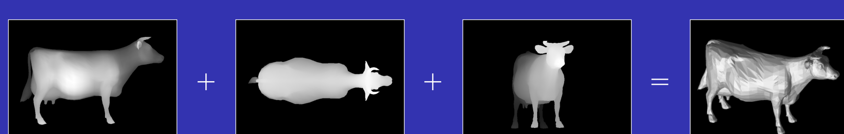
Efficient Estimation of 3D Euclidean Distance Fields from 2D Range Images

Sarah F. Frisken and Ronald N. Perry
Mitsubishi Electric Research Laboratories

Focus

- Of the many approaches for reconstructing 3D models from range data, we focus on
  - Implicit methods which
    - Compute a 3D distance field from the range data
    - Reconstruct the 3D model at an iso-surface of the distance field
  - Range data in the form of *range images* which
    - Exploits coherency between adjacent range values

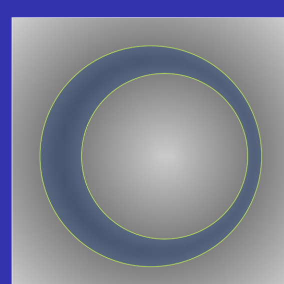


Our Contribution

- An efficient method for computing 3D distance fields from one or more 2D range images
  - Use Euclidean distance - faster, more accurate, less memory
    - Much of the prior art uses line-of-sight or projected distances
  - Perform most of the computation in a preprocessing step in the 2D coordinate space of each range image
    - Substantial reduction in computation
      - 10-100x faster than the prior art
  - Use Adaptively Sampled Distance Fields (ADFs)
    - Reduces distance evaluations and memory requirements

Distance Fields

- An object's distance field represents, for any point in space, the distance from that point to the object
- The distance can be signed to distinguish between the inside and outside of the object
- The metric used to measure distance can take many forms, but minimum Euclidean distance is common



History of Distance Fields

- Distance fields are a specific example of implicit functions (see Bloomenthal 1997)
- Distance fields have many applications
  - CAD/CAM
    - Ricci 1973, Rockwood 1989, Breen 1990, Schroeder et al. 1994, Perry and Frisken 2001
  - Medical imaging and surgical simulation
    - Blum 1973, Raya and Udupa 1990, Payne and Toga 1990, Jones and Chen 1995, Szeliski and Lavalle 1996, Frisken-Gibson 1999
  - Modeling deformation and animating deformable models
    - Bloomenthal and Wyville 1990, Bloomenthal and Shoemake 1991, Payne and Toga 1992, Gascuel 1993, Whitaker 1995, Sethian 1996, Cani-Gascuel 1998, DesBrun and Cani-Gascuel 1998, Breen 1998, Fisher and Lin 2001
  - Scan conversion or 'voxelization'
    - Payne and Toga 1992, Jones 1996, Gibson 1998, Sramek and Kaufman 1999
  - Robotics
    - (e.g., Koditschek 1989)

Representing Distance Fields

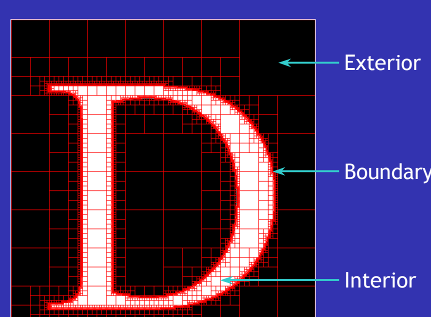
- Implicit representation
  - Distances computed at query points as needed
    - Precise but slow for complex models
- Sampled volumes
  - Distances computed and stored in a regular 3D grid
  - Interpolate distances at non-grid locations
  - Requires adequate sampling for alias free reconstruction of detailed models
    - Large memory requirements
    - Slow processing times

Sampled Volumes - Efficiency

- Exploit graphics hardware to compute distances
  - Hoff 1999, Hoff 2001
- Restrict distance computations to near the object surface (shell or narrow band methods)
  - Curless 1996, Jones 1996, Desbrun 1998, Whitaker 1998, Jones 2001, Zhao et al. 2001, Kimmel and Sethian 1996, Breen et al. 1998, and Fisher 2001
  - Can propagate distances outside the shell using
    - fast distance transforms
    - fast marching methods from level sets
- Use classic, or 3-color, octrees to reduce distance evaluations
  - Szeliski and Lavalle 1996, Wheeler 1998, and Strain 1999

