version of structured light scanning
- Other patterns are possible

Shadow scanning



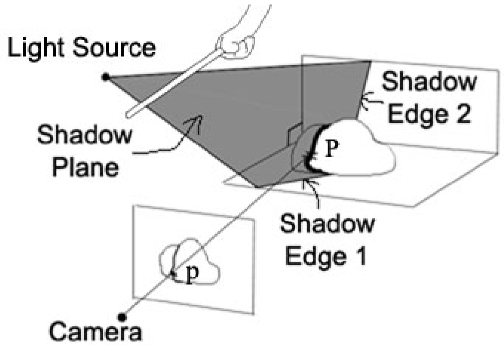
http://www.vision.caltech.edu/bouguetj/ICCV98/

Basic idea



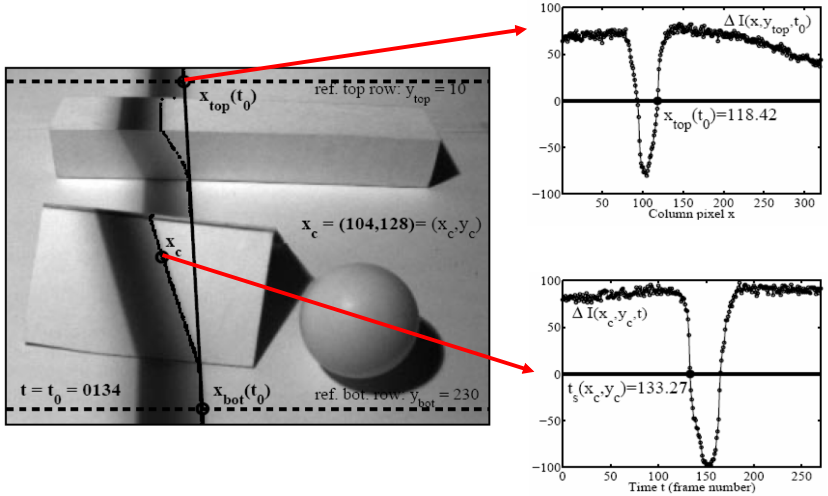
- Calibration issues:
  - where's the camera wrt. ground plane?
  - where's the shadow plane?
    - depends on light source position, shadow edge

Two Plane Version



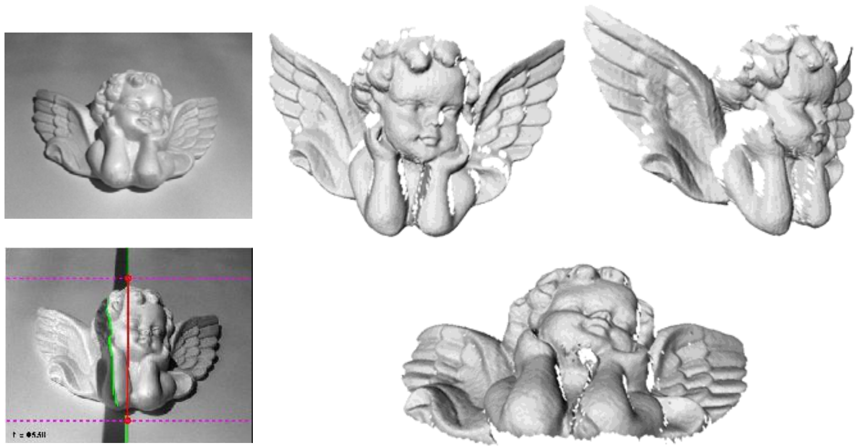
- Advantages
  - don't need to pre-calibrate the light source
  - shadow plane determined from two shadow edges

