Lecture 5.
Dense Reconstruction and Tracking with Real-Time Applications

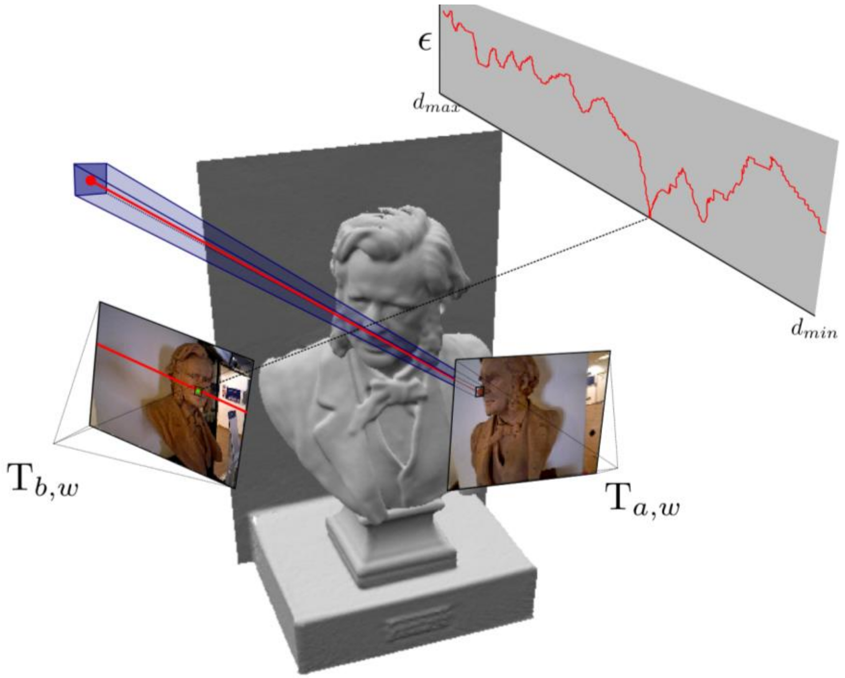
Part 2: Geometric Reconstruction

Dr Richard Newcombe and Dr Steven Lovegrove

Slide content developed from:

[Newcombe, “Dense Visual SLAM”, 2015] [Lovegrove, “Parametric Dense Parametric SLAM”]

and [Szeliski, Seitz, Zitnick UW CSE576 CV lectures]