3-Color Octrees

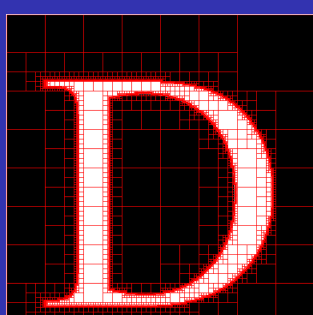
- A 3-color octree labels cells as interior, exterior, or boundary
- Boundary cells are always subdivided to the maximum level of the octree



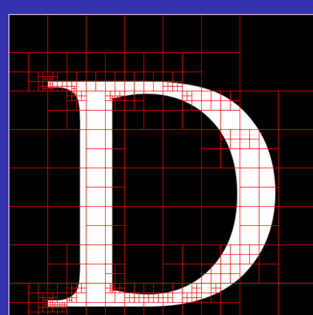
Adaptively Sampled Distance Fields

- Sample the distance field adaptively and store the distances in a spatial hierarchy (e.g., octree)
- Adaptive sampling is *detail-directed*
  - Sample the distance field according to local frequency content rather than whether or not a surface is present
    - Frisken et al. 2000, Perry and Frisken 2001
  - Substantially fewer distance evaluations and less memory requirements than a 3-color octree
- Provides high quality surfaces, efficient processing, and a reasonable memory footprint
  - A practical representation of distance fields

A 2D example



3-color quadtree



ADF

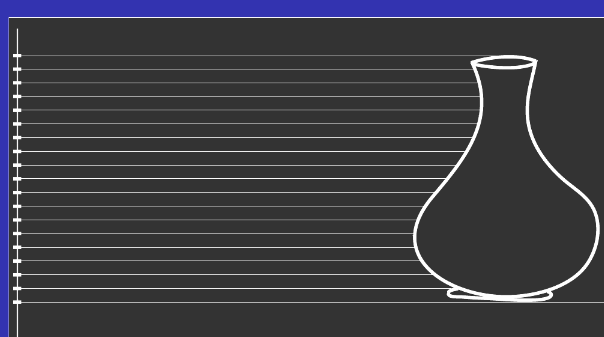
- 3-color quadtree - 20,813 cells
- Quadtree-based ADF using a bi-quadratic interpolant - 399 cells
- Equivalent regularly sampled volume (512x512) - 262,144 cells

Related Work

- Reconstructing 3D models using distance fields from
  - Unorganized surface points
    - Hoppe et al. 1992, Edelsbrunner 2002, Bajaj et al. 1995, Boissonnat and Cazals 2000, Carr et al. 2001
  - *Range surfaces*
    - Curless and Levoy 1996, Hilton et al. 1996, Wheeler et al. 1998, Perry and Frisken 2001, Sagawa et al. 2001
    - Weighted averaging to combine distances from multiple scans
    - Methods to
      - Compress the volume
      - Reduce the number of distance computations
    - Fill holes near occluded regions separately
  - *Range images*
    - Whitaker 1998, Zhao et al. 2001
    - Use line-of-sight distances
    - Use level set methods to reduce scanner noise

Geometry of Range Scanning

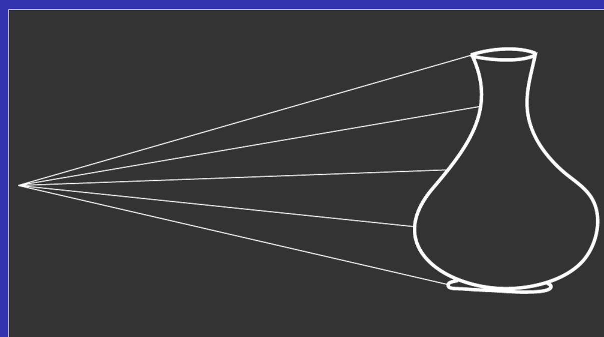
- Our algorithm requires range images composed of perpendicular projected distances