Estimating shadow lines



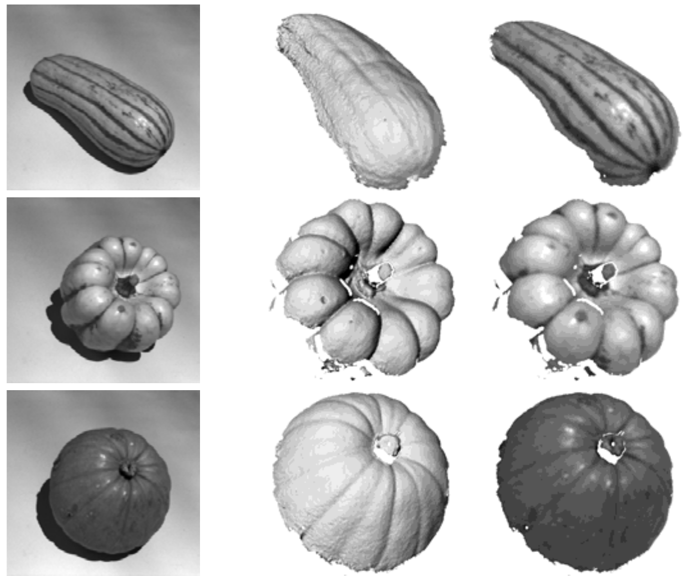
Shadow scanning in action

Results

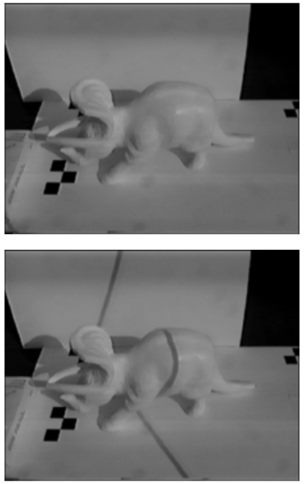


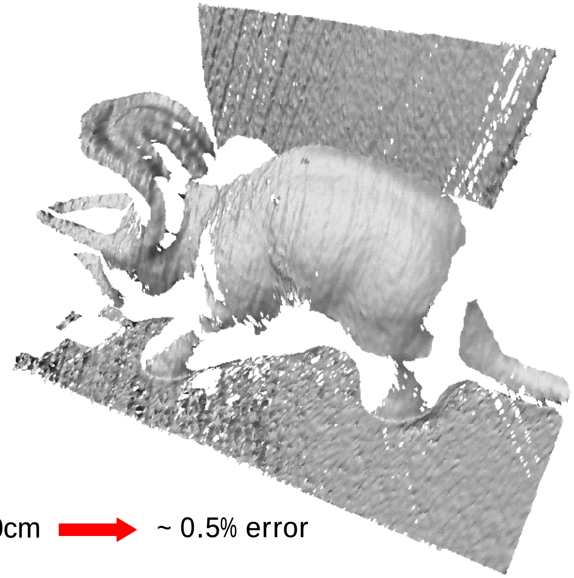
accuracy: 0.1mm over 10cm → ~ 0.1% error

Textured objects



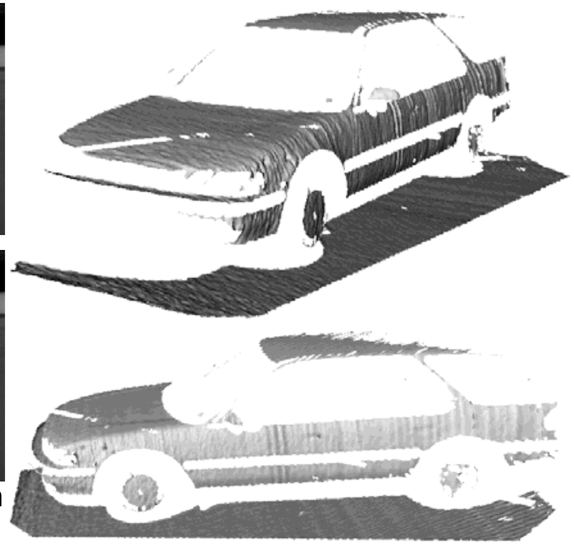
Scanning with the sun





accuracy: 1mm over 50cm → ~ 0.5% error

Scanning with the sun



accuracy: 1cm over 2m → ~ 0.5% error

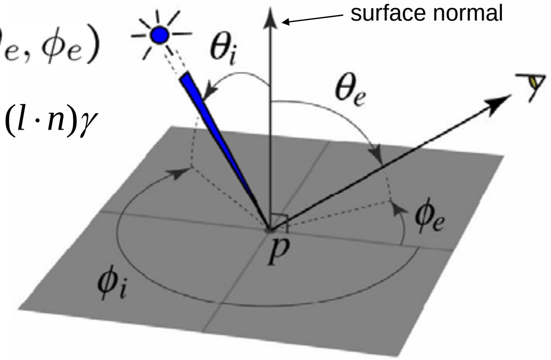
Active variants of passive approaches

The BRDF

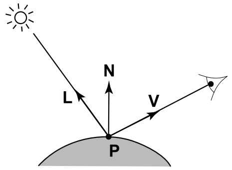
- The Bidirectional Reflection Distribution Function
  - Given an incoming ray (θ_i, ϕ_i) and outgoing ray (θ_e, ϕ_e) what proportion of the incoming light is reflected along outgoing ray?

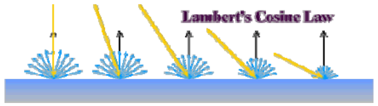
$$\rho(\theta_i, \phi_i, \theta_e, \phi_e)$$

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



Diffuse reflection (Lambertian)



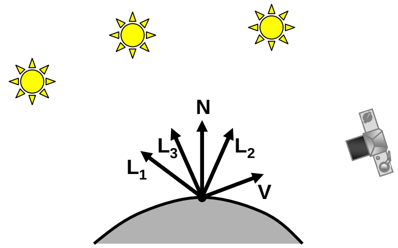


$\rho(l,v) = k_d$ ← **albedo**

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

Assuming that light strength is 1.

