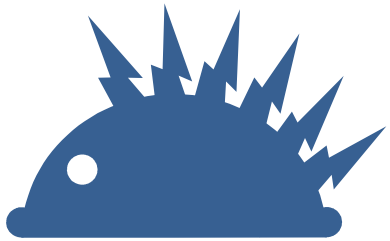
BME IIT CG

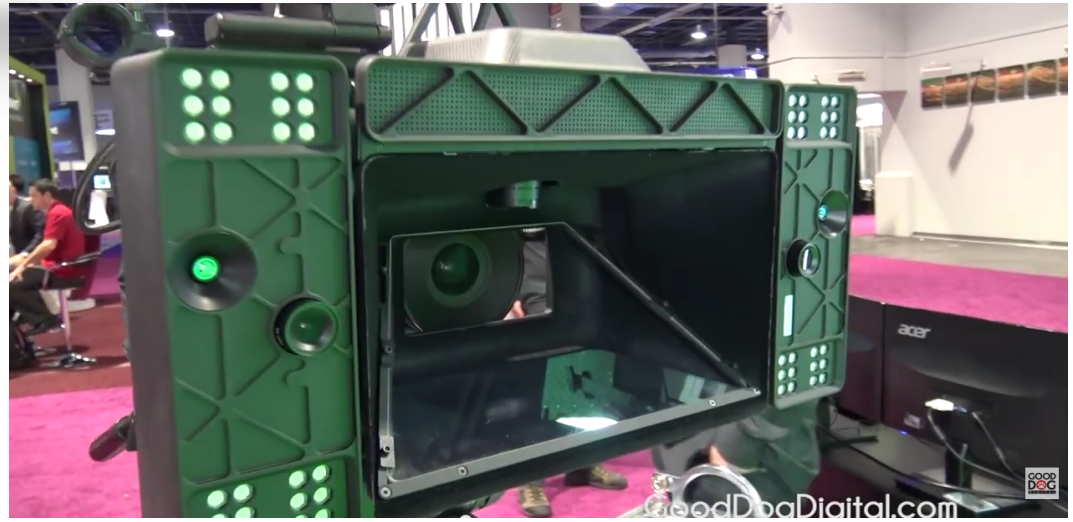


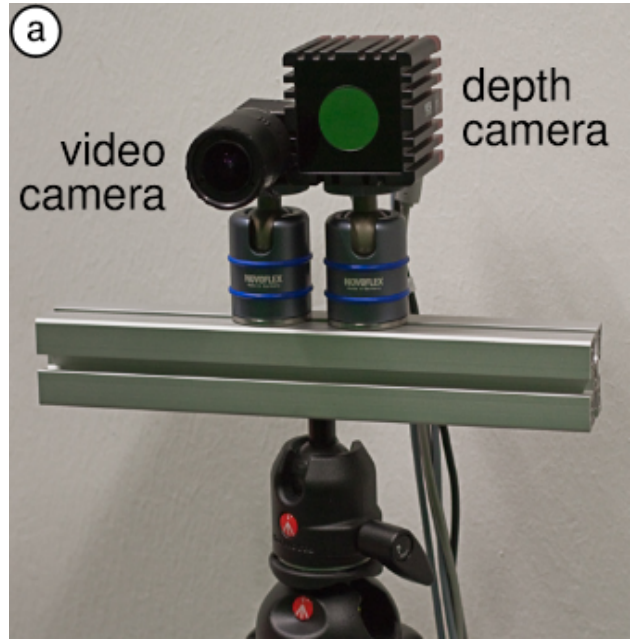
Hierarchical Volumetric Fusion of Depth Images

László Szirmay-Kalos, Milán Magdics
Balázs Tóth, Tamás Umenhoffer

Real-time color & 3D information

- Affordable integrated depth and color cameras

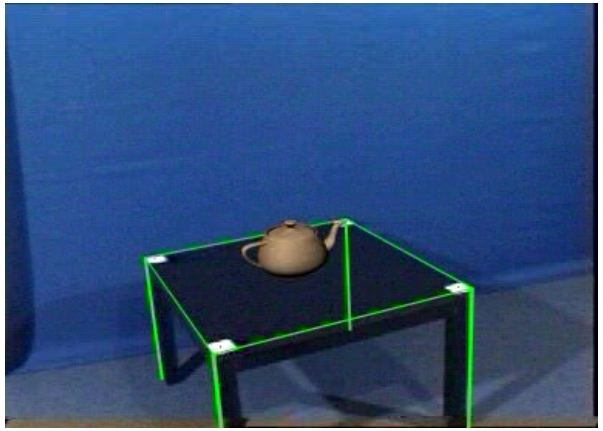




Application: 3D scanning

Application: limitations of compositing

Chroma keying



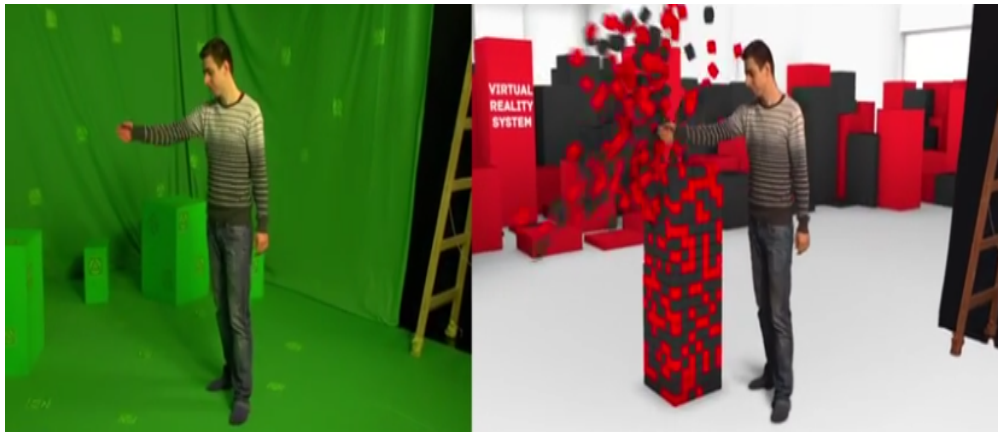
Augmented reality

Compositing can be based on color:

- Fixed order
- No shadows
- No reflections, refractions, cross illumination

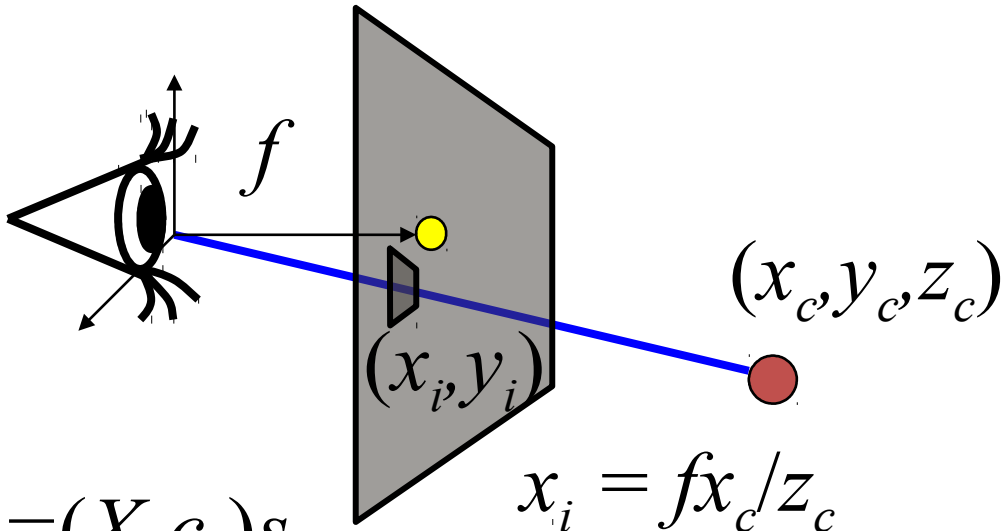
Deph compositing (Zinemath)

- Zinemath - ZLense



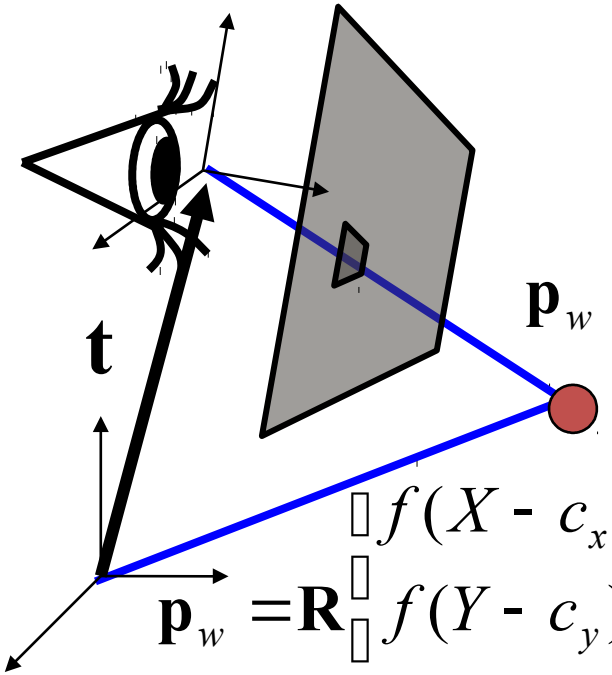
3D reconstruction of a point

$$\mathbf{R} = \begin{bmatrix} \mathbf{i}_x^* & \mathbf{j}_x^* & \mathbf{k}_x^* \\ \mathbf{i}_y^* & \mathbf{j}_y^* & \mathbf{k}_y^* \\ \mathbf{i}_z^* & \mathbf{j}_z^* & \mathbf{k}_z^* \end{bmatrix}$$