Perpendicular projected distances

Geometry of Range Scanning

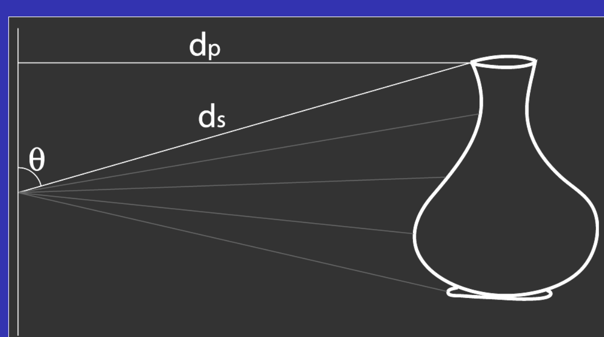
- Some scanning systems (e.g., laser striping) require conversion from line-of-sight distances to perpendicular projected distances



Line-of-sight distances

Geometry of Range Scanning

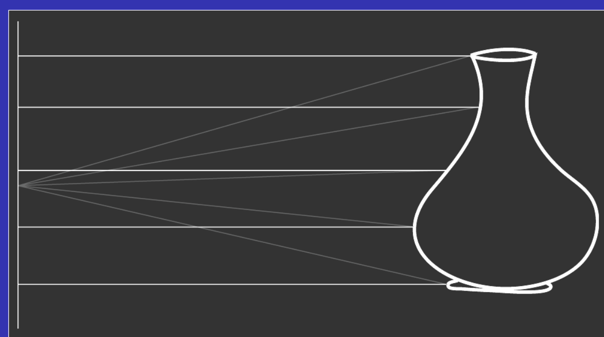
- Some scanning systems (e.g., laser striping) require conversion from line-of-sight distances to perpendicular projected distances



Use scanner geometry to derive perpendicular distances from line-of-sight distances

Geometry of Range Scanning

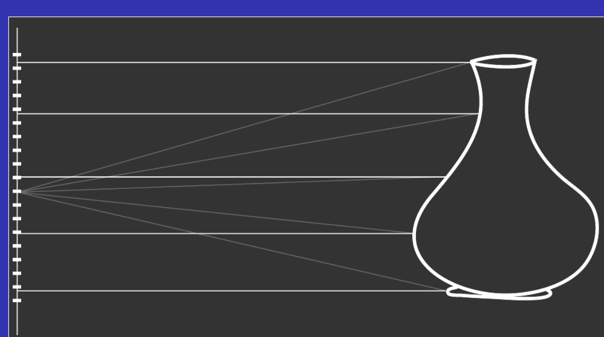
- Some scanning systems (e.g., laser striping) require conversion from line-of-sight distances to perpendicular projected distances



Back projection to the image plane

Geometry of Range Scanning

- Some scanning systems (e.g., laser striping) require conversion from line-of-sight distances to perpendicular projected distances



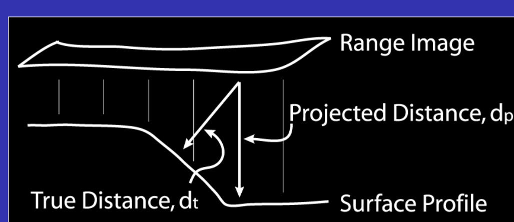
Resample the image plane

Euclidean vs. Non-Euclidean

- Range images provide line-of-sight or projected distances to the surface
  - Can be used directly to reconstruct the 3D model
  - e.g., Curless and Levoy 1996, and Whitaker 1998
- However
  - Line-of-sight and projected distances are not minimum Euclidean distances
    - Can introduce artifacts in the reconstructed surface
  - Euclidean distances can be exploited to provide
    - More efficient processing and memory usage

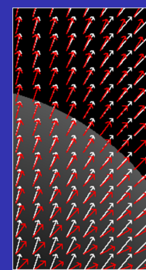
Range Data is Non-Euclidean

- Projected Distances
  - When the surface is at an angle to the scanning direction, the minimum Euclidean distance is smaller than the projected distance

