A Volumetric Method for Building Complex Models from Range Images

Contribution

Given a set of aligned, dense range images, provides a manifold closely approximating the original model with several *desirable properties*

Desirable Properties

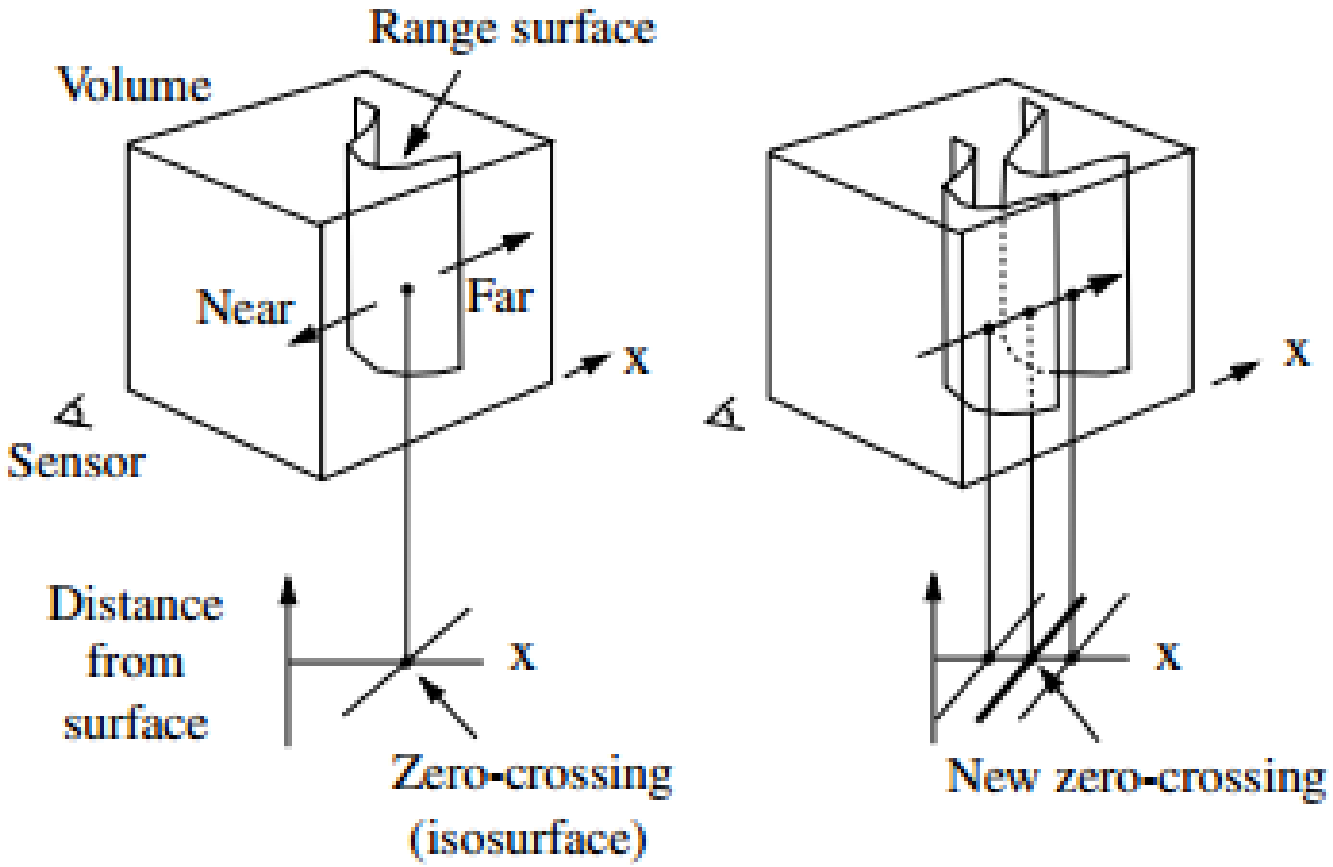
- Representation of range uncertainty
- Utilization of all input data
- Incremental, order-independent updating
- Water-tight reconstruction
- Robust
- Efficient performance
- No restriction on model topology