$$x_i = f x_c / z_c$$

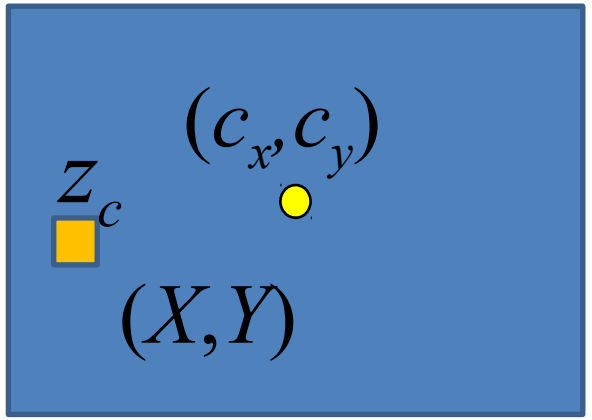
$$y_i = f y_c / z_c$$



$$\mathbf{p}_w = \mathbf{R}\mathbf{p}_c + \mathbf{t}$$

$$x_i = (X - c_x) s_x$$

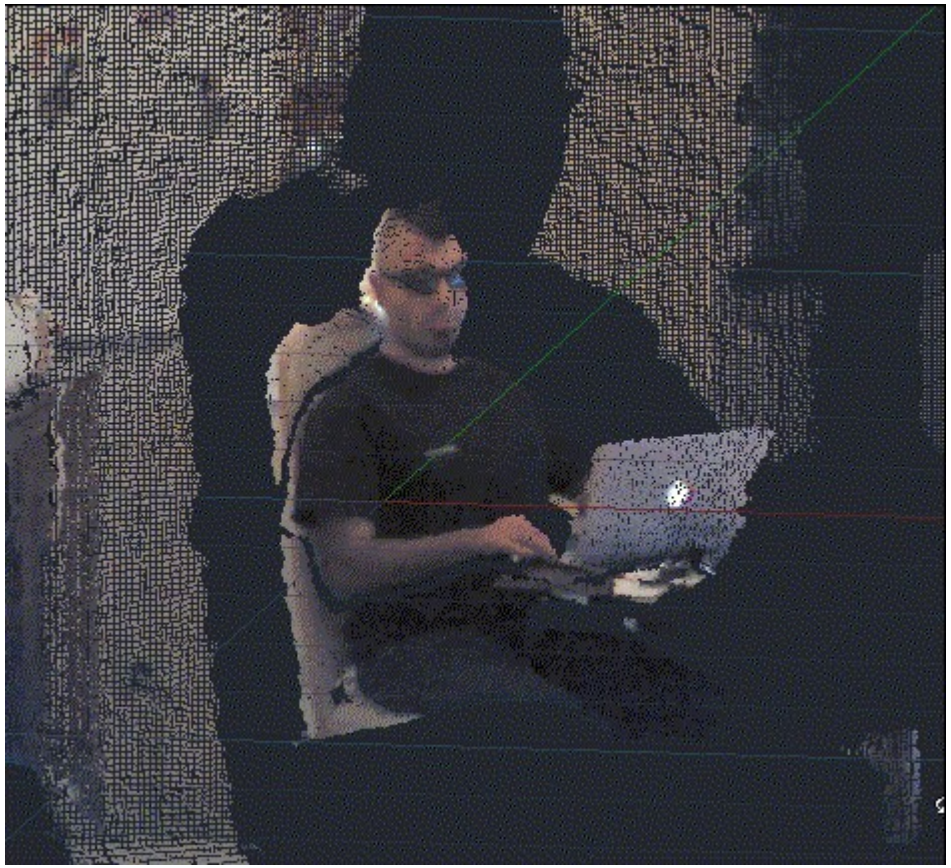
$$y_i = (Y - c_y) s_y$$



$$\mathbf{p}_w = \mathbf{R} \begin{bmatrix} f(X - c_x)s_x \\ f(Y - c_y)s_y \\ 1 \end{bmatrix} z_c + \mathbf{t} = \mathbf{d}(X,Y) z_c + \mathbf{t}$$

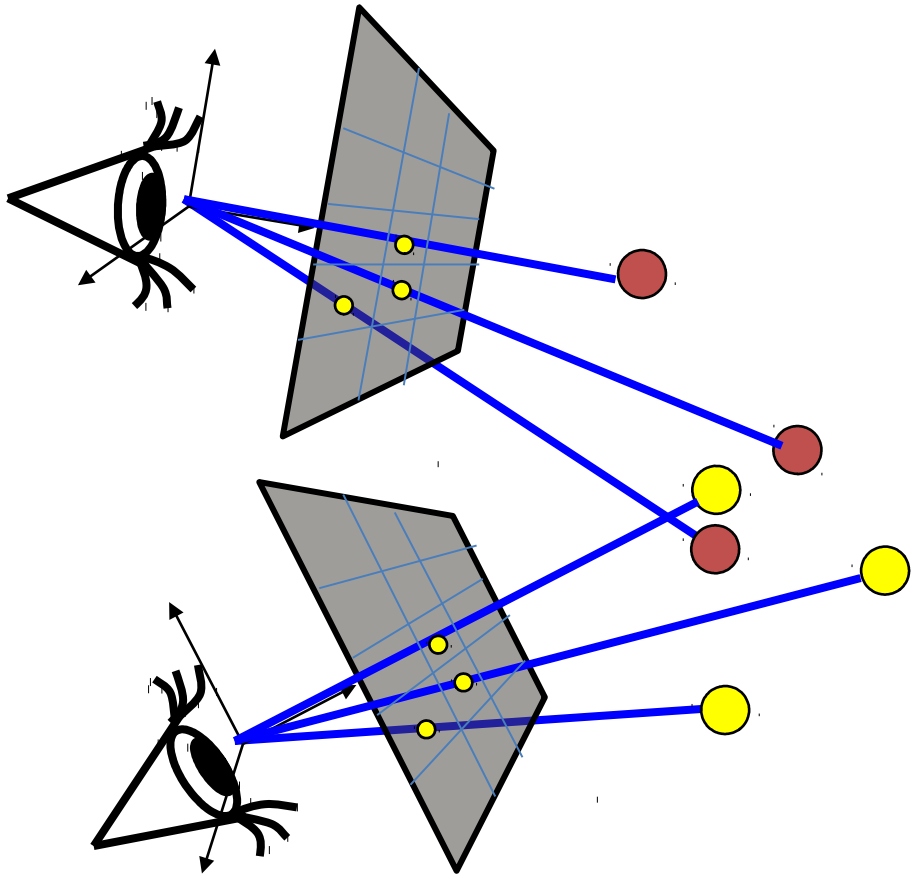
Back projection

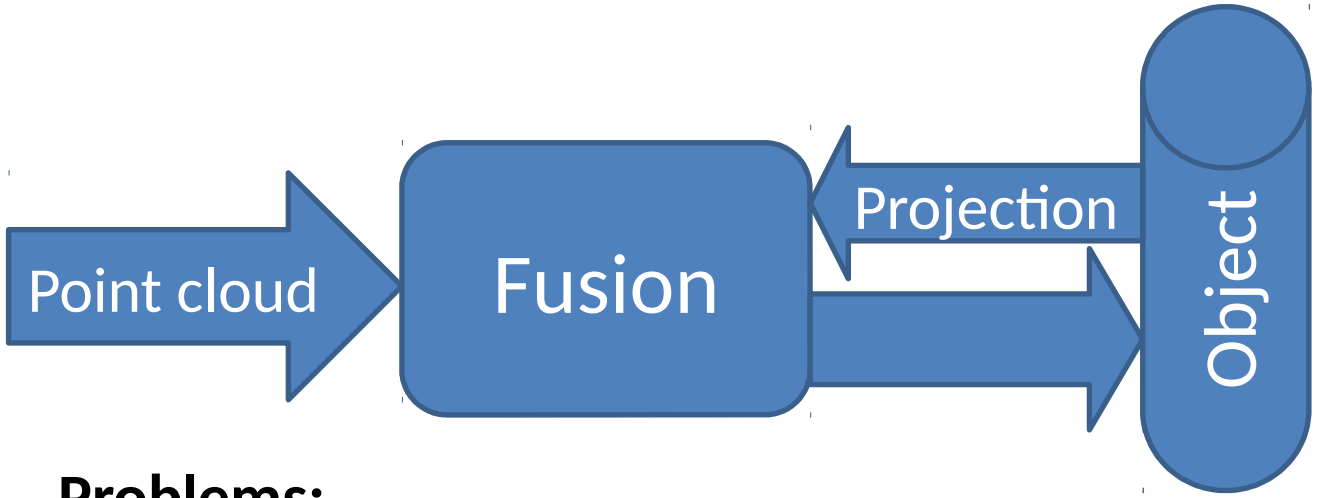
3D point cloud



http://through-the-interface.typepad.com/through_the_interface/2012/11/updated-autocad-integration-samples-for-kinect-sdk-v16.html

Dynamic camera, static scene





Problems:

- in different images the camera changes
  camera tracking based on static objects
- in different frames different points are visible
  We need to maintain surface information between points

Solution (Curless/Levoy):