Photometric stereo



$$\begin{aligned} I_1 &= k_d \mathbf{N} \cdot \mathbf{L_1} \\ I_2 &= k_d \mathbf{N} \cdot \mathbf{L_2} \\ I_3 &= k_d \mathbf{N} \cdot \mathbf{L_3} \end{aligned}$$

- 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

$$\underbrace{\begin{bmatrix} I_1 \\ I_2 \\ I_3 \end{bmatrix}}_{\substack{\mathbf{I} \\ 3\times 1}} = \underbrace{\begin{bmatrix} \mathbf{L_1}^T \\ \mathbf{L_2}^T \\ \mathbf{L_3}^T \end{bmatrix}}_{\substack{\mathbf{L} \\ 3\times 3}} \underbrace{k_d \mathbf{N}}_{\substack{\mathbf{G} \\ 3\times 1}}$$

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

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

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

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:

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

- Solve for N, k_d as before

Trick for handling shadows

- Weight each equation by the pixel brightness:

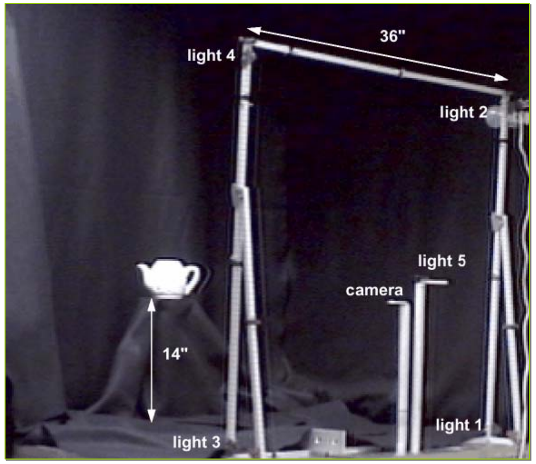
$$I_i(I_i) = I_i[k_d \mathbf{N} \cdot \mathbf{L_i}]$$

  - 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 N, k_d as before

Photometric Stereo Setup



Procedure

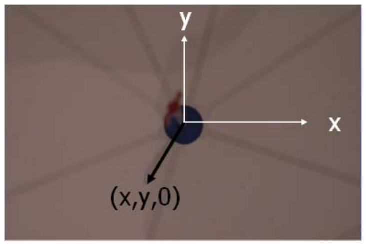
- Calibrate camera
- Calibrate light directions/intensities
- Photographing objects (HDR recommended)
- Estimate normals
- Estimate depth

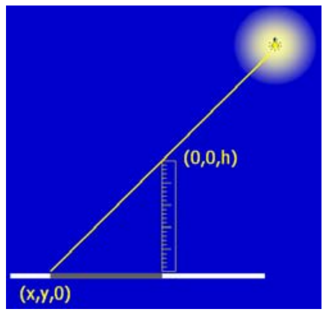
Estimating light directions

- Trick: place a chrome sphere in the scene

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

- Use a ruler

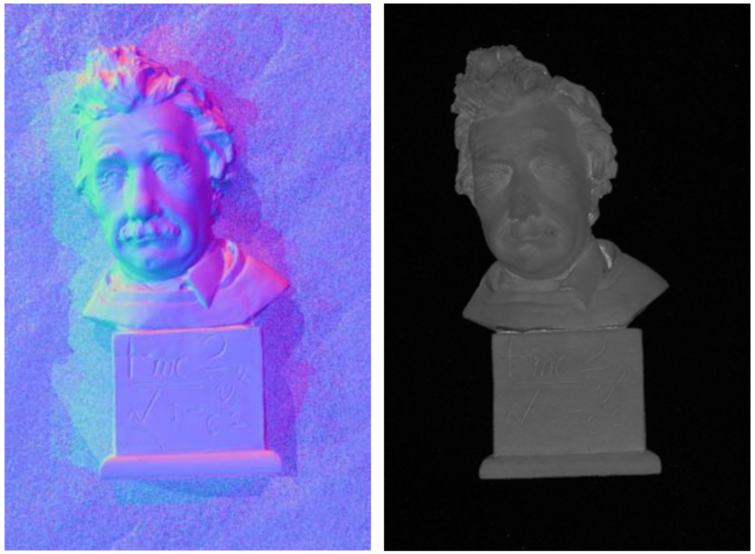




Photographing objects

Normalize light intensities

Estimate normals



Depth from normals

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

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

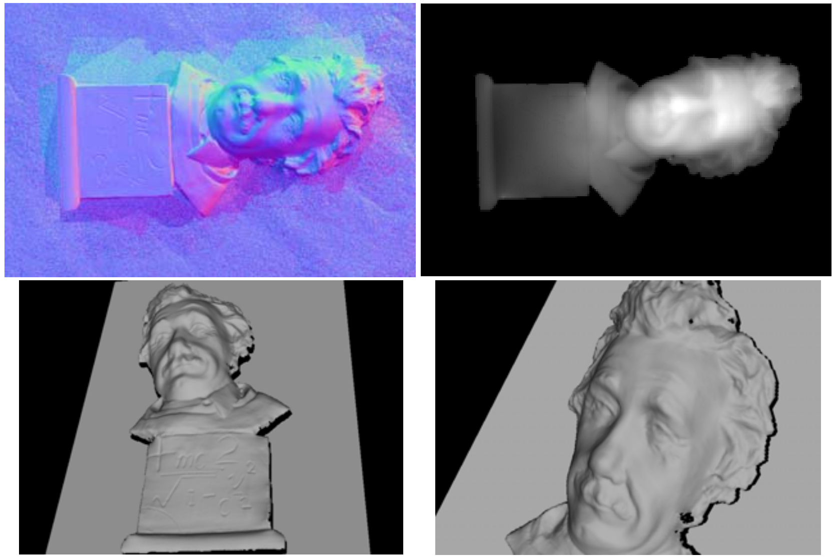
$$= \sum_{i,j} w_{data} * \left[\left(\frac{\partial z(i,j)}{\partial x} - p_{ij} \right)^2 + \left(\frac{\partial z(i,j)}{\partial y} - q_{ij} \right)^2 \right]$$