Overview

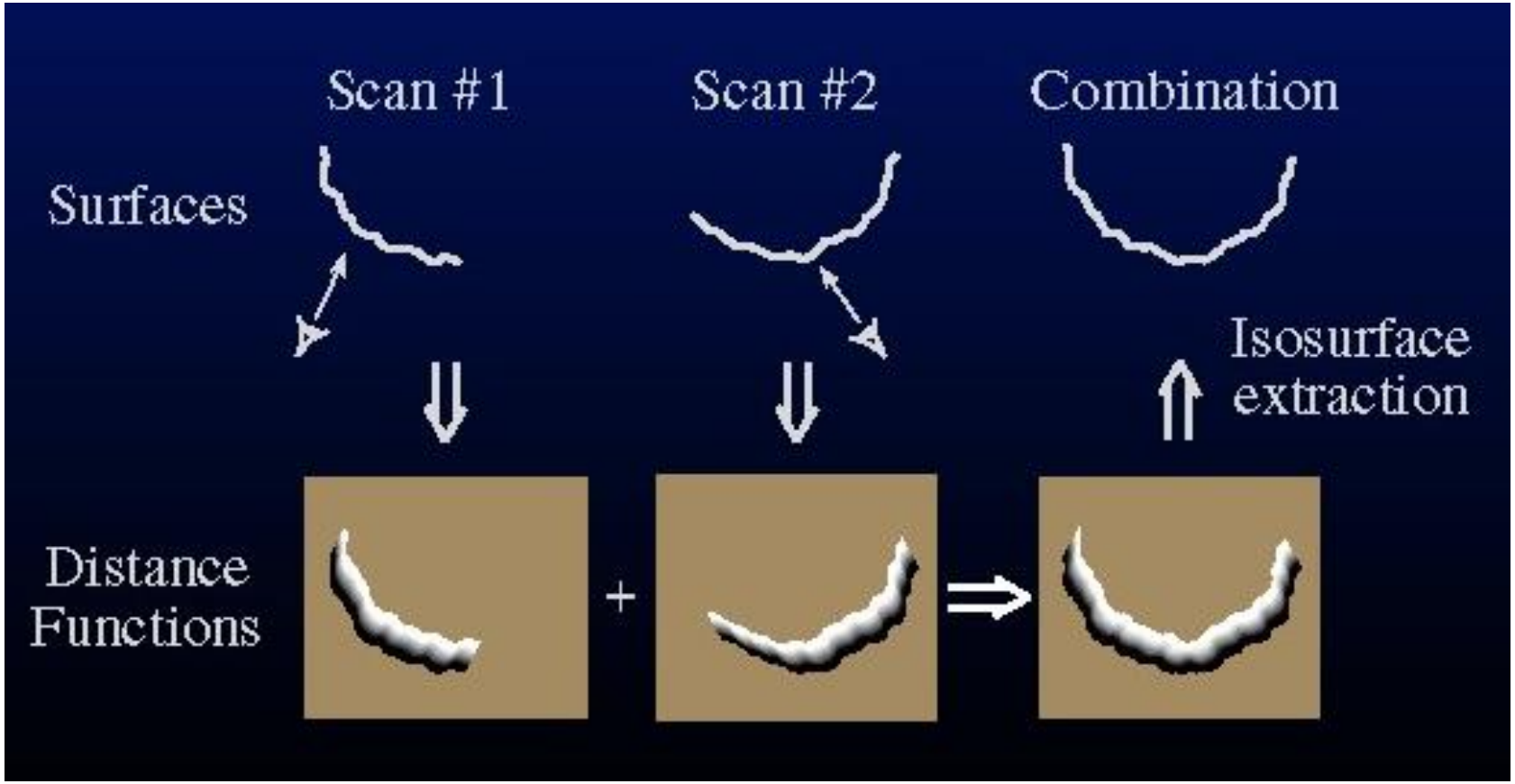
- Given set of range images: $R_1, R_2, \ldots R_{N-1}, R_N$
- Create signed distance functions:

  $d_1(x), d_2(x), \ldots, d_{N-1}(x), d_N(x)$
- Compute the cumulative distance function D(x) from these functions
- Extract the manifold as the isosurface D(x) = 0

Generating d(x) → Computing D(x)



Computing D(x)