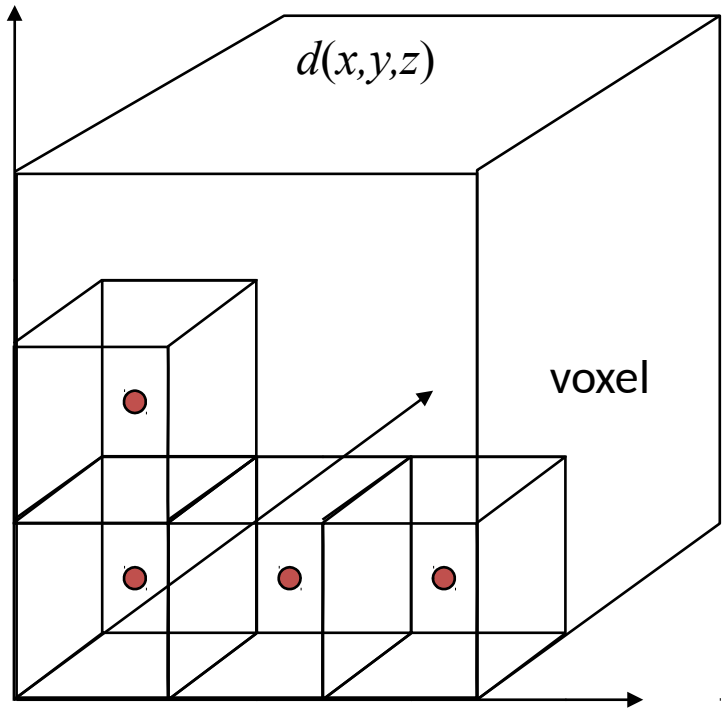
- Scene is represented by an emerging distance field

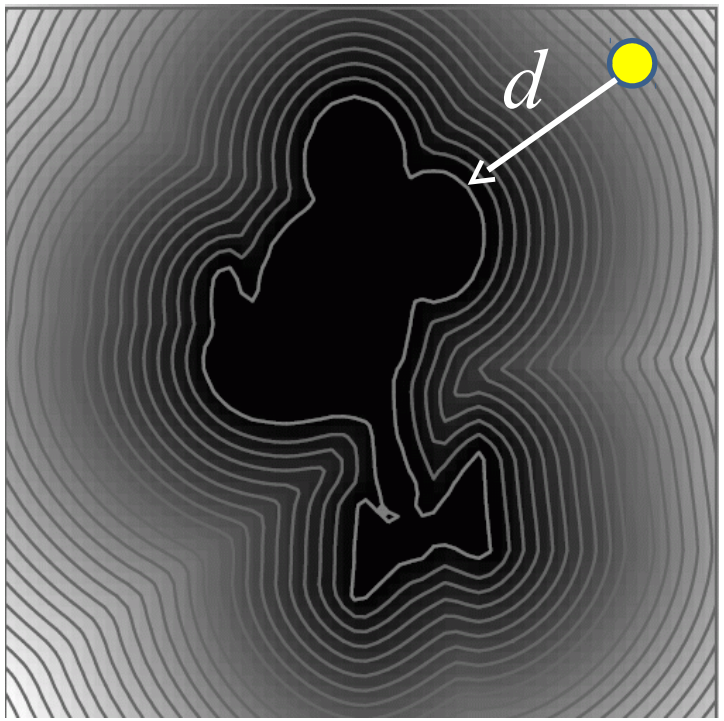
3D reconstruction input

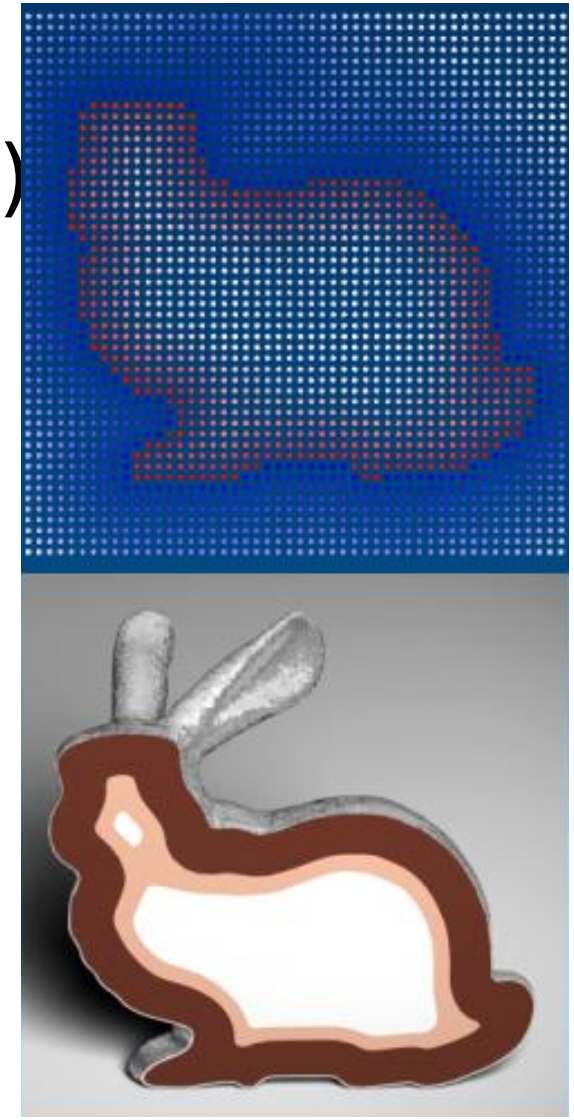
- Depth image: distance of the visible surface in each pixel
  - Noisy and unreliable

Surface reconstruction

- Curless-Levoy algorithm
- Truncated Signed Distance Field (TSDF)





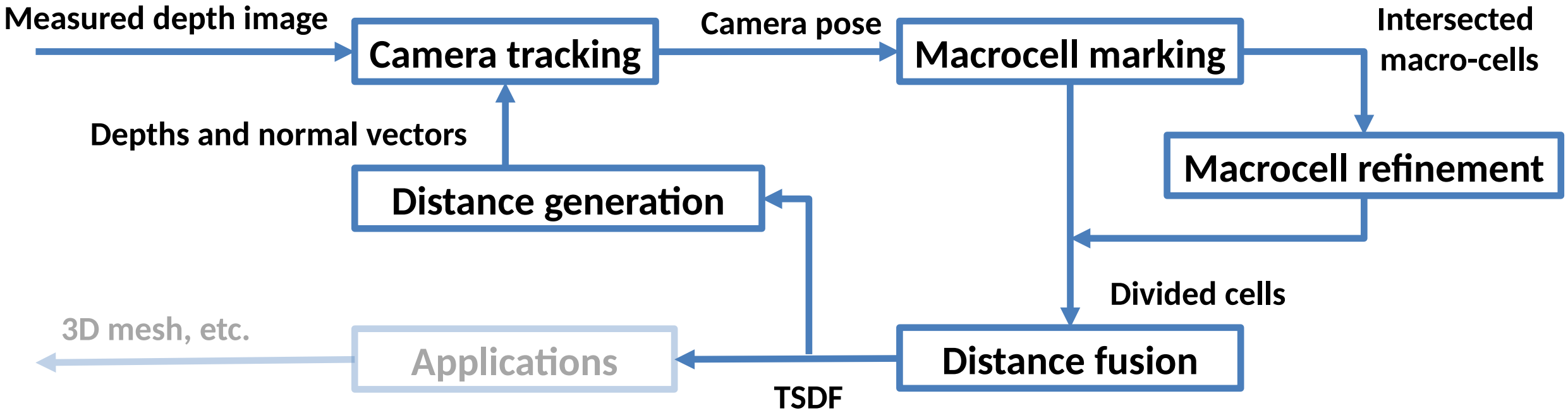


Aims

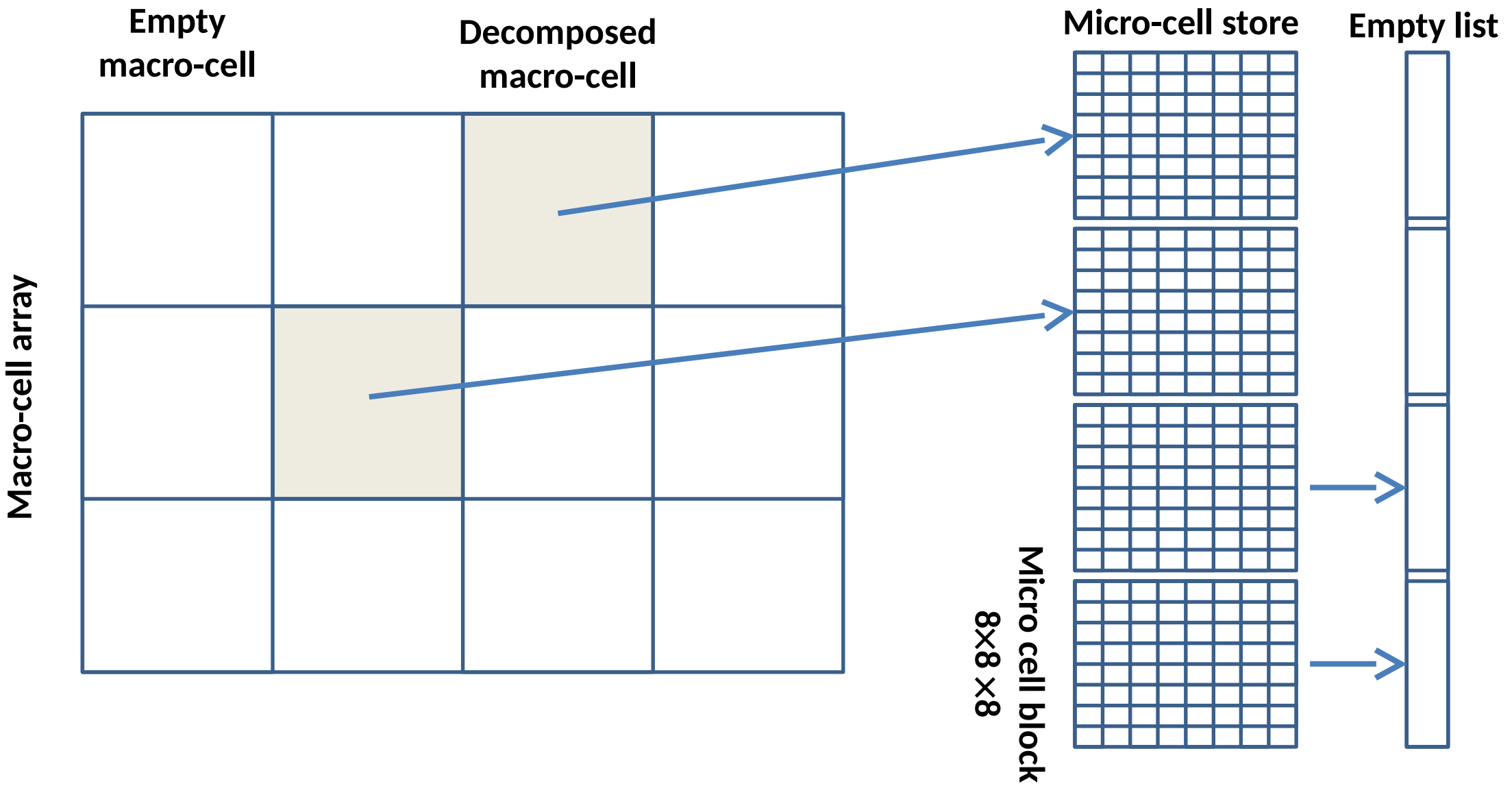
- Reconstruct static scenes with moving camera
- **Real-time** reconstruction
  - GPU-based implementation
- Fast camera tracking
  - Common methods (SIFT, SURF etc.) are slow
- Efficient, **high resolution** TSDF storage
  - To reconstruct fine geometric details
  - GPU memory is limited