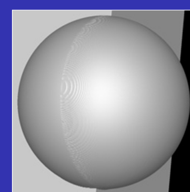


Range Data is Non-Euclidean

- Projected Distances
  - The projected and Euclidean distance fields have the same iso-surface but different gradient fields
    - Problematic for methods that use the gradient to evolve a surface towards the zero-value iso-surface

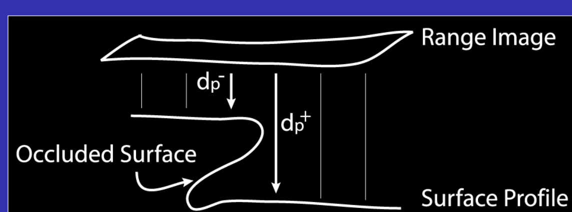


  - Artifacts arise when combining multiple scans using windowed, weighted, averaging



Range Data is Non-Euclidean

- Cliffs and Occlusions
  - Projected distances in the range image are discontinuous near cliffs and occlusions
    - Produces abutting large positive and large negative distances along the cliff face, resulting in excessive ADF cell subdivision near cliffs



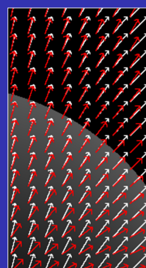
Why Euclidean Distances?

- Accuracy
  - Off-surface gradient points to the closest surface point
  - Fewer artifacts when multiple scans are combined using windowed weighted averaging
- Efficiency
  - Cell size and distance values can be used to terminate adaptive subdivision of interior and exterior cells
    - Faster generation of the ADF (and hence the model)
    - Better than 10x fewer distance evaluations
    - Significant reduction in temporary storage
  - Eliminate distance field discontinuities near cliffs
    - Smaller ADF

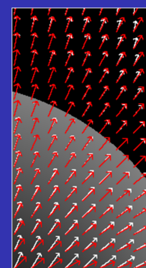
Correcting Projected Distances

- Approach
  - Near planar surfaces, projected distance is related to minimum Euclidean distance according to
    - $d_t = d_p * \cos(\theta) = d_p \,/\, |\nabla(d_p)|$
  - *Correct* the projected distance field near relatively planar regions of the surface by dividing the projected distance by the magnitude of the local gradient of the projected distance field

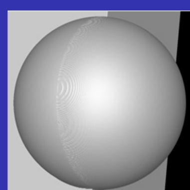
Correcting Projected Distances



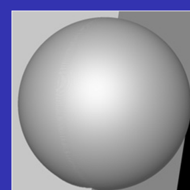
Gradient field before gradient magnitude correction



Gradient field after gradient magnitude correction



Artifacts when combining multiple scans without gradient magnitude correction



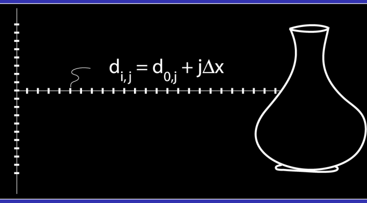
Artifact free with gradient magnitude correction

Correcting Projected Distances

- Need to compute the gradient magnitude at each sample point during ADF generation
  - Easy for a regularly sampled distance volume, BUT…
  - Requires several additional distance evaluations for each sample point in the ADF that may not otherwise be needed (e.g., 6 additional distance evaluations when using central differences)
    - Reduces generation speed significantly

Correcting Projected Distances

- Observation
  - The projected distance decreases at a constant rate along rays perpendicular to the range image
    - The gradient of the projected distance field is constant along these rays



$$2Dgrad_{i,j} = (d_{i+1,j} - d_{i-1,j}, d_{i,j+1} - d_{i,j-1})$$
$$= (2\Delta x, d_{0,j+1} - d_{0,j-1})$$
$$|2Dgrad_{i,j}| = (4\Delta x^2 + |1Dgrad_j|)^{0.5}$$