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

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

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

Results

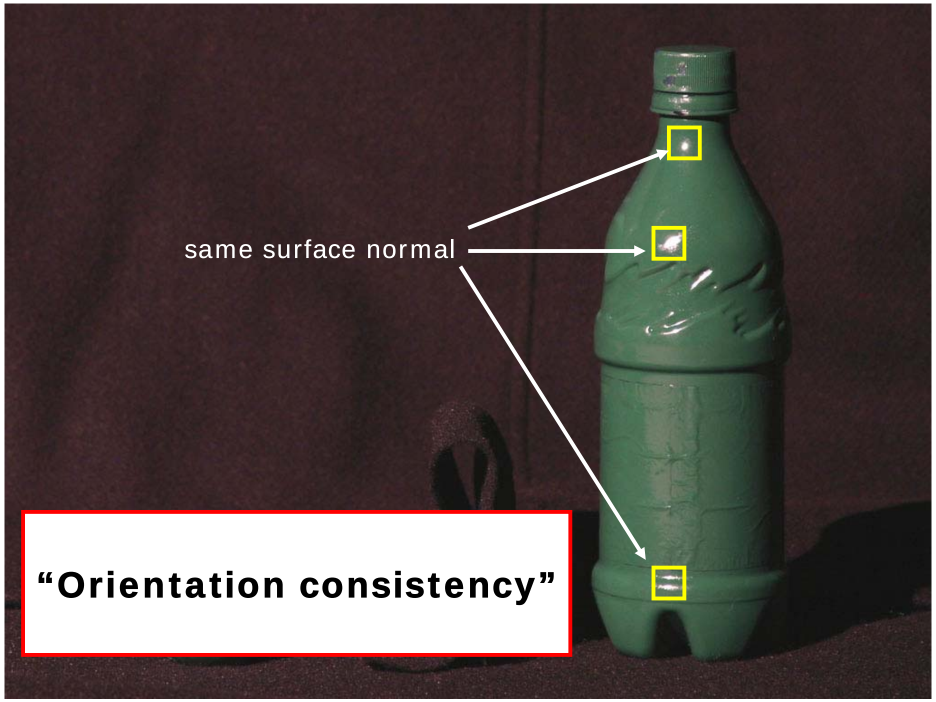


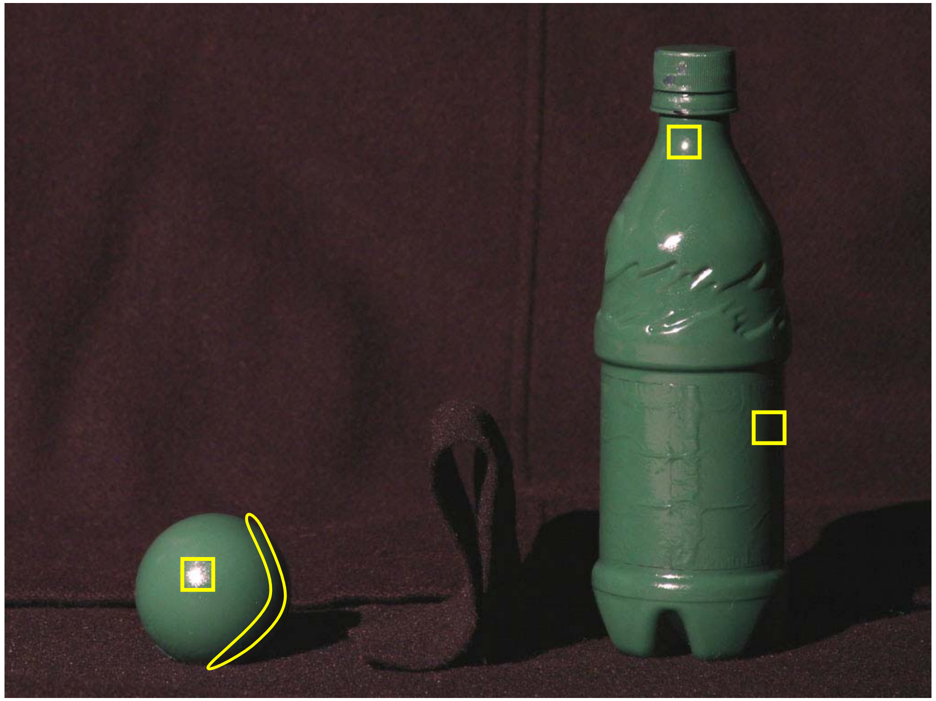
Limitations

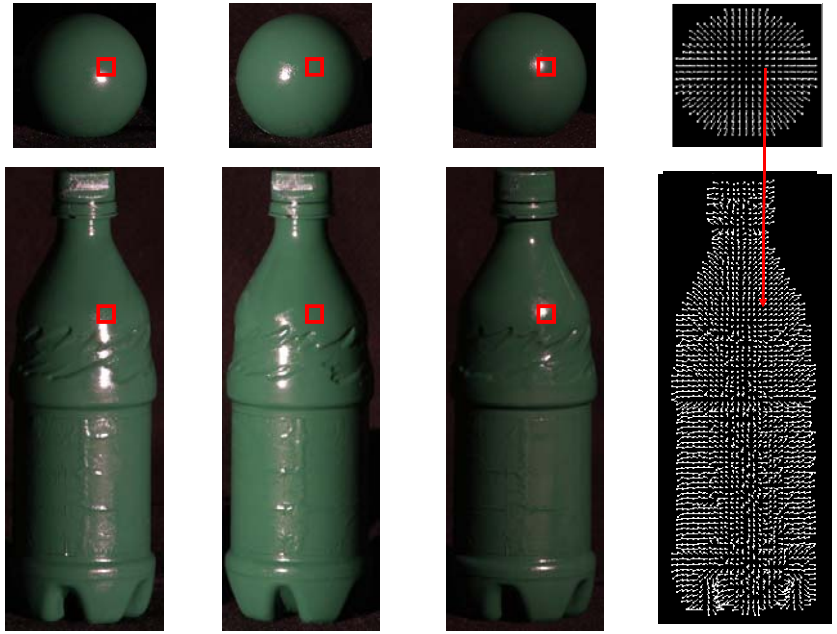
- Big problems
  - doesn’t work for shiny things, semi-translucent things
  - shadows, inter-reflections