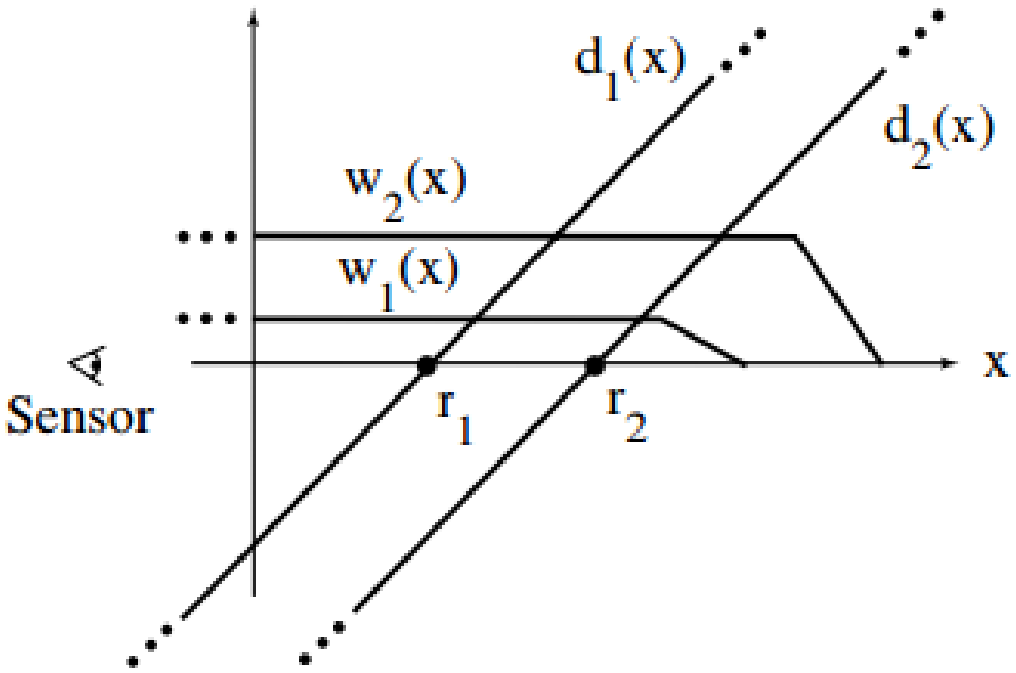


Computing D(x)

- Distance functions: $d_1(x)$, $d_2(x)$, …, $d_{N-1}(x)$, $d_N(x)$
  - Generated by following line of sight from sensor to reconstructed range surface
- Weights: $w_1(x)$, $w_2(x)$, …, $w_{N-1}(x)$, $w_N(x)$

  Measure of certainty of correctness of range estimate
  - $w_i(x)$ = Dot(vertex normal, line of sight)
  - Adjusted to taper off as distance from the line of sight increases

Computing D(x)



Computing D(x)

- Computation

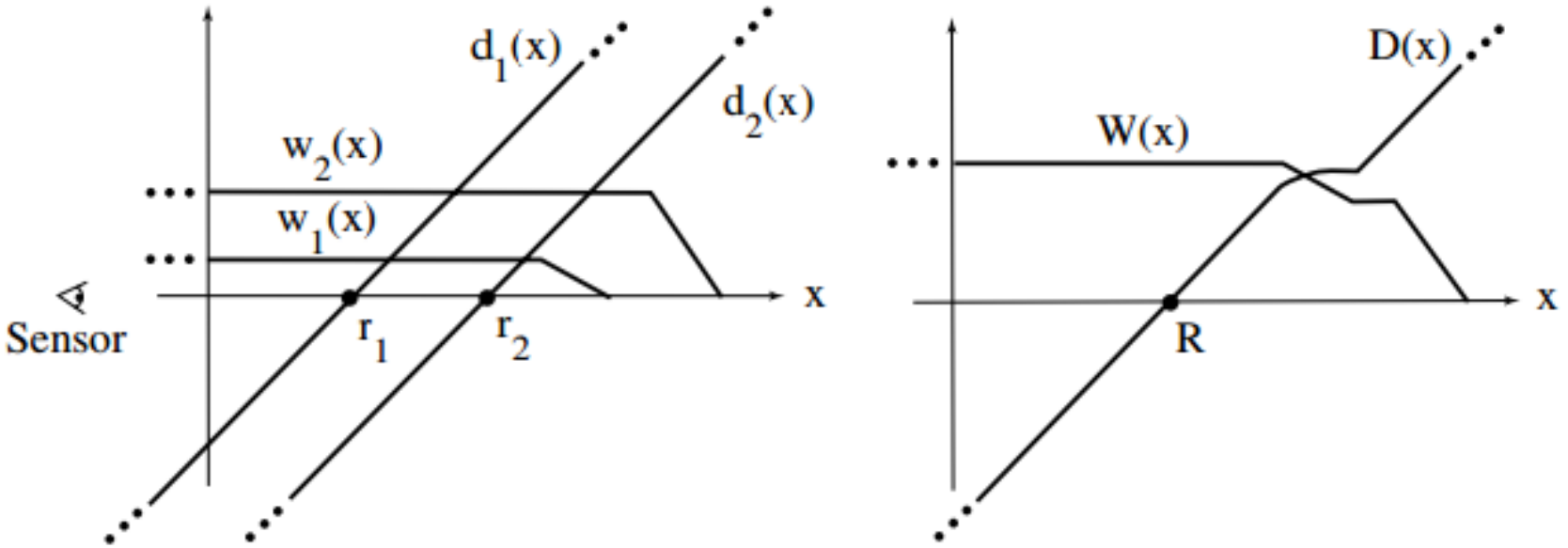
$$W(x) = \sum w_i(x),\ D(x) = \frac{\sum d_i(x) w_i(x)}{\sum w_i(x)}$$