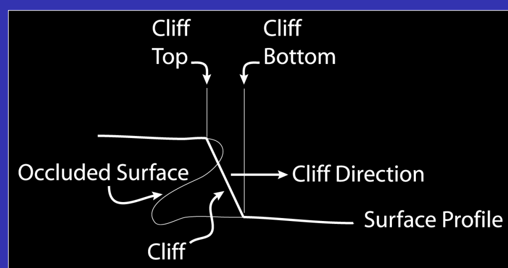
  - The gradient of the 3D projected distance field can be represented by a 2D field in the plane of the range image

Correcting Projected Distances

- Method
  - Pre-compute a 2D gradient magnitude correction image in the plane of the range image
  - For each sample point during ADF generation
    - Interpolate the 2D range image to compute a projected distance
    - Interpolate the 2D gradient magnitude correction image to derive the gradient magnitude
    - Divide the projected distance by the gradient magnitude to approximate the Euclidean distance

Correcting Distances Near Cliffs

- Approach
  - Assume that the surface forms a continuous “cliff” across a range image discontinuity
    - Eliminates holes in the reconstructed surface
    - Provides a reasonable guess at unobserved regions of the surface



Correcting Distances Near Cliffs

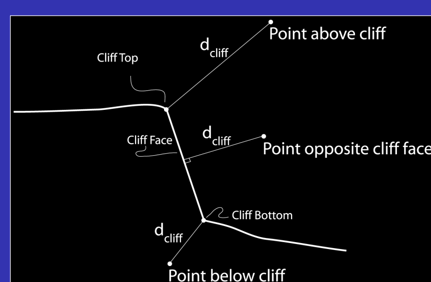
- Approach (continued)
  - Locate the nearest cliff for each sample point and choose the smaller of the gradient magnitude corrected distance and the distance to the cliff
    - When combining multiple scans, favor distances computed from range images with better views of an occluded region over cliff distances

Correcting Distances Near Cliffs

- Computing cliff distances requires searching each range image for the closest cliff in 3D space
  - Too slow even if we
    - Locate *cliff pixels* adjacent to discontinuities in the range image in a pre-processing step,
    - Bin cliff pixels in a spatial hierarchy, AND
    - Use fast search techniques

Correcting Distances Near Cliffs

- Observation
  - Cliff distances can be computed from the horizontal distance to the cliff and the vertical distance to the cliff top or bottom



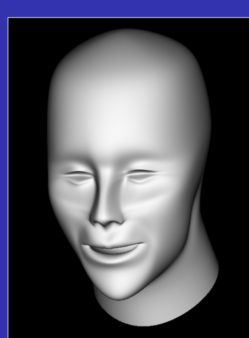
  - The horizontal distances can be pre-computed from the range image and stored in an annotated 2D image, or *cliffmap*, which also encodes the heights of the top and bottom of the nearest cliff

Correcting Distances Near Cliffs

- Approach
  - Create a 2D cliffmap for the range image in a preprocessing step
  - During ADF generation
    - Interpolate the cliffmap to determine the horizontal and vertical distances to the top and bottom of the nearest cliff
    - Compute the cliff distance from the interpolated values

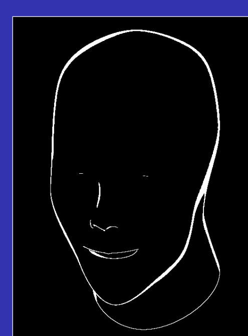
Correcting Distances Near Cliffs

- Approach – Creating the cliffmap
  - Step 1: Detect pixels at the tops and bottoms of each cliff



3D object

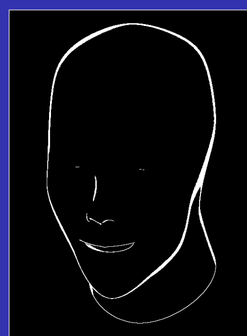
Range image



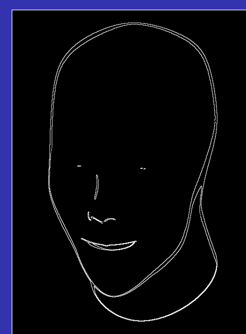
Cliff pixels

Correcting Distances Near Cliffs

- Approach – Creating the cliffmap
  - Step 2: Combine adjacent cliff pixels to form multi-pixel wide cliffs