- Smaller problems
  - calibration requirements
    - measure light source directions, intensities
    - camera response function

Example-based photometric stereo

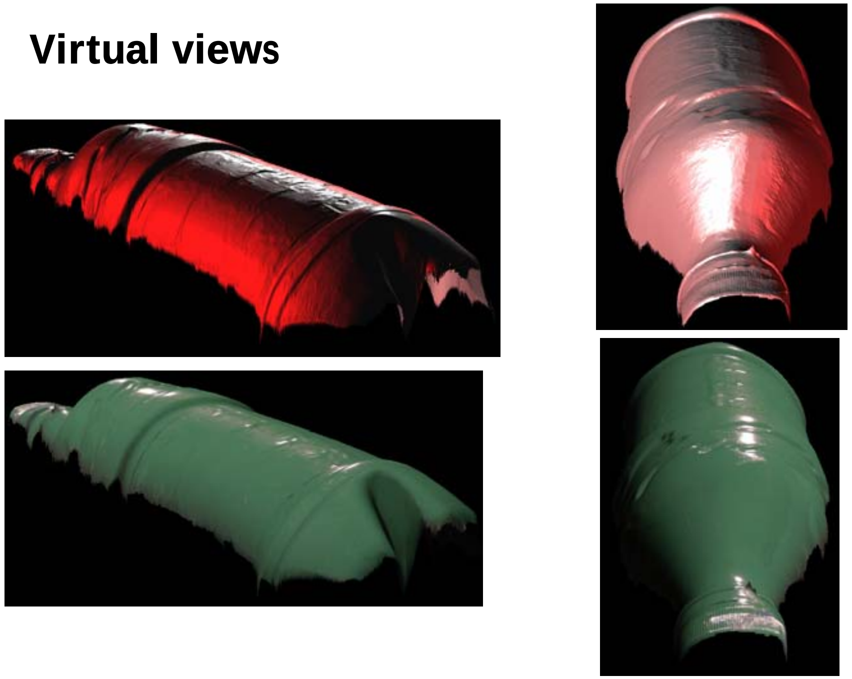
- Estimate 3D shape by varying illumination, fixed camera
- Operating conditions
  - any opaque material
  - distant camera, lighting
  - reference object available
  - no shadows, interreflections, transparency