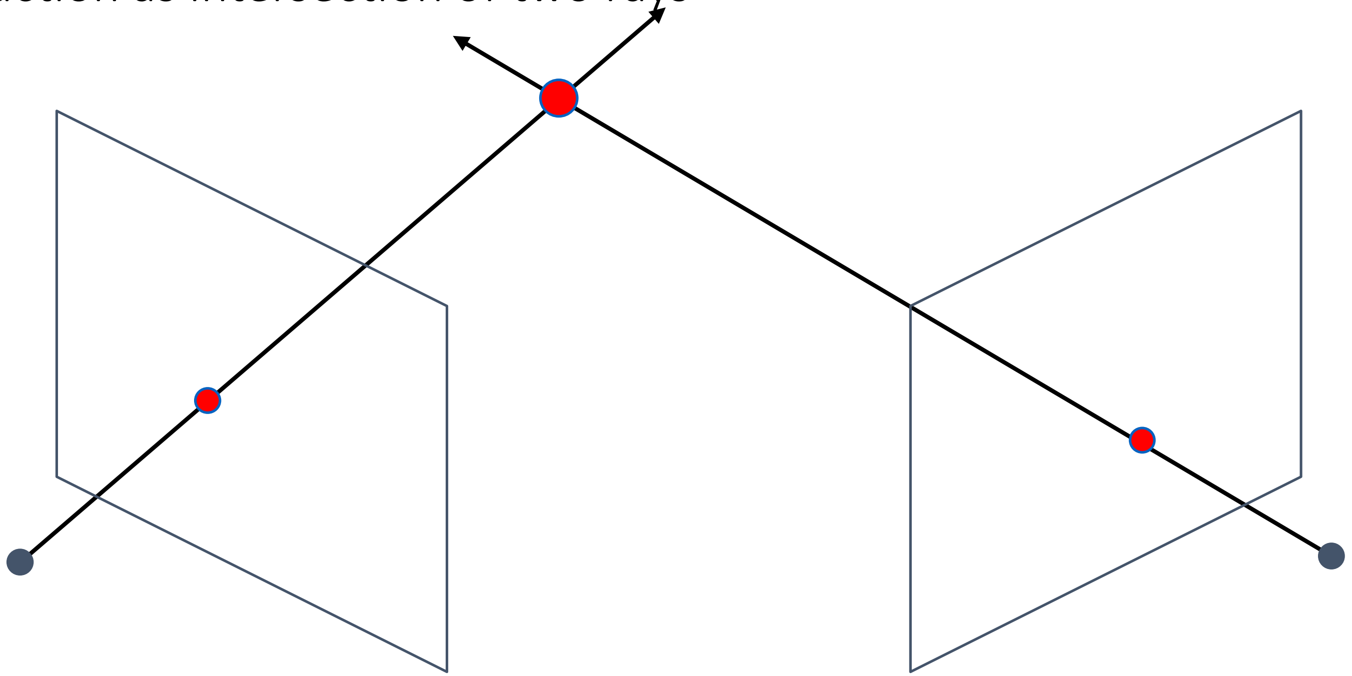
Geometric Reconstruction

Dense reconstruction of scene geometry

Stereo and Constrained Correspondence

Basic Principle: Triangulation

- Gives reconstruction as intersection of two rays

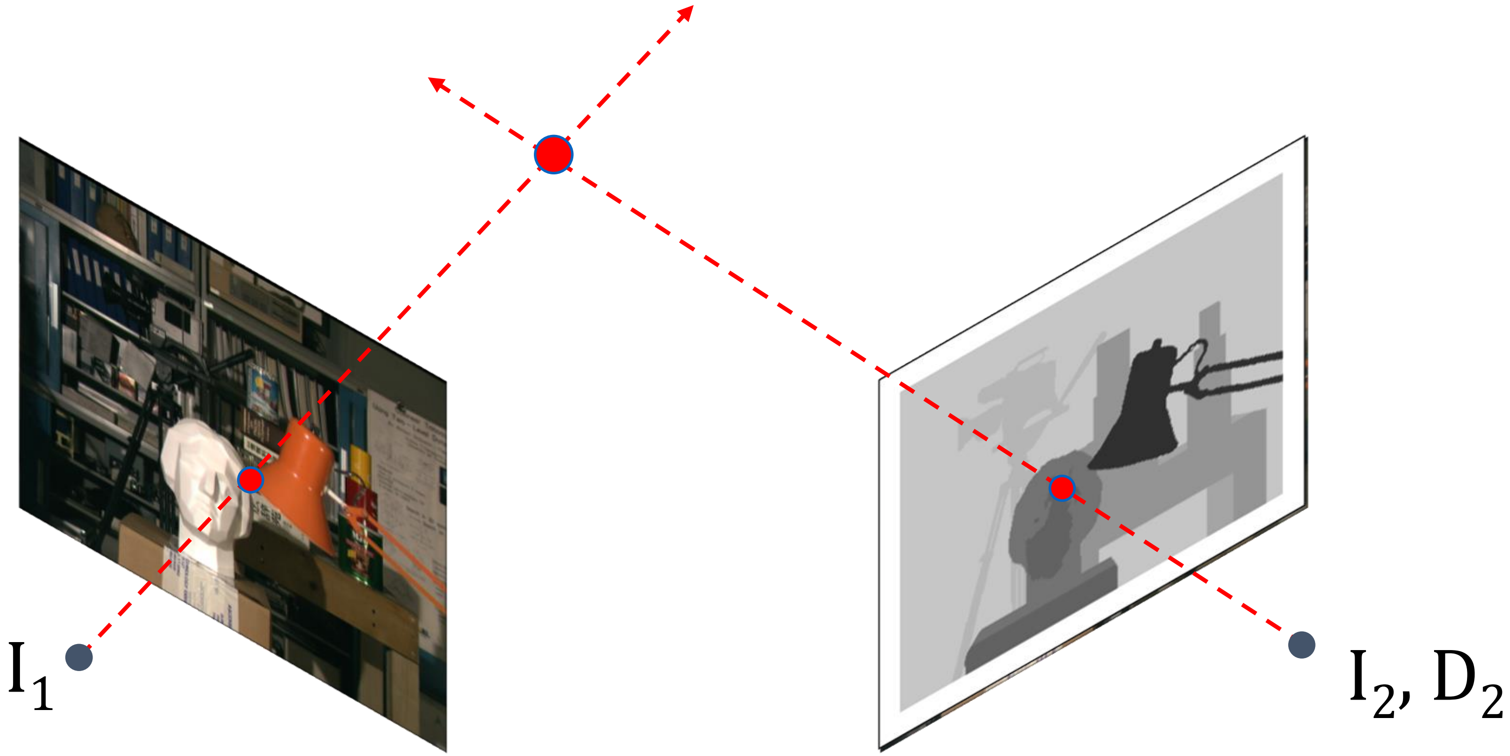


Requires

– camera pose (calibration)