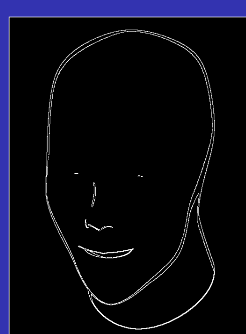
Cliff pixels



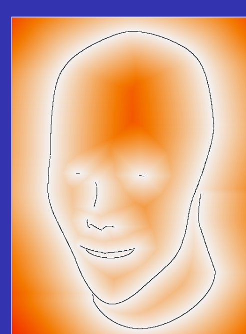
Multi-pixel wide cliffs

Correcting Distances Near Cliffs

- Approach – Creating the cliffmap
  - Step 3: Compute unsigned 2D distances to cliffs using a 2D Euclidean distance transform



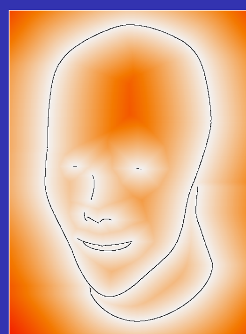
Multi-pixel wide cliffs



Unsigned 2D distances

Correcting Distances Near Cliffs

- Approach – Creating the cliffmap
  - Step 4: Derive signed 2D distances to cliffs by negating distances on the outward facing side of cliffs



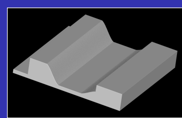
Unsigned 2D distances

Signed 2D distances

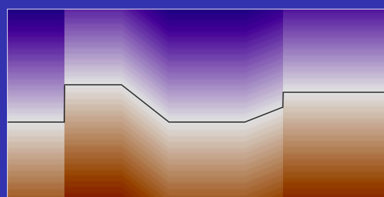
Summary of Correction

- To estimate the 3D Euclidean distance directly from a 2D range image
  - Compute the projected distance at ***p*** from the range image
  - Correct the projected distance using the gradient magnitude at ***p*** interpolated from the pre-computed 2D gradient magnitude correction image
  - Compute the distance to the nearest cliff using the pre-computed 2D cliffmap
  - Choose the smaller of the corrected projected distance and the cliff distance

Effect of Correction



3D model

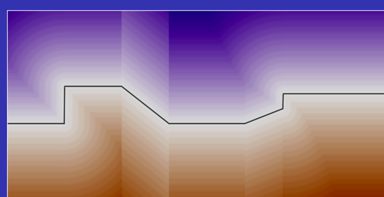


Projected distance field - cross section

Black line represents surface

Discontinuities where the surface is vertical

Compression of the field where the surface is at an angle to the scanning direction



Corrected distance field

Smooth and uniformly dense all along the surface and better approximates the Euclidean distance field

Combining Multiple Range Images

- Prior art uses weighted averaging for robust treatment of noise and image alignment error
  - Curless and Levoy 1996, Hilton et al. 1996, Wheeler et al. 1998, and Whitaker 1998
- The results in this paper use a simple combining scheme that favors
  - Corrected projected distances over cliff distances
  - Corrected projected distances with the
    - Smallest gradient magnitude correction
    - Smallest absolute value

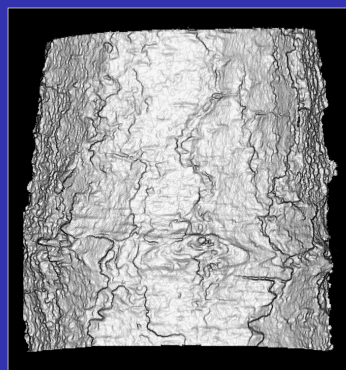
Summary of the Algorithm

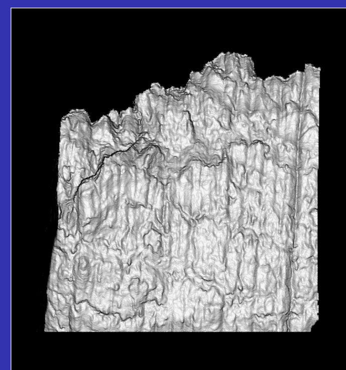
- If necessary, convert line-of-sight range images to perpendicular projected distances
- Pre-compute gradient magnitude images
- Pre-compute cliffmaps
- Generate an octree-based ADF of the Euclidean distance field where
  - Distances are computed via the correction method
  - The simple combining scheme is used to choose the best distance from multiple range images