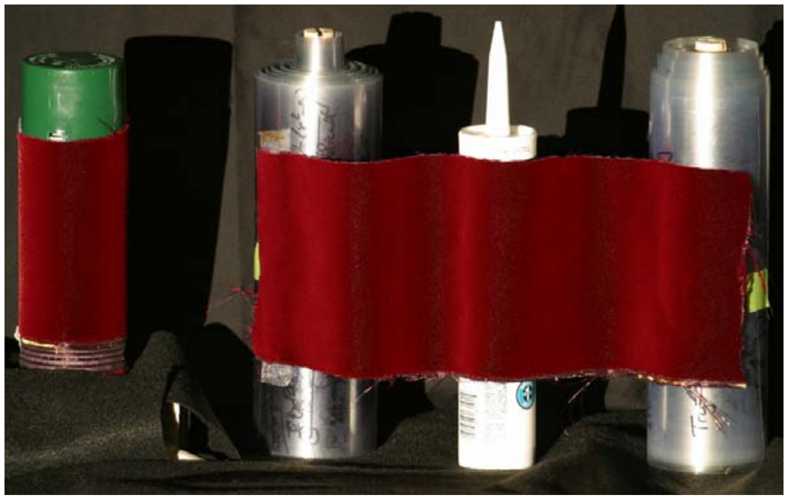



Virtual views

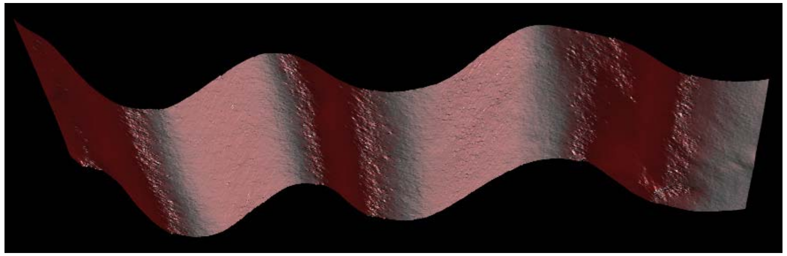


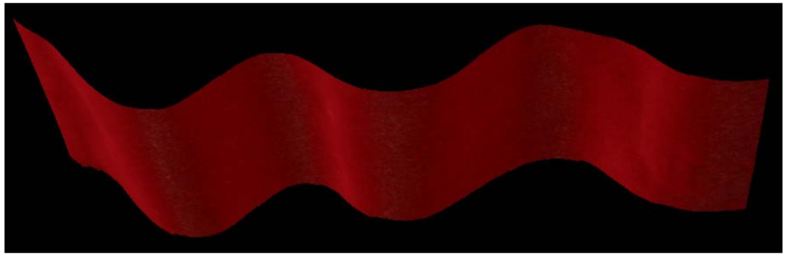
Velvet

DigiVFX

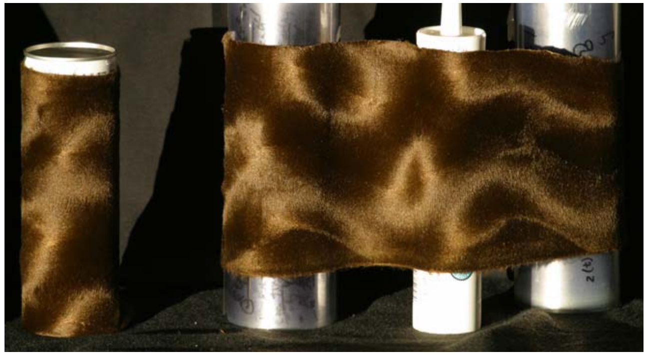


Virtual Views

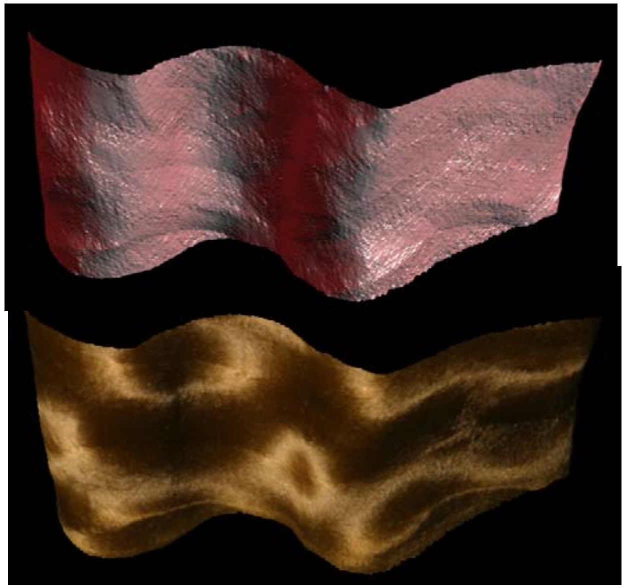


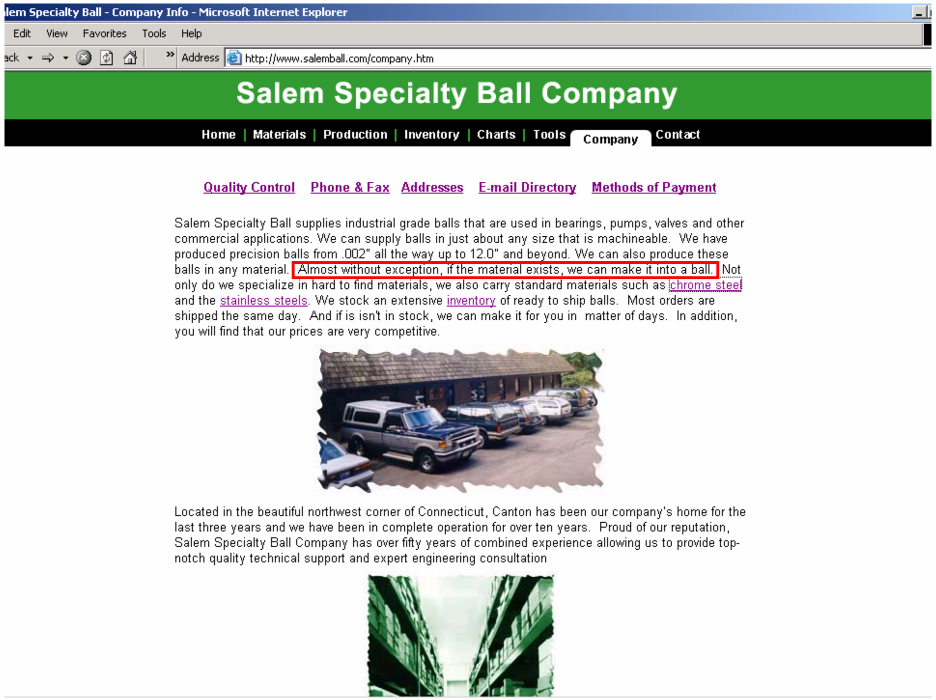


Brushed Fur



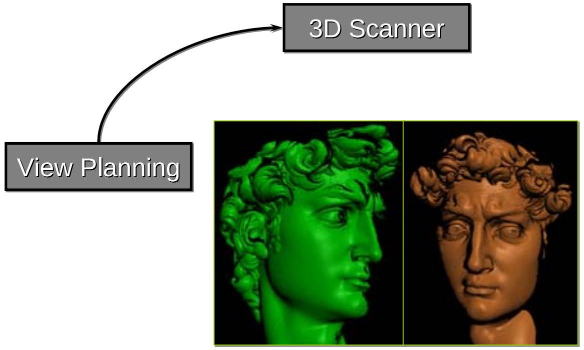
Virtual Views



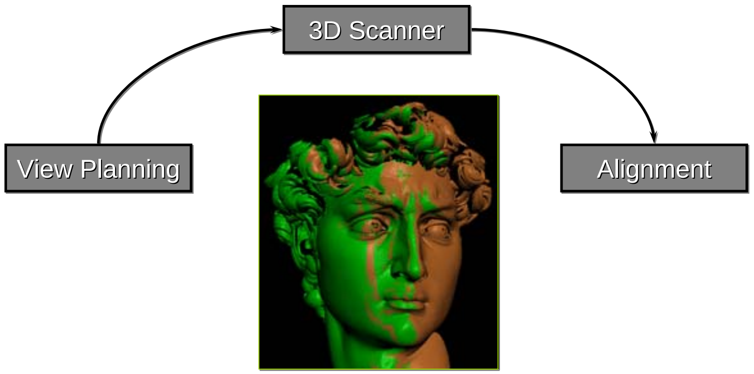


3D Model Acquisition Pipeline

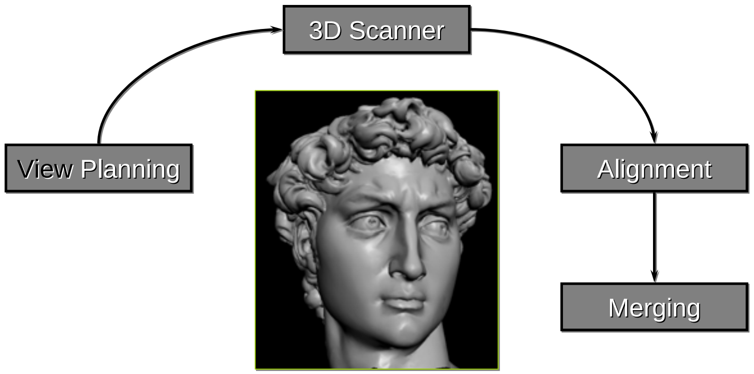
3D Model Acquisition Pipeline



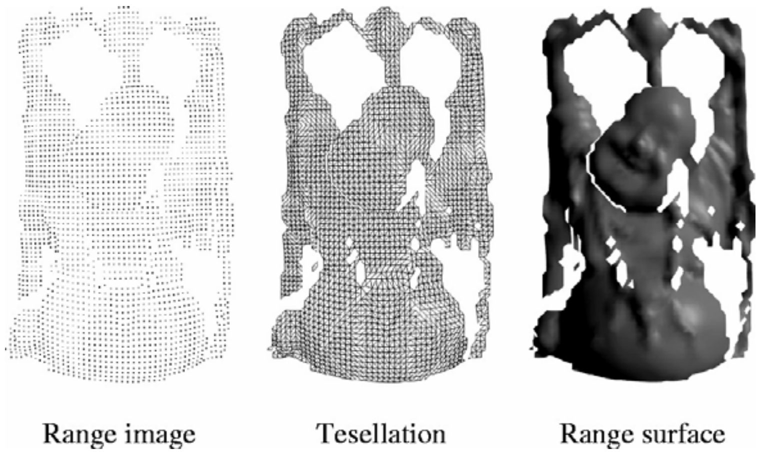
3D Model Acquisition Pipeline