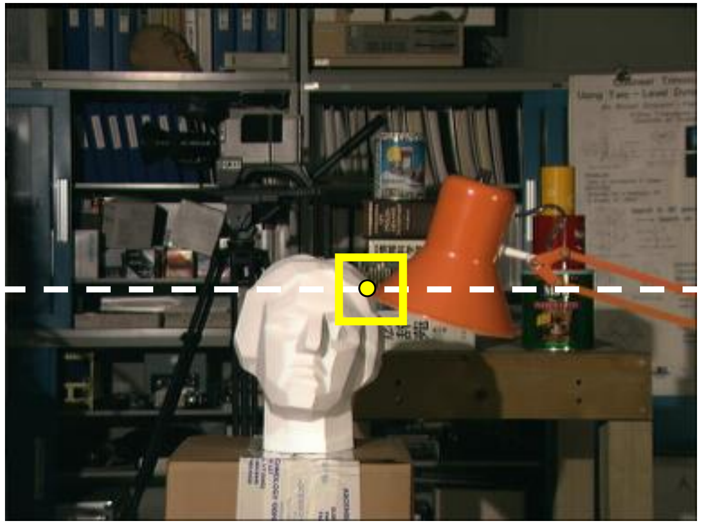
– *point correspondence (e.g. feature extraction and matching)*

Dense Scene Geometry Generative Model

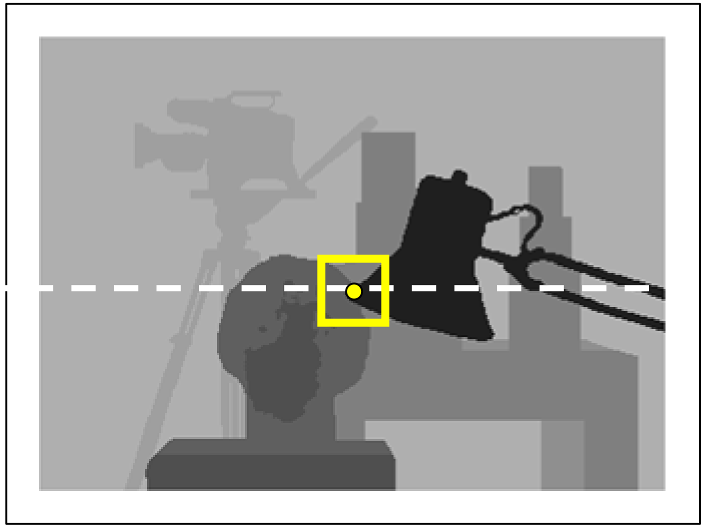


$$I_2\,(x,y) = I_1(\,\pi(\,T_{12}\,K^{-1}\,D_2\,(x,y)\,[x,y]\,)\,)$$

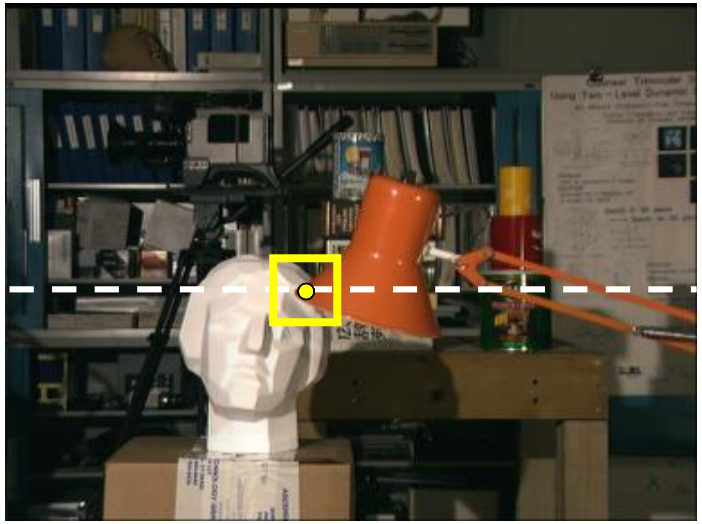
Special Case for a Rectified Stereo Image Pair



I_1

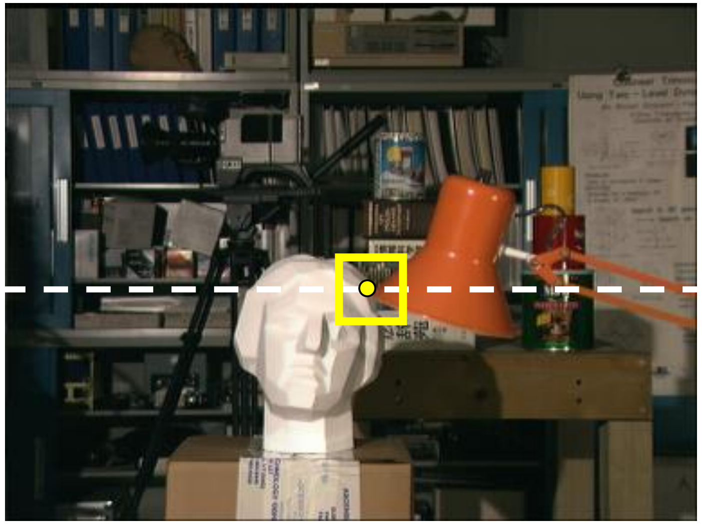


Depth Image (D_2)