Results

- Timings measured on a 1GHz Pentium IV processor
- Timings include
  - Pre-computation of correction images
  - ADF generation
  - Rendering
- ADF resolution reported in equivalent volume size
  - Level 9 (2^9) ADF has a resolution of 512^3
  - Level 10 (2^{10}) ADF has a resolution of 1024^3

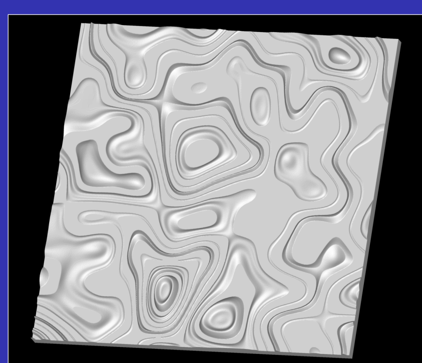
Cyberware Echo Data (Single Scan)

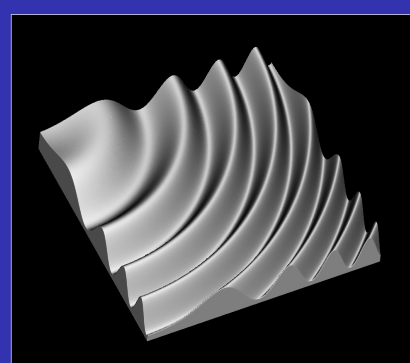




	Resolution	Time (seconds)
Oak bark (left)	512 x 512 x 512	3.85
	1024 x 1024 x 1024	12.5
Redwood bark (right)	512 x 512 x 512	2.94
	1024 x 1024 x 1024	10.2

Synthetic Range Data (Single Scan)





	Resolution	Time (seconds)
Pond ripples (left)	512 x 512 x 512	2.9
	1024 x 1024 x 1024	9.3
Waves (right)	512 x 512 x 512	5.5
	1024 x 1024 x 1024	8.9

Synthetic Range Data (z-buffer)

Resolution	1 scan	3 scans	6 scans	10 scans
512 x 512 x 512	7 secs	18 secs	36 secs	63 secs
1024 x 1024 x 1024	28 secs	68 secs	139 secs	214 secs

Comparison with Prior Art

- Whitaker 1998
  - 20 minutes for 10 range images for a 140 x 140 x 140 volume using a Sparc 10 workstation
  - Timing is for the full reconstruction but “most of that time was spent on the initialization and resampling” of the distance volume
  - Volumes larger than 140 x 140 x 140 caused thrashing

Comparison with Prior Art

- Wheeler et al. 1998
  - 52 minutes for 48 range images using an SGI Indy 5000
  - Used a 3-color octree equivalent in resolution to a 128 x 128 x 128 volume

Comparison with Prior Art

- Curless and Levoy 1996
  - 197 minutes for 61 range images on a 712 x 501 x 322 volume
  - 259 minutes for 71 range images on a 407 x 957 x 407 volume
  - 250 MHz MIPS R4400 processor

Comparison with Prior Art

- Our algorithm
  - ~1 second per range image for an ADF equivalent in resolution to a 256 x 256 x 256 volume
  - 3 to 7 seconds per range image for an ADF equivalent in resolution to a 512 x 512 x 512 volume
  - 9 to 28 seconds per range image for an ADF equivalent in resolution to a 1024 x 1024 x 1024 volume
  - Times are $kO(N)$, $k < 1$, for N range images
- These timings and resolutions compare very favorably with the prior art

Future work

- Add probabilistic weighting functions for combining multiple scans
- Extend the approach to permit incremental model updating with each new scan
  - Display confidence in distance measures to guide interactive determination of the next-best-view

Acknowledgments

- Gene Sexton from Cyberware for range data
- The Digital Michelangelo Project at Stanford University for providing a triangle model of Michelangelo’s David