Proposed method

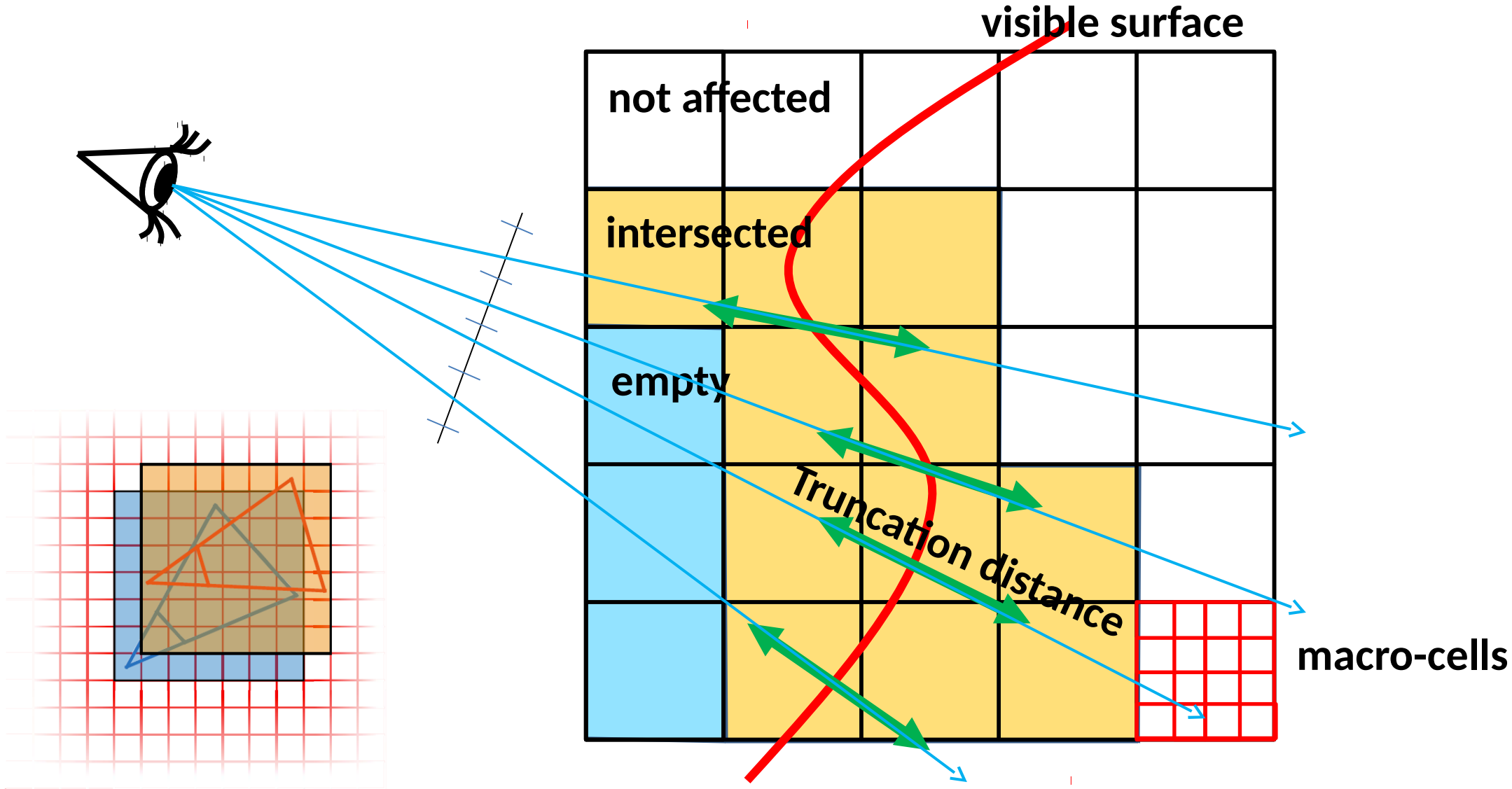
- Two-level, **hierarchical** TSDF
  - Observation: usually most of the scanned 3D space is empty
- Iterative reconstruction algorithm