- Can be performed incrementally, orderlessly

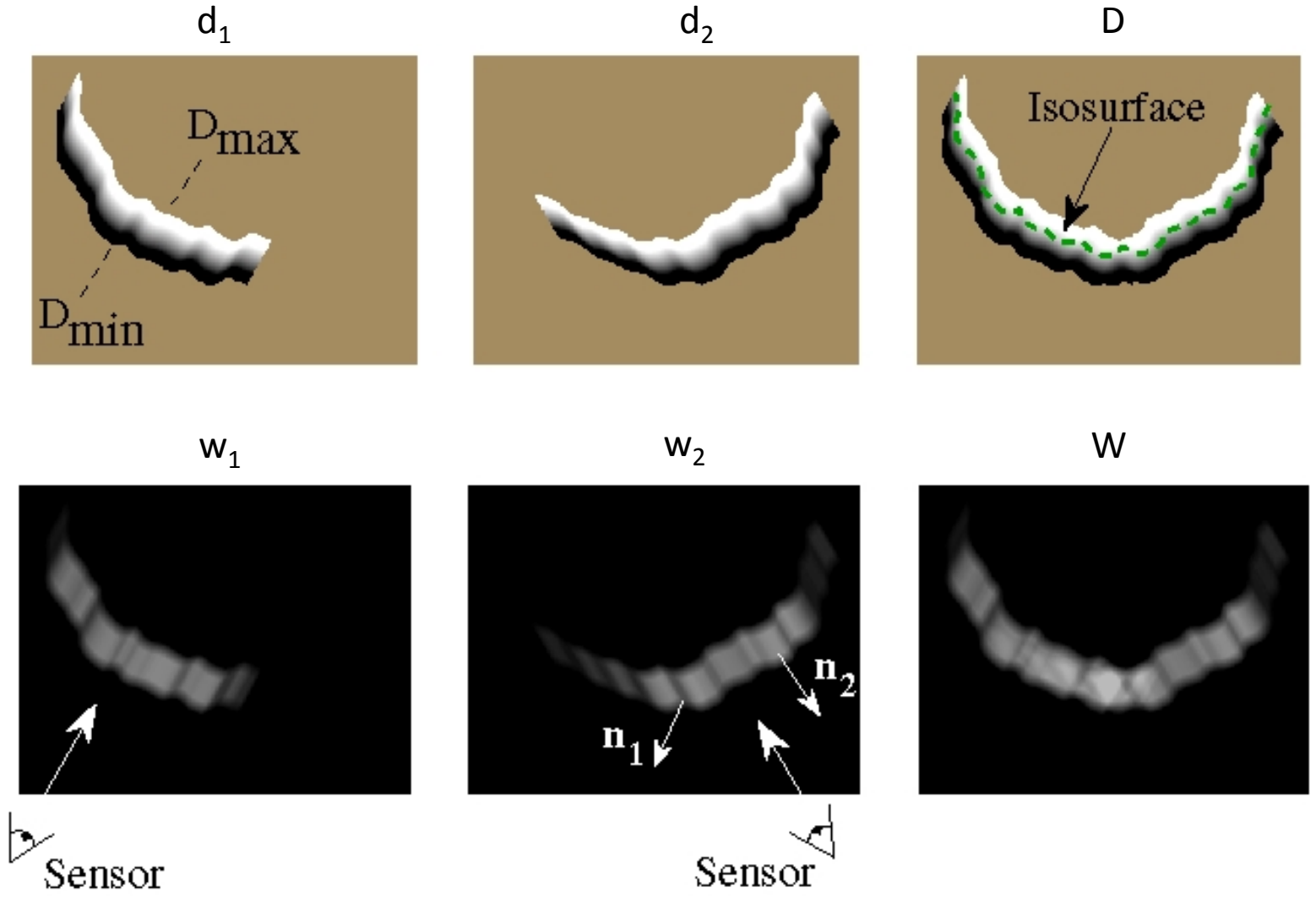
$$W_i(x) = W_{i-1}(x) + w_i(x)$$

$$D(x) = \frac{d_i(x) w_i(x) + D_{i-1}(x)}{w_i(x) + W_{i-1}(x)}$$

Computing D(x)