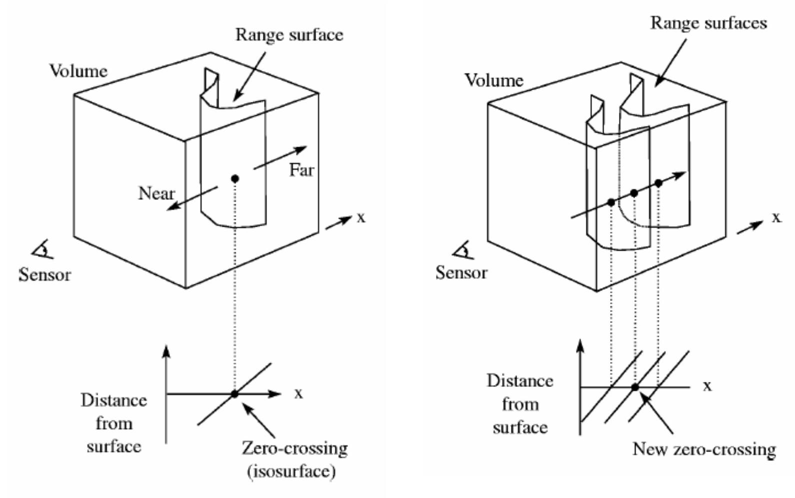
3D Model Acquisition Pipeline



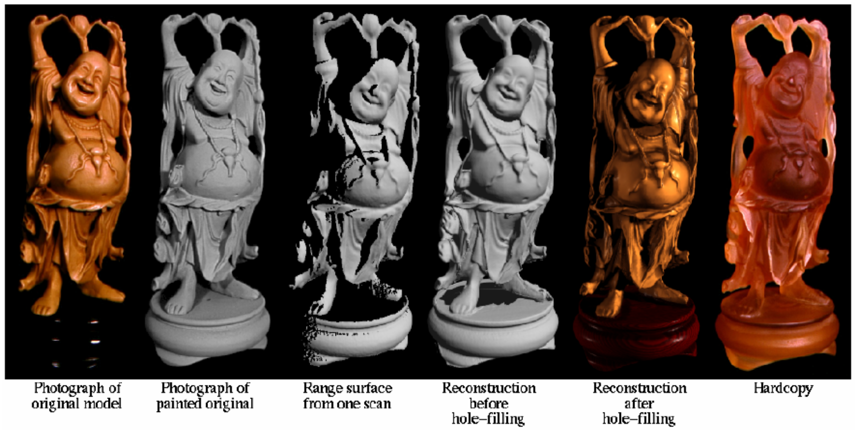
Volumetric reconstruction



Signed distance function



Results

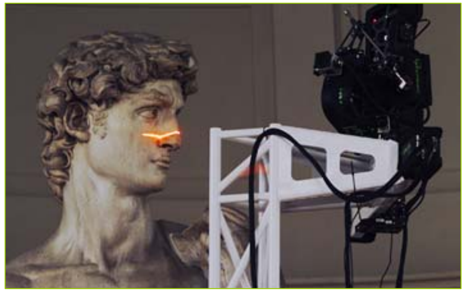


The Digital Michelangelo Project

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

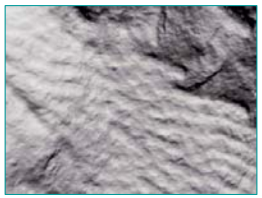
Systems, projects and applications

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

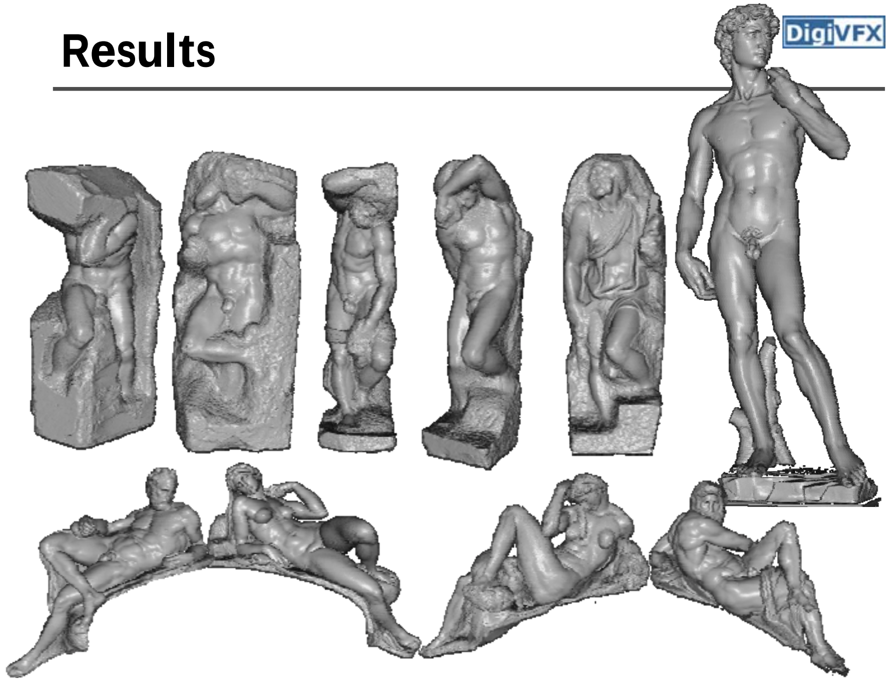
- 480 individually aimed scans
- 2 billion polygons
- 7,000 color images
- 32 gigabytes
- 30 nights of scanning
- 22 people

Comparison

photograph

1.0 mm computer model

Results



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

View interpolation

Bullet time video

View interpolation

High-quality video view interpolation

Final project

Final project

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

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 or CVPR 2006/2007 papers

Film

- It must be an “effect” film.
- You can use any tools as you want. But, I guess 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/
- Submit two videos, final and making-of.