Hierarchical TSDF

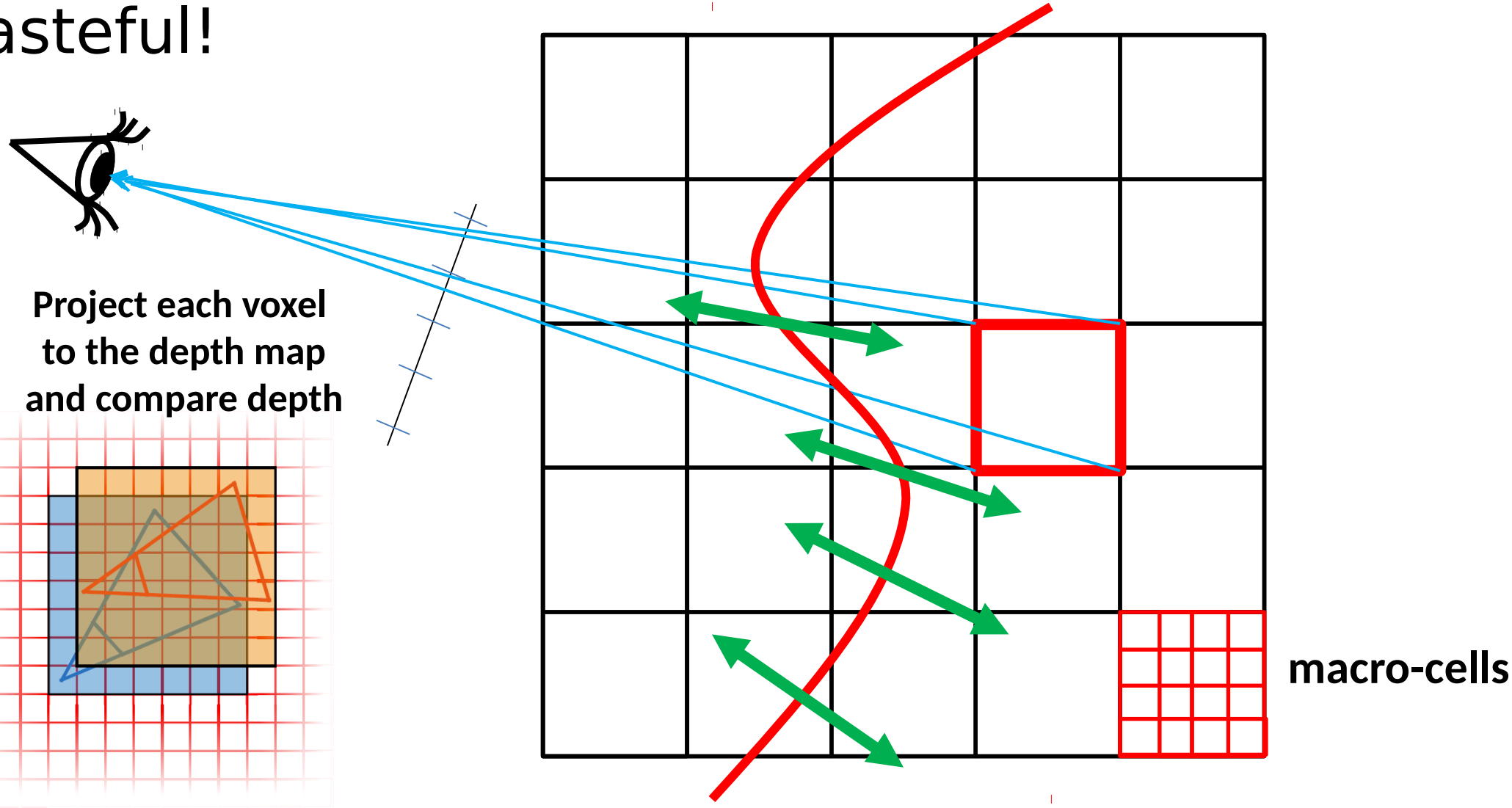


Macro-cell marking

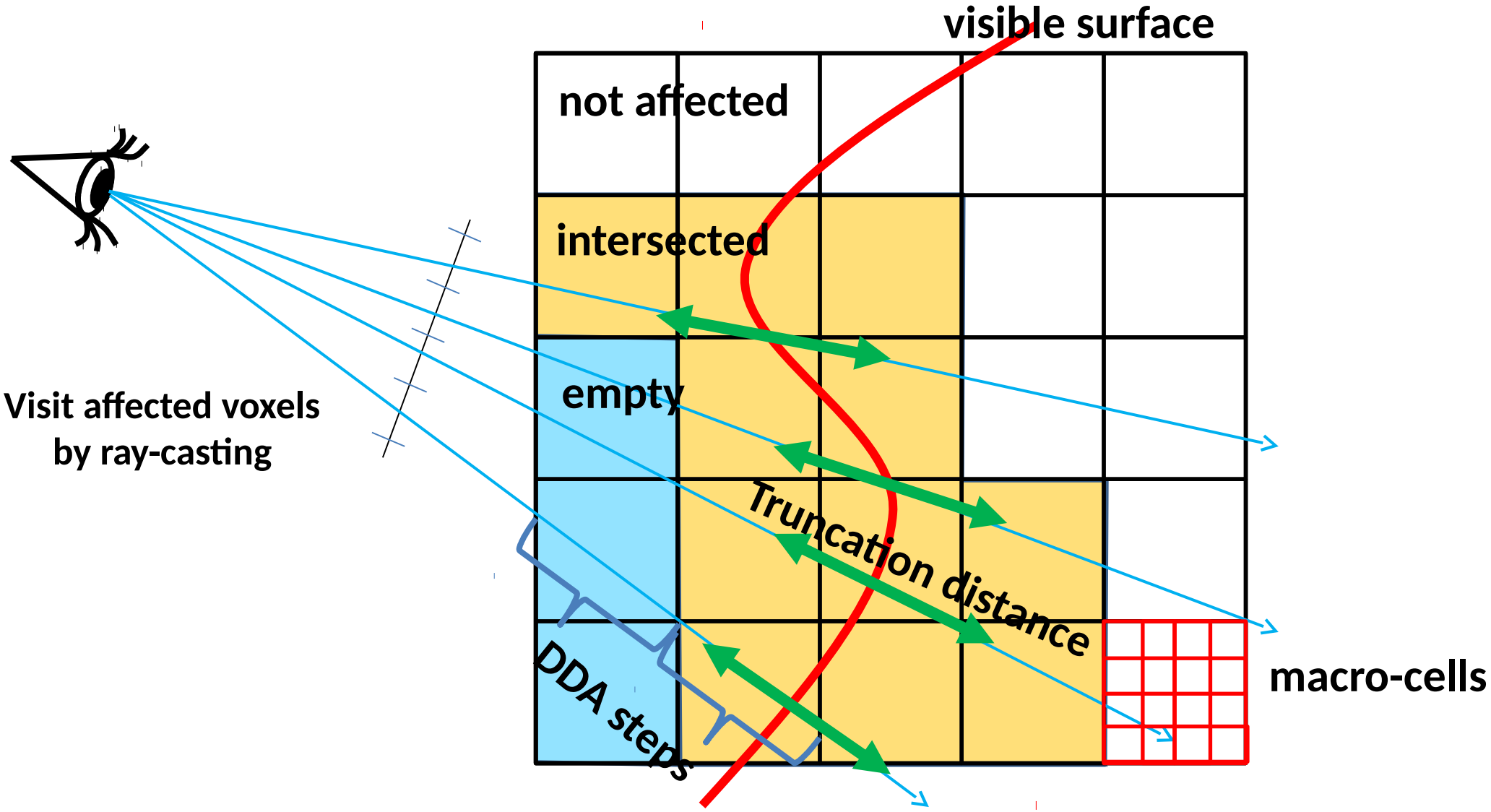


Macro-cell marking: gather-style

- Wasteful!

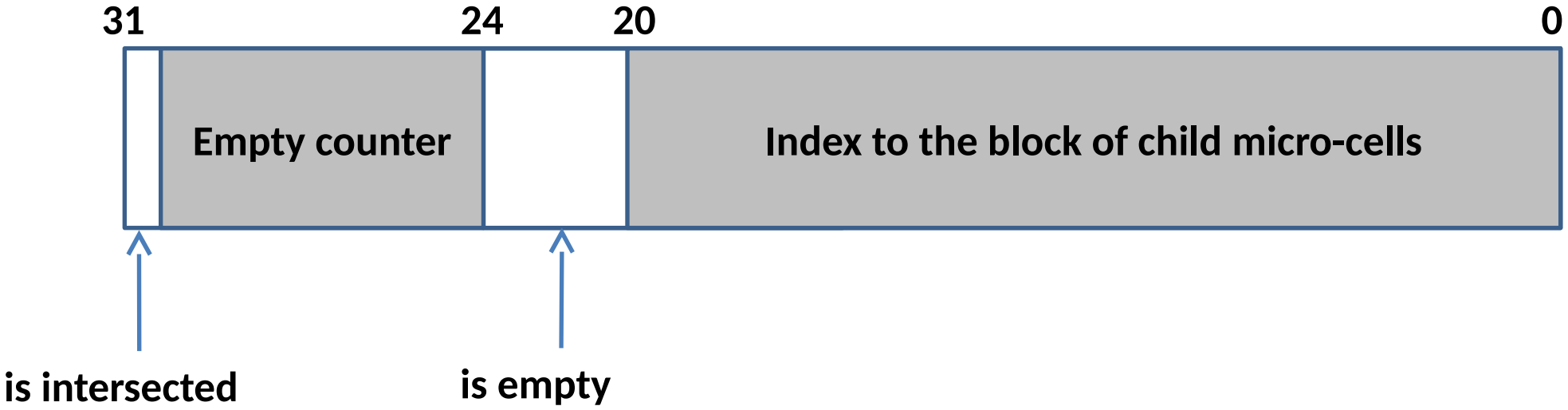


Scatter-style marking (but still faster)



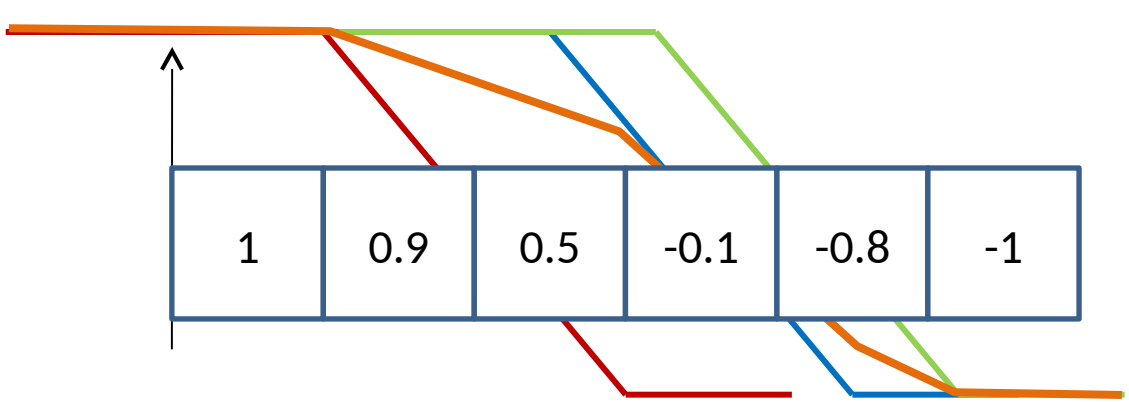
Avoiding atomic operations

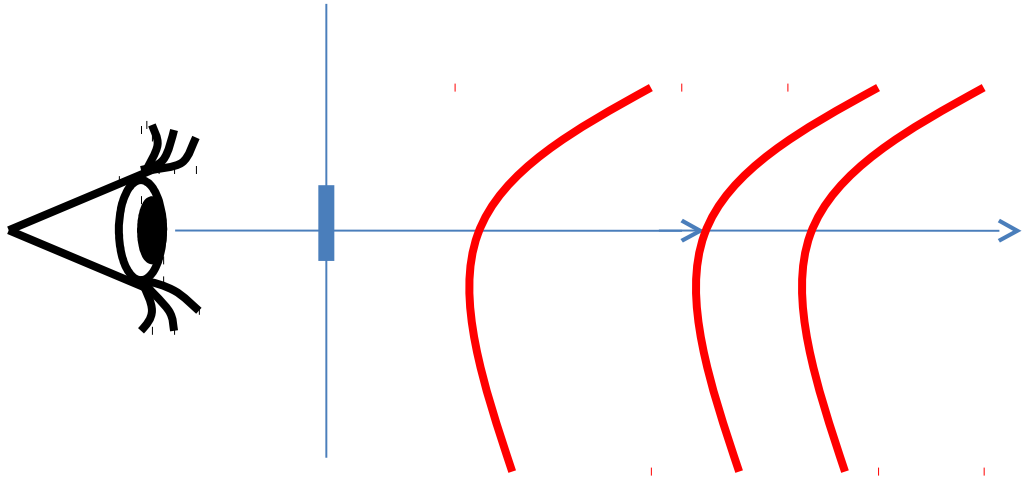
- Macro-cell marking
  - Determine empty and intersected cells
  - Without synchronization!