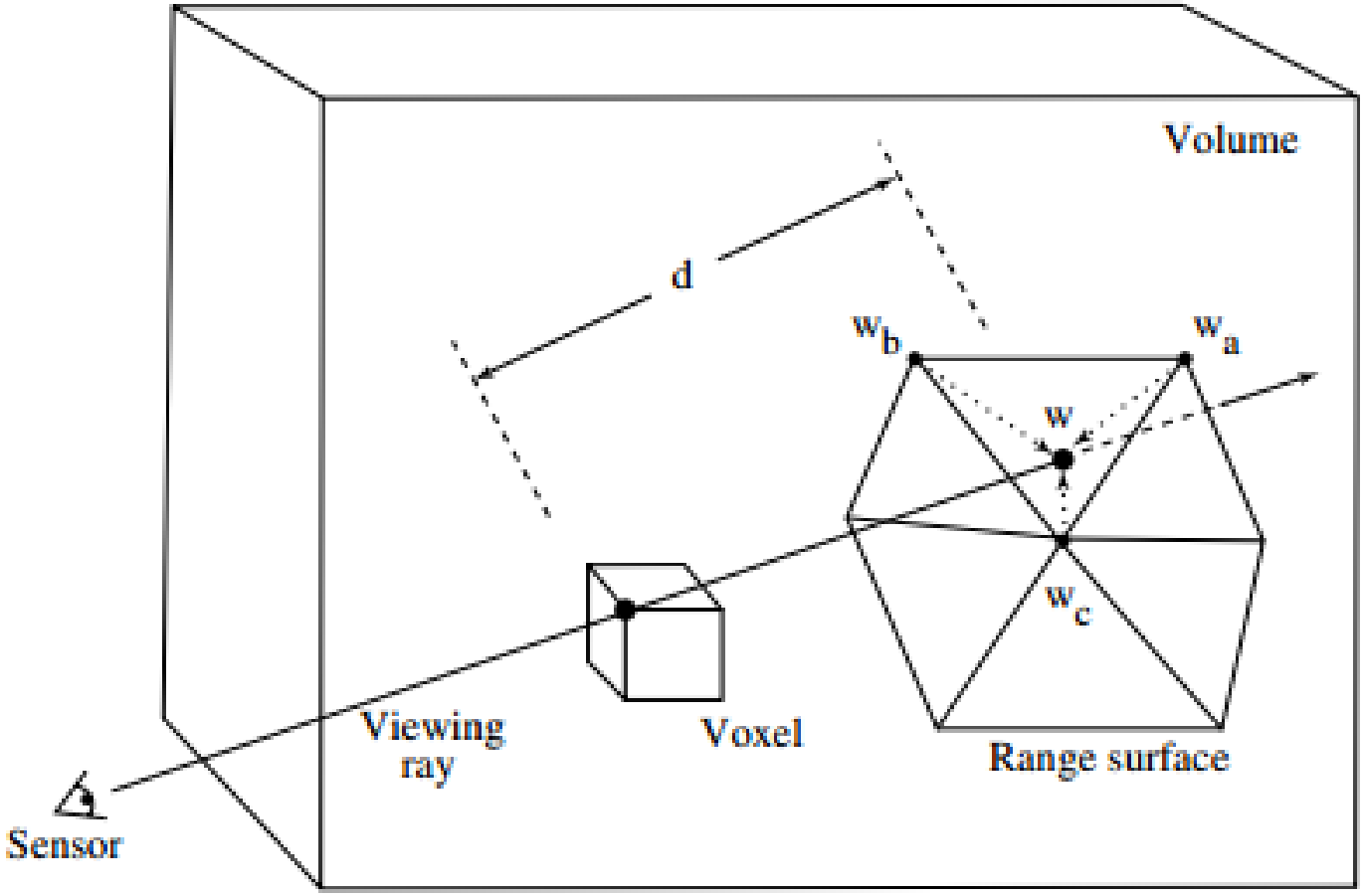
Computing D(x)



Algorithm So Far

- Set all voxel weights to zero
- For each range image R_i
  - Obtain its triangle mesh
    - Weight = dot(vertex normal, line of sight) at each sample vertex
    - Signed distance at each voxel

Determining distance contribution



Algorithm So Far

- Set all voxel weights to zero
- For each range image R_i
  - Obtain its triangle mesh
    - Sample vertex weight = dot(vertex normal, line of sight)
    - At each voxel
      - Signed distance = distance along viewing ray from voxel to range surface
      - Weight = interpolation of vertex weights of triangle intersected by viewing ray
  - Compute D(x) as described
  - Isosurface extraction: D(x) = 0
    - Ignore remaining zero weight, i.e. unseen regions

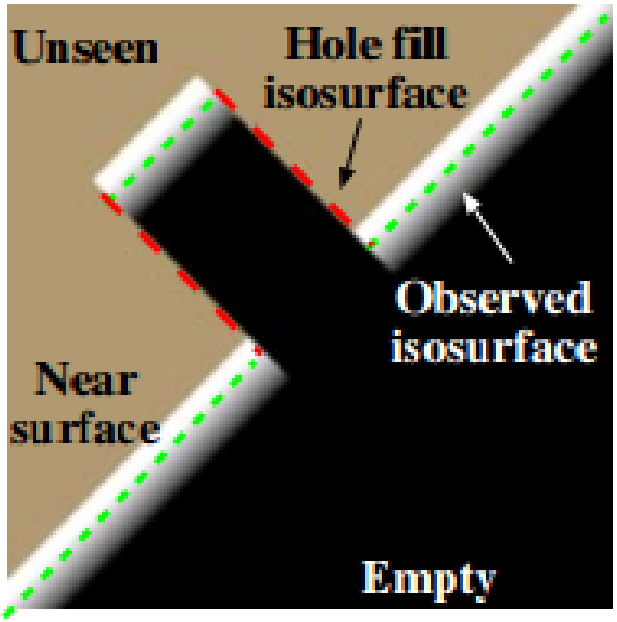
Hole filling

- Unseen portions of the surface appear as holes in the current reconstruction
- Direct filling in polygonal model
  - Does not use all available information
  - Difficult to obtain robust result
- Solution provided: *Space carving*