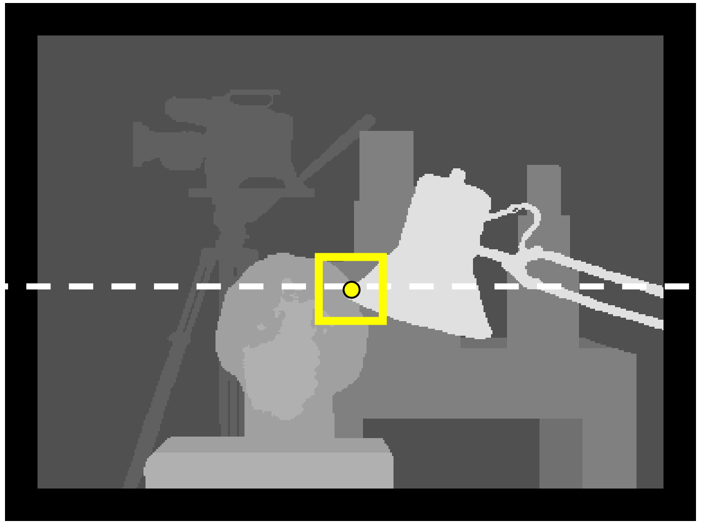


I_2

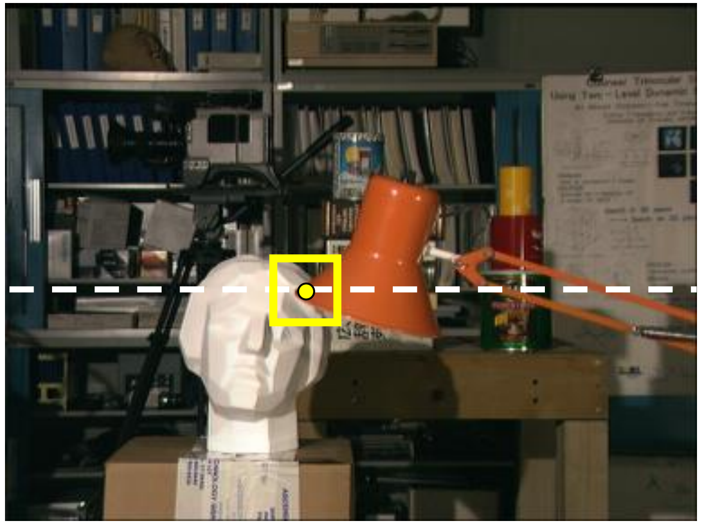
Disparity for Rectified Stereo Pairs



I_1



Disparity Image (d_2)



I_2

Rectified Stereo generative model with Brightness Constancy:

$$I_2(x,y) = I_1(x + d_2(x,y), y)$$

Stereo Correspondence as energy minimization

$I_1(x, y)$

$I_2(x, y)$

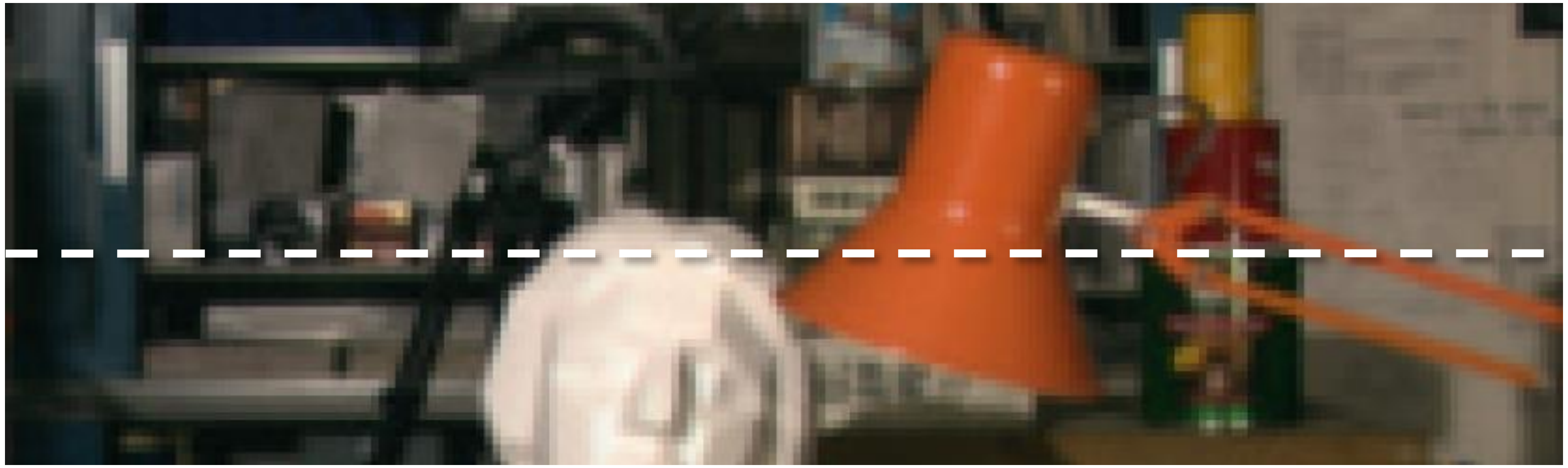
Pixel Error:

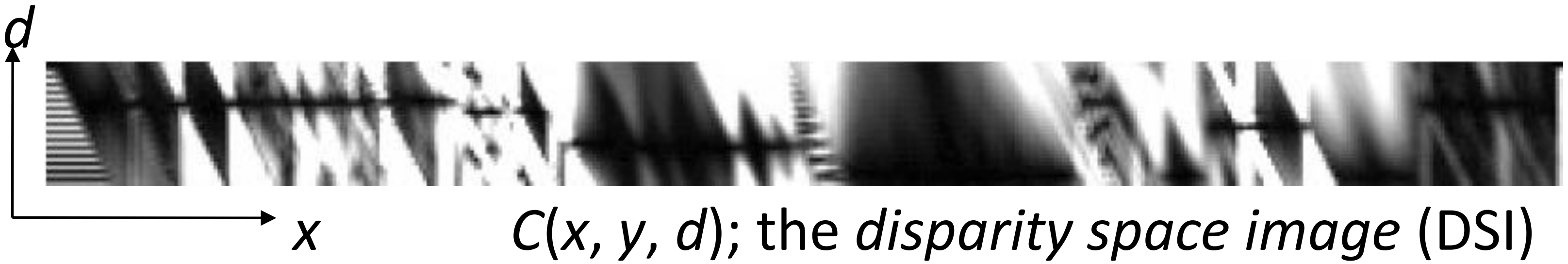
$e(x, y, d) = I_1(x + d, y) - I_2(x, y)$

Cost (with quadratic penalty):

$C(x, y, d) = (I_1(x + d, y) - I_2(x, y))^2$

y = 141





$C(x, y, d)$; the *disparity space image* (DSI)

Stereo as energy minimization

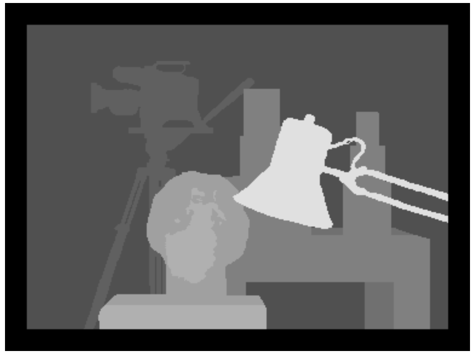
Simple pixel / window matching: choose the minimum of each column in the DSI independently:

$$d(x, y) = \arg\min_{d'} \; C(x, y, d')$$

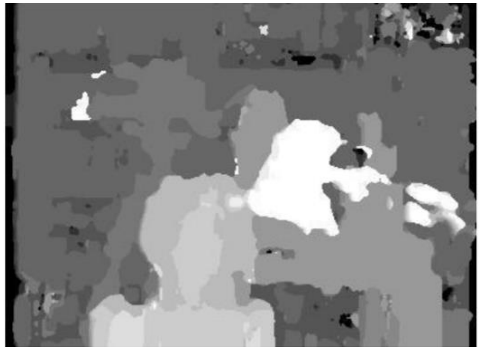
Aggregation window, error and cost functions

Effect of window size (W) for aggregating the photometric cost:

$$\sum_{(i,j)\in W} |I_1(i,j) - I_2(x+i, y+j)|$$


Ground truth


SAD W=3


SAD W=11

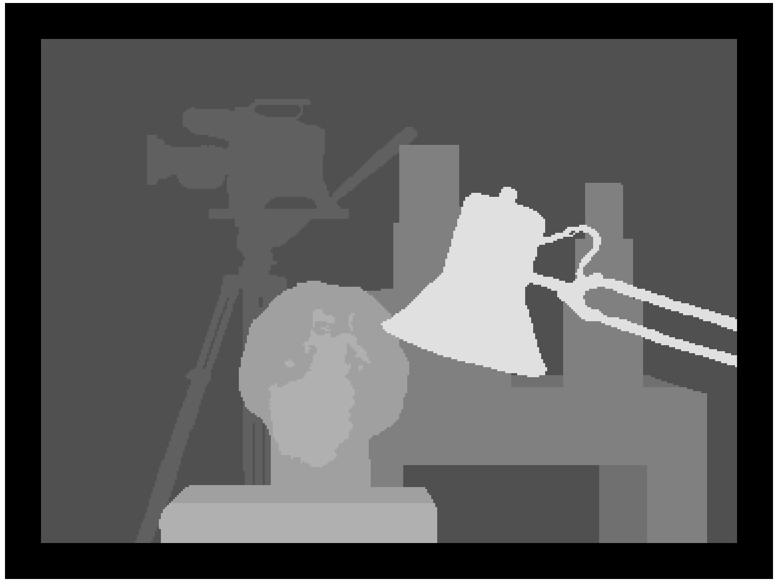

SAD W=25

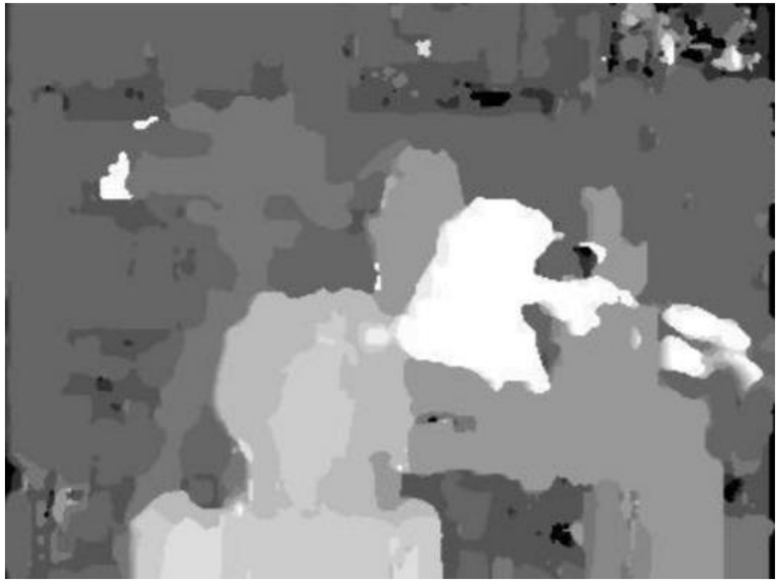
Aggregation window, error and cost functions

The design of the cost function, including *window size* for aggregation, *image error* function and *penalty* can improve quality of correspondence:

Similarity Measure	Formula
Sum of Absolute Differences (SAD)	$\sum_{(i,j)\in W} \lvert I_1(i,j) - I_2(x+i,y+j)\rvert$
Sum of Squared Differences (SSD)	$\sum_{(i,j)\in W} \left(I_1(i,j) - I_2(x+i,y+j)\right)^2$
Zero-mean SAD	$\sum_{(i,j)\in W} \lvert I_1(i,j) - \bar{I}_1(i,j) - I_2(x+i,y+j) + \bar{I}_2(x+i,y+j)\rvert$
Locally scaled SAD	$\sum_{(i,j)\in W} \lvert I_1(i,j) - \frac{\bar{I}_1(i,j)}{\bar{I}_2(x+i,y+j)} I_2(x+i,y+j)\rvert$
Normalized Cross Correlation (NCC)	$\frac{\sum_{(i,j)\in W} I_1(i,j).I_2(x+i,y+j)}{\sqrt[2]{\sum_{(i,j)\in W} I_1^2(i,j).\sum_{(i,j)\in W} I_2^2(x+i,y+j)}}$

More Advanced Aggregation Functions



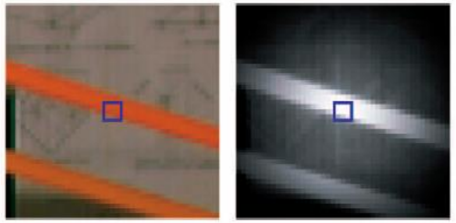


Ground truth

SAD, W=11

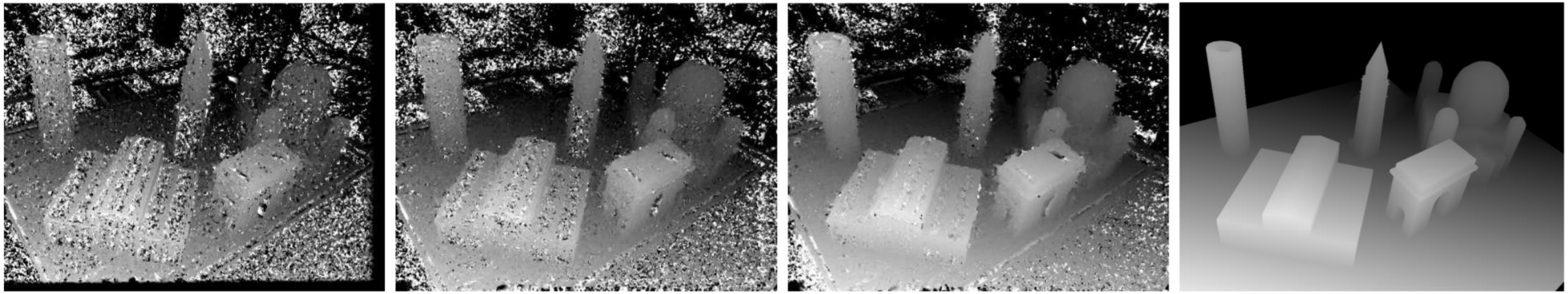
Adaptive Support-Weight Approach for Correspondence Search

[Yoon and Kweon, 2006]



Plane Sweep for Multiple view aggregation

- How to Integrate more information from Multiple Views?