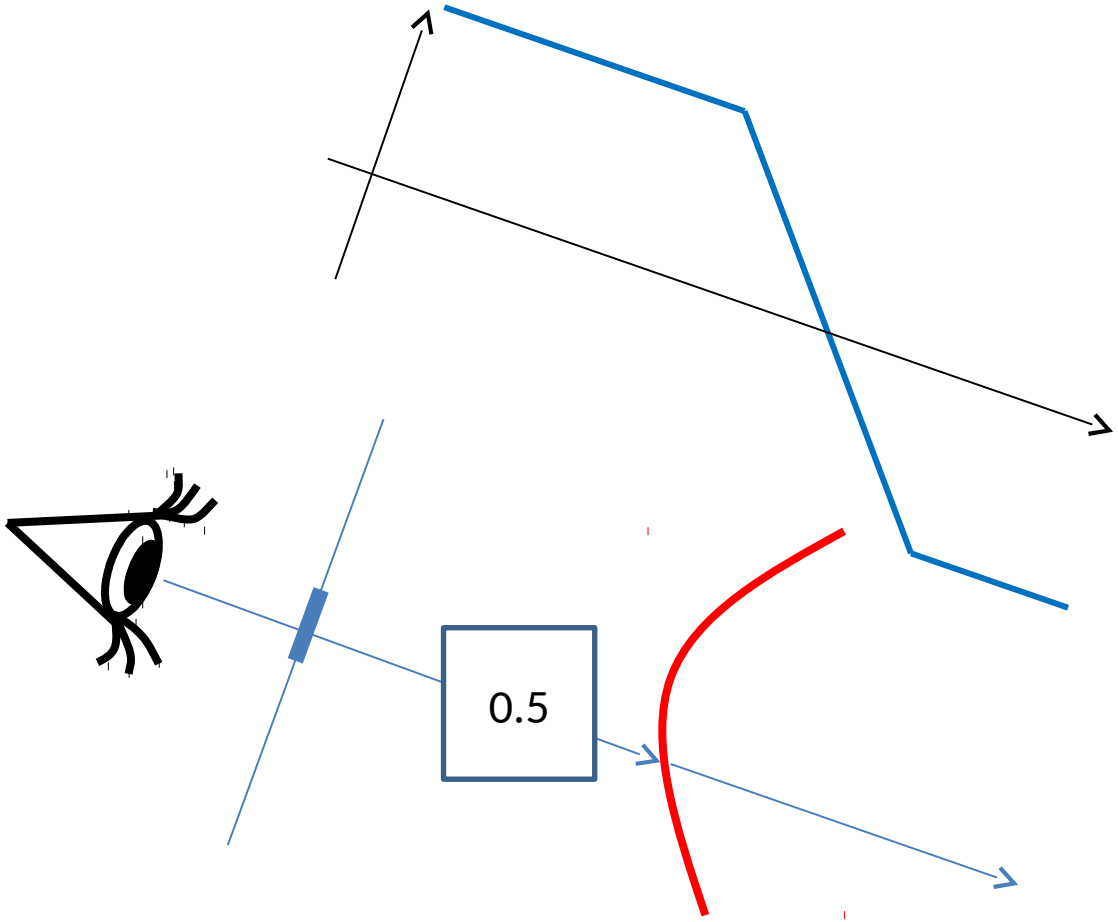


Fusion

- Distance fusion
  - Only for the previously marked micro-cells

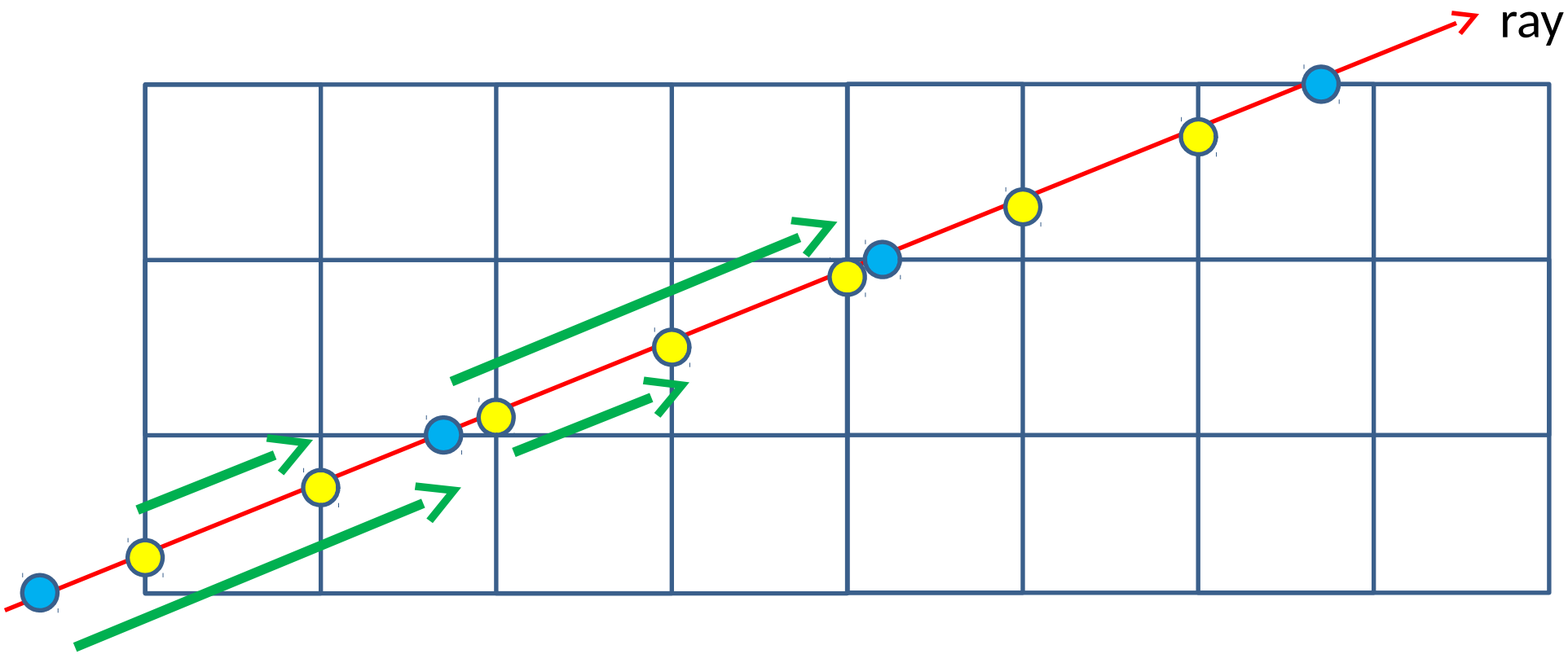






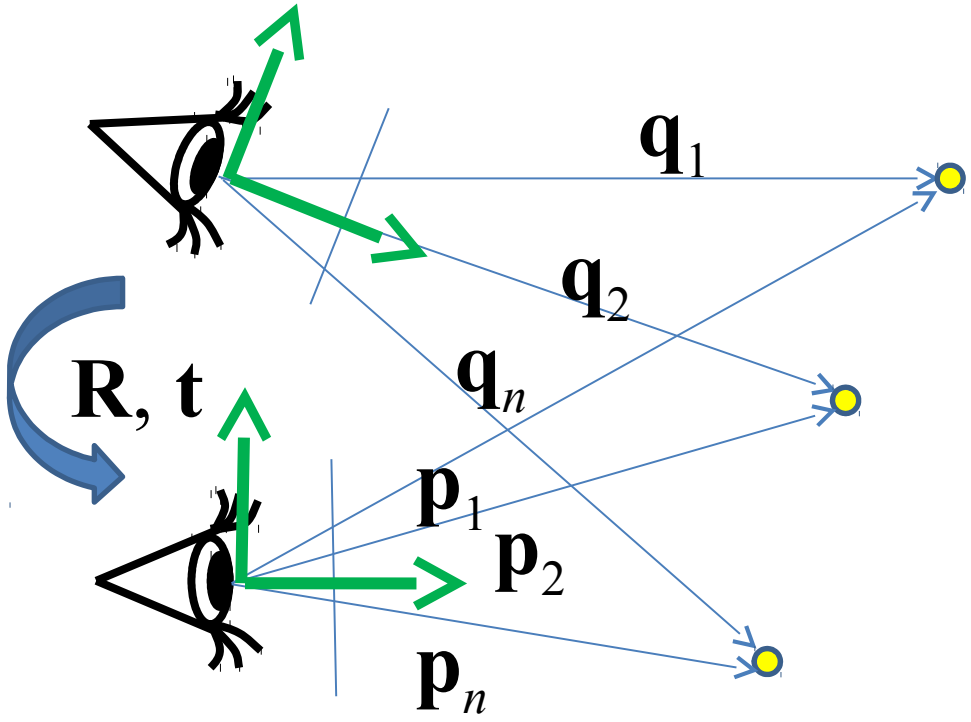
Rendering

- Distance map generation
  - Hierarchical DDA
    - Different step size in the macro and micro cells

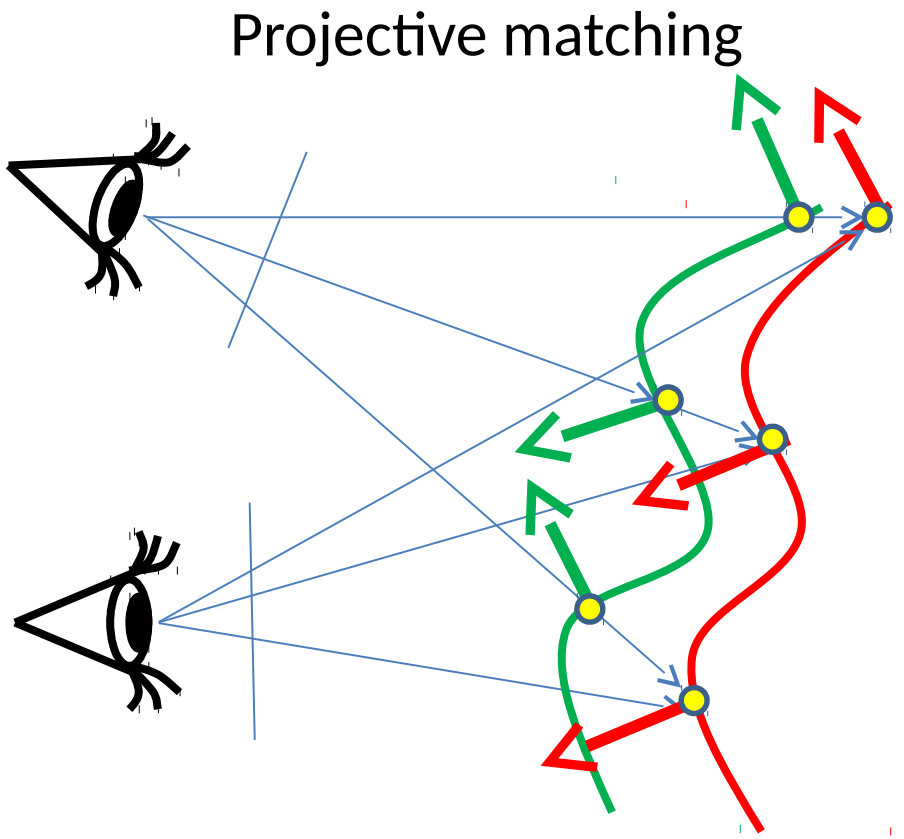


Camera tracking

- Iterative Closest Point (ICP)