Algorithm with Space Carving

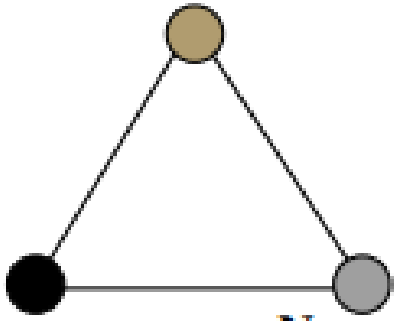
- Initialize all voxels to the “unseen” state
- Update near range surface voxel distance and weight values as before
- Space carving:
  - Follow the lines of sight back from the observed surface
  - Mark the corresponding voxels as “empty”
- Isosurface extraction: $D(x) = 0$
- Fill holes: Create a surface between regions marked “empty” and regions marked “unseen”

Algorithm with Space Carving



$D(\mathbf{x}) = D_{max}$

$W(\mathbf{x}) = 0$



$D(\mathbf{x}) = D_{min}$

$W(\mathbf{x}) = 0$

$D_{min} < D(\mathbf{x}) < D_{max}$

$W(\mathbf{x}) > 0$

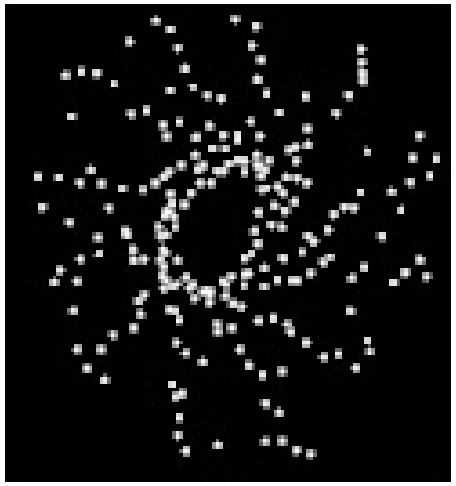
Implementation and Usage

- Software optimizations
  - Run-length encoded data structures
  - Memory coherent traversal
  - Binary depth trees
  - Restricted Marching cubes
- Typical data size
  - 60 scans
  - 10 million input vertices
  - 100 million voxels

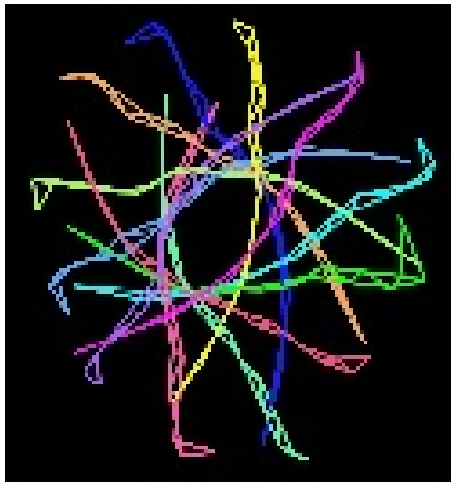
Results: Robustness

Merging ranged images of a 1.6mm drill bit from 12 orientations at a 30 degree spacing

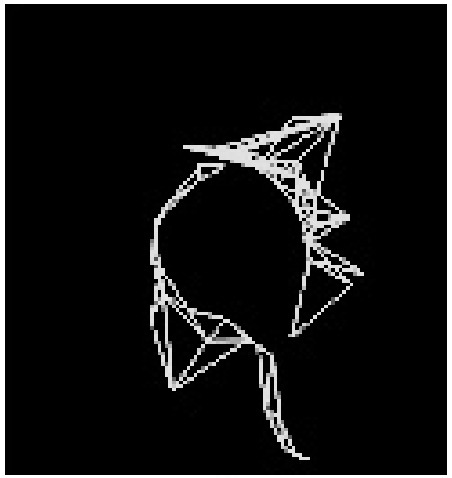
(a) Unorganized points
(b) Wire frame tessellations
(c) Slice generated by polygonal method
(d) Slice generated by volumetric method



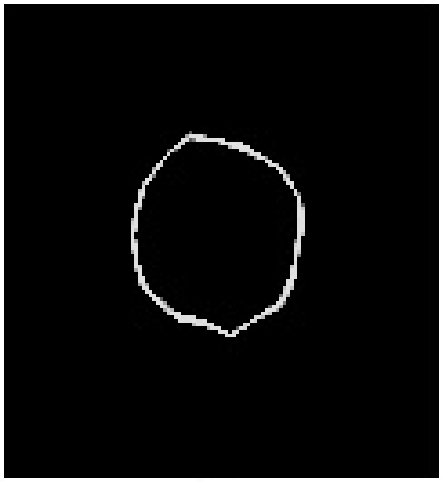
(a)