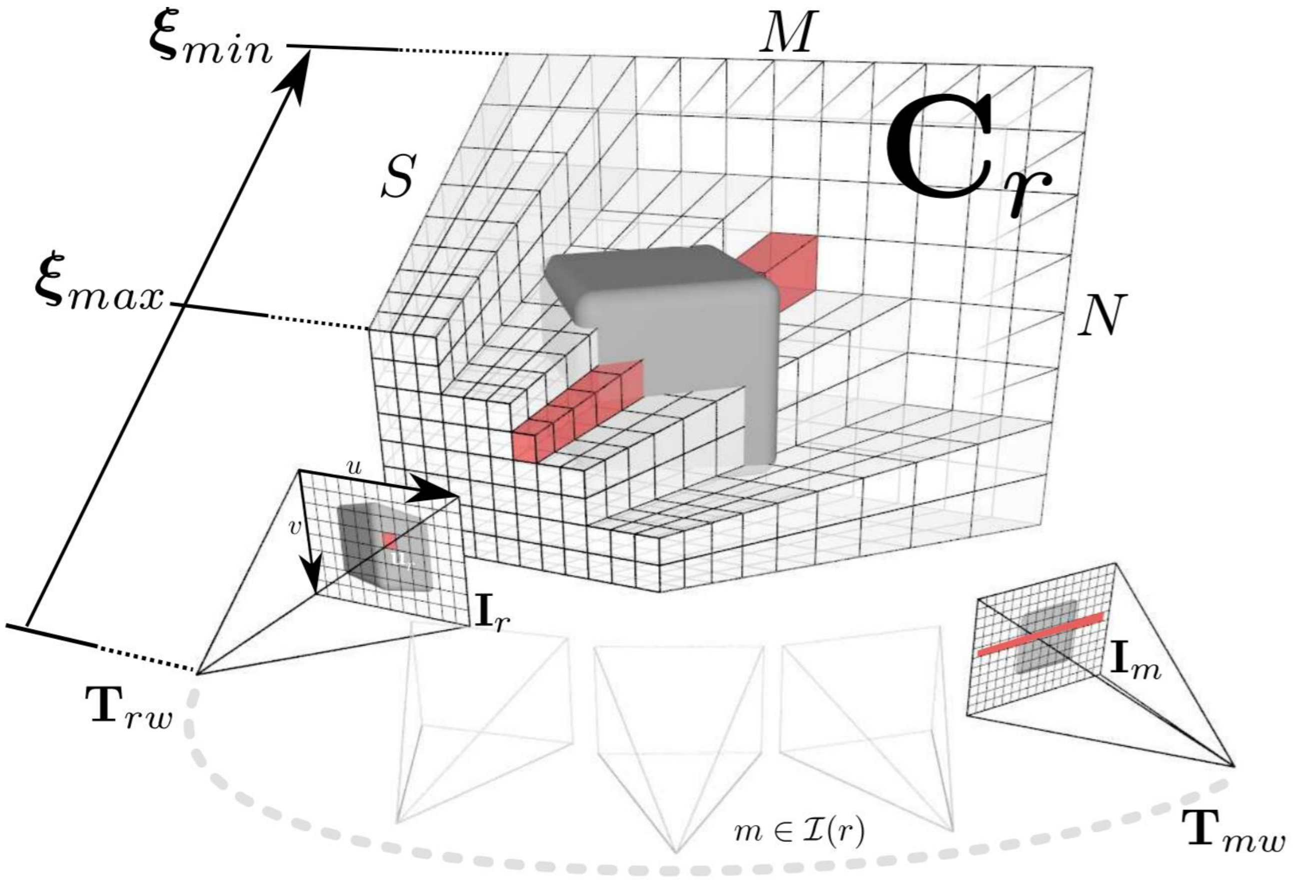
(a) 2 views (b) 5 views (c) 20 views (d) Ground Truth

[Newcombe, 2013]

Plane Sweep for Multiple view aggregation

- Compute the photo-metric data-term between a reference frame and all available frames
- Integrate photo-metric costs into a single (3D) voxel volume
- Use a Plane Induced Homography to efficiently transfer pixels:

$$\begin{pmatrix} u' \\ v' \\ 1 \end{pmatrix} = \pi\left(K\left(R - \frac{t \cdot n^{\mathrm{T}}}{d}\right)' K^{\mathrm{T}} \begin{pmatrix} u \\ v \\ 1 \end{pmatrix}\right)$$