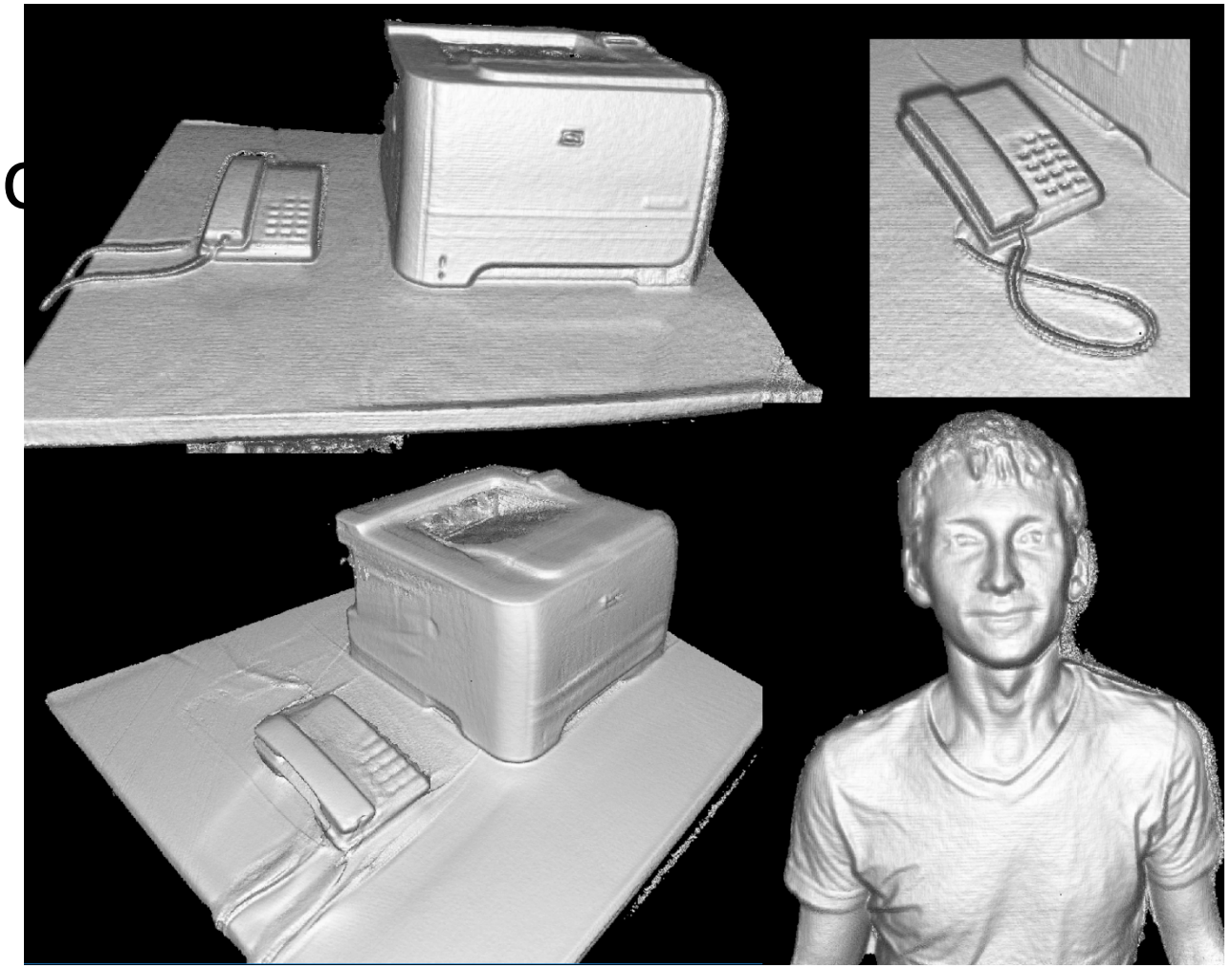


$$E(\mathbf{R},\mathbf{t}) = \frac{1}{n}\sum_i \left\| \mathbf{p}_i - \mathbf{R}\cdot\mathbf{q}_i - \mathbf{t} \right\| \rightarrow \min$$



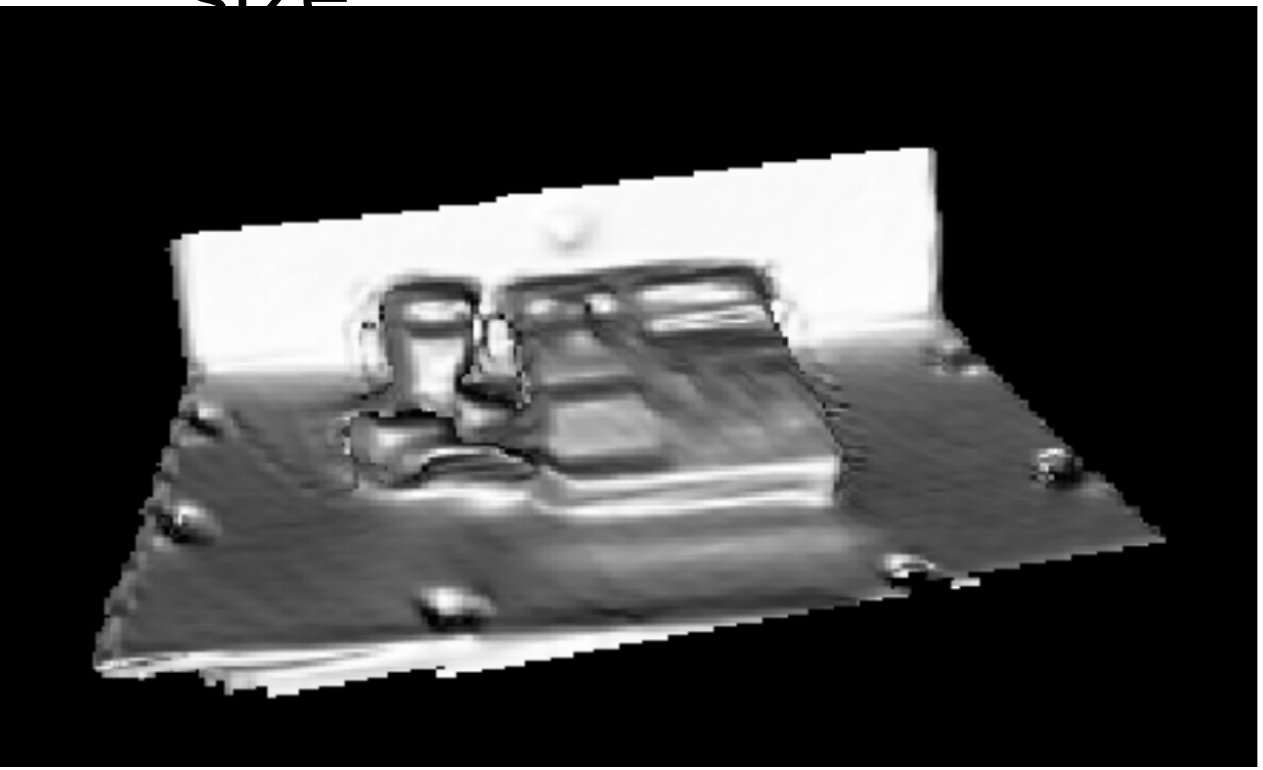
Results

- Kinect2 depth camera
- NVIDIA 690GTX GPU
- Real-time reconstructio
- 1mm cell resolution