(b)



(c)

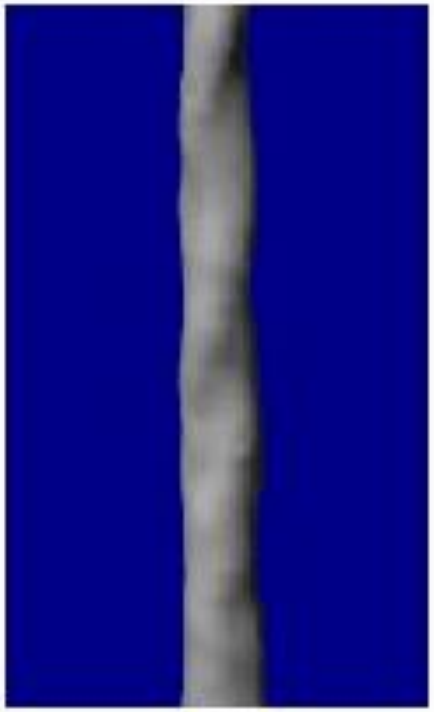


(d)

Results: Robustness

(e)

Polygonal method rendering



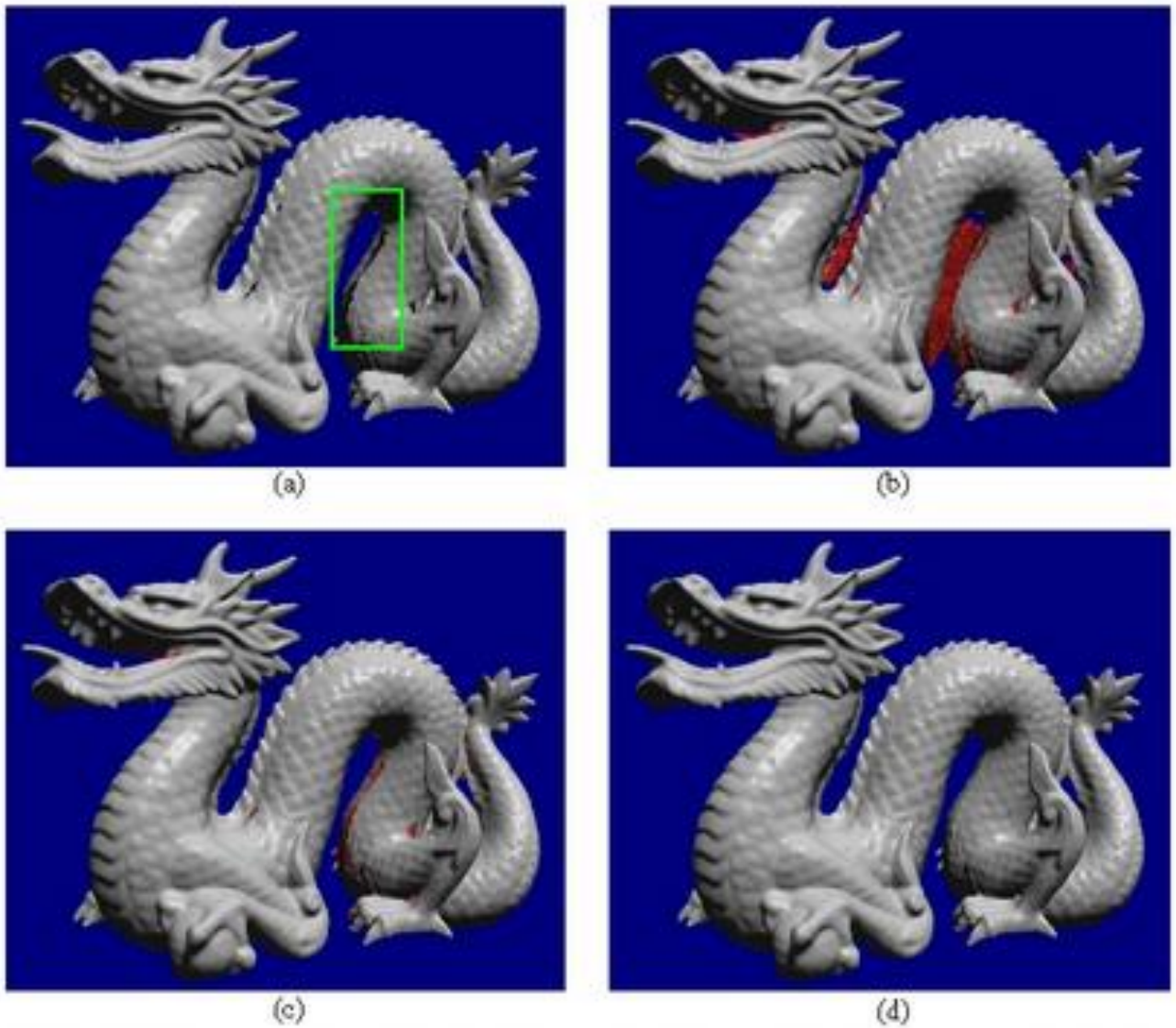
(f)