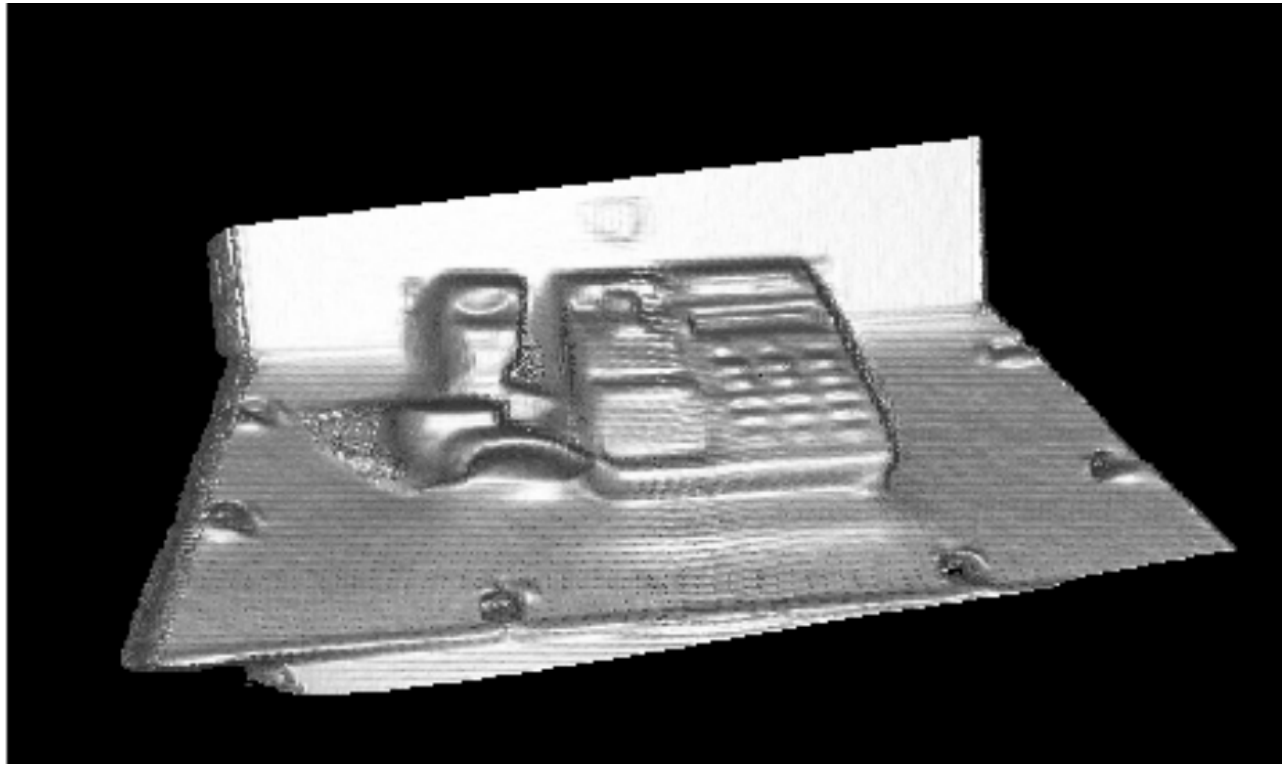


Results

- With the same memory usage: 8mm vs 1mm cell size

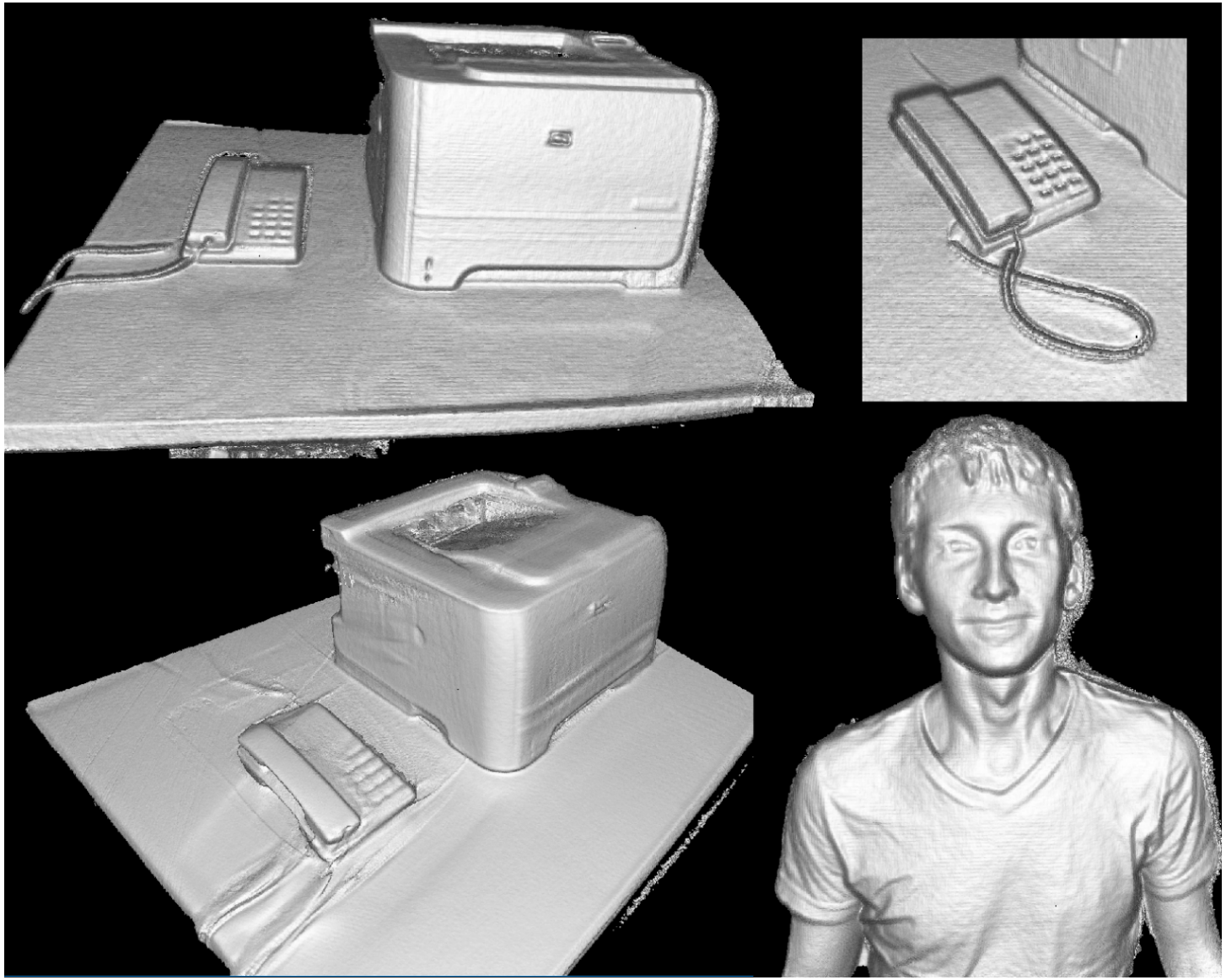


Kinect Fusion

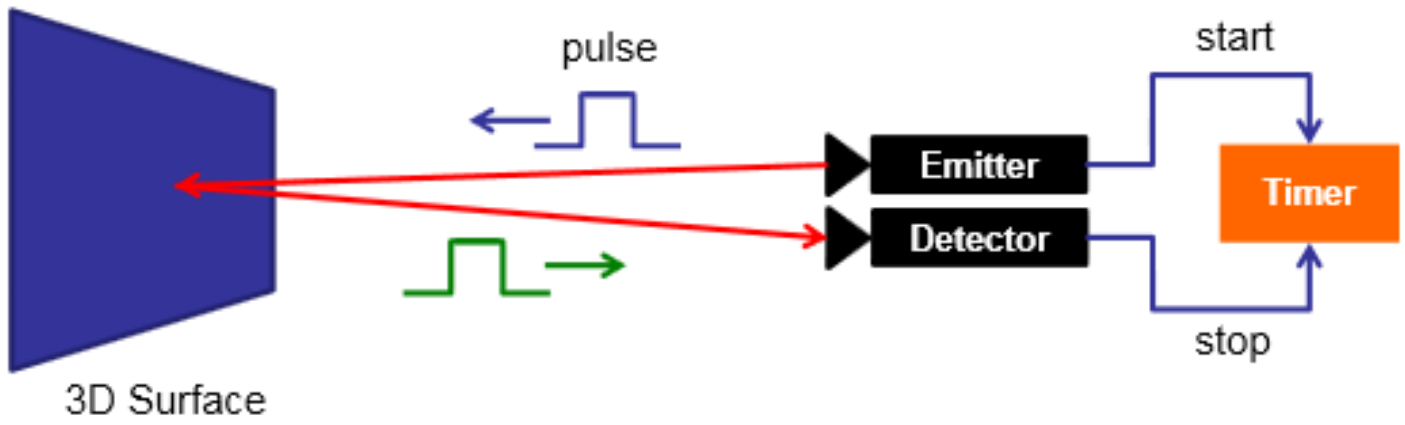


Proposed method

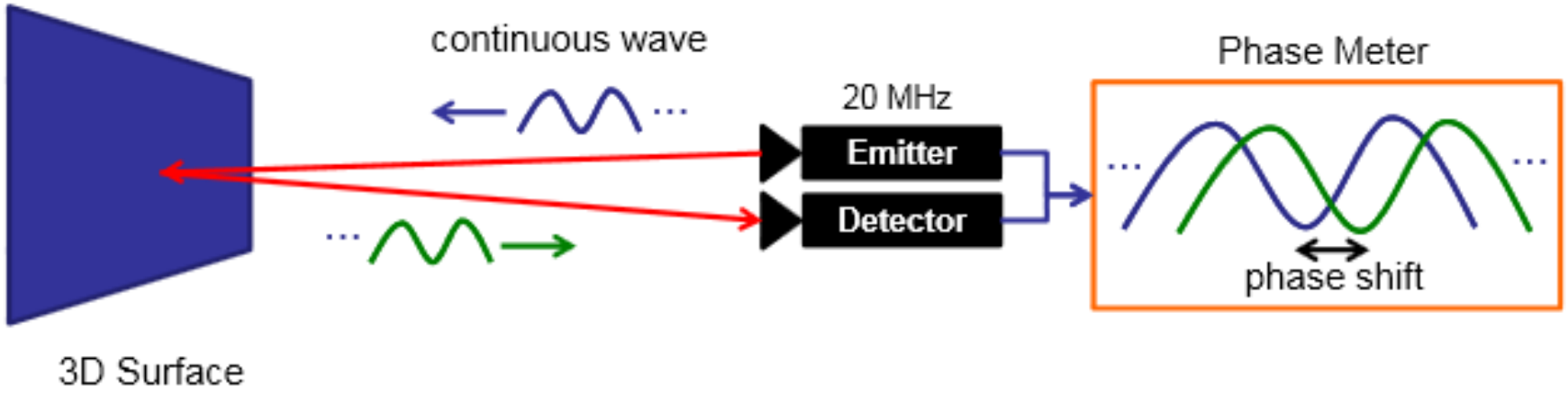
Thank you!



Time of flight depth sensors



Pulsed modulation:
Accurate time measurement expensive



Continuous modulation
Periodic distance