Volumetric method rendering

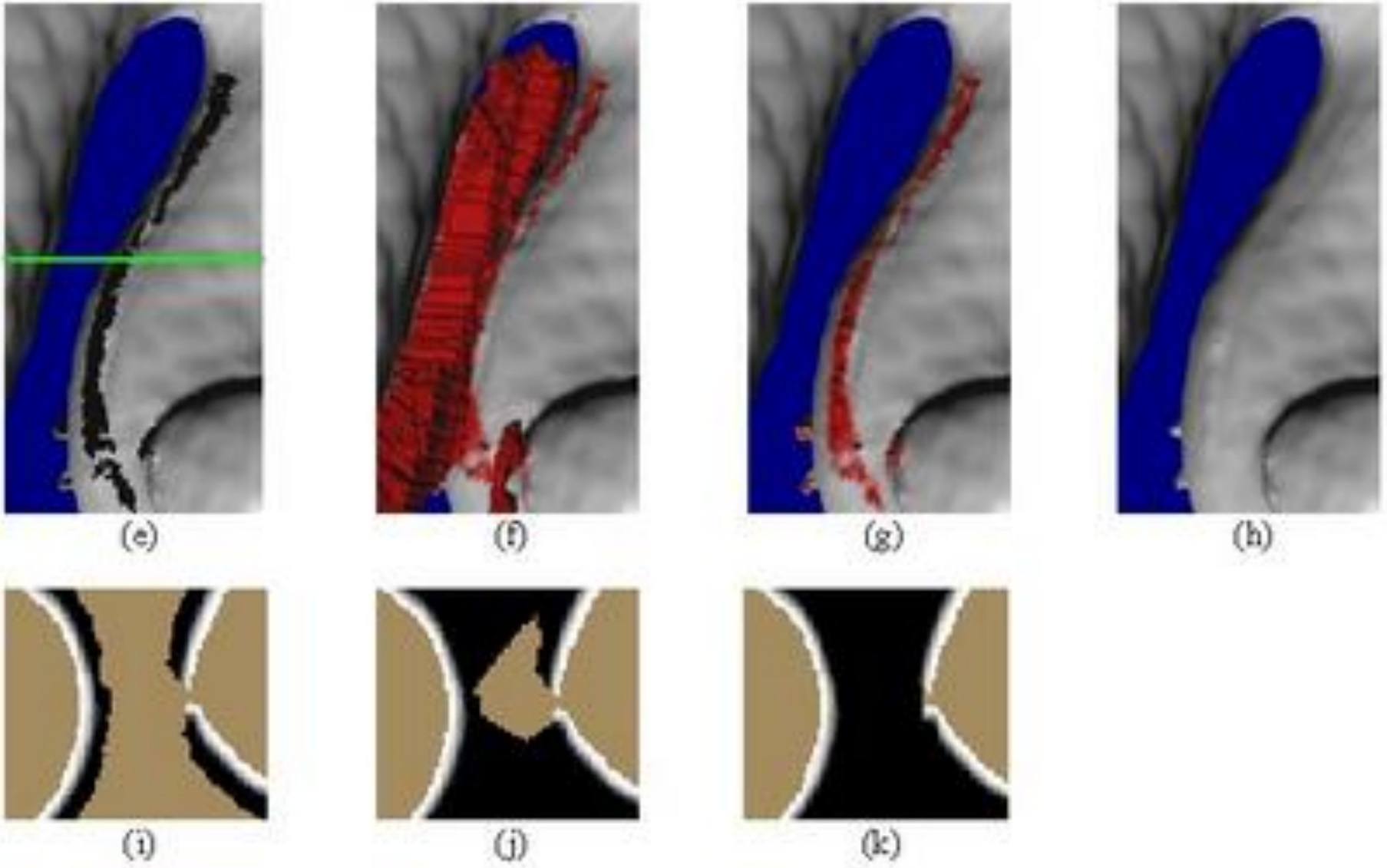
(g)

Original drill bit

Results: Hole filling



Results: Hole filling



Results: Detail achieved

(a) (b) (c) (d) (e)

Limitations

- Scanning
  - Accessing all surface points is challenging
  - Reflectance affects results
- Volumetric method
  - Thin surfaces
  - Sharp corners

Possible Extensions

- Improving execution time of space carving algorithm
- Large-scale scenes