[Extracted from “DTAM: Dense Tracking and Mapping in Real-Time. Newcombe, Lovegrove, Davison, 2011]

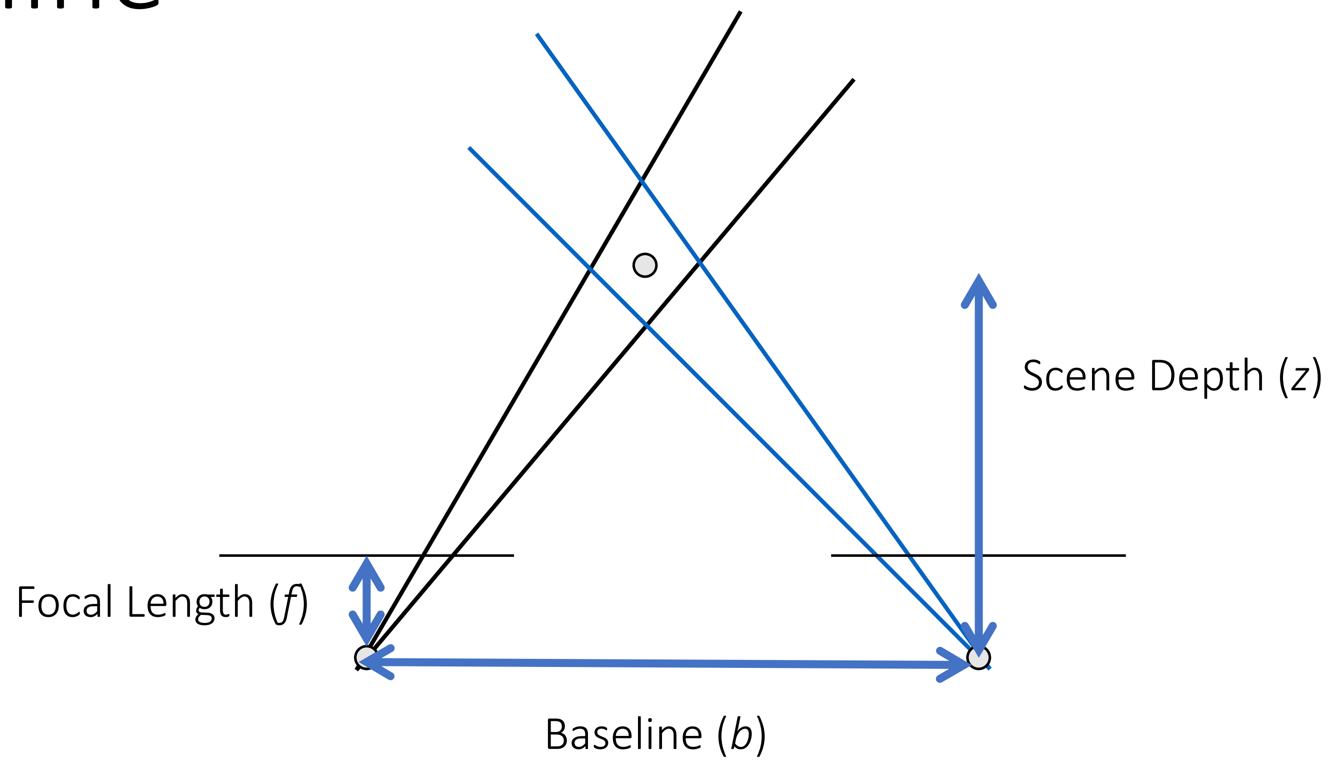
Switch to live coding demo of Plane Sweep

- And take a break!

Effect of Stereo Baseline

Recap Stereo Steps

- Calibrate cameras
- Compute disparity (d)
- Estimate depth (z)

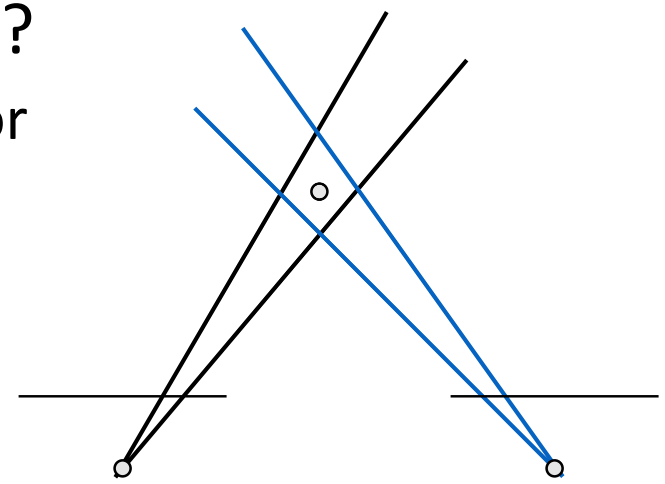


For *Rectified* Stereo: $$z = \frac{bf}{d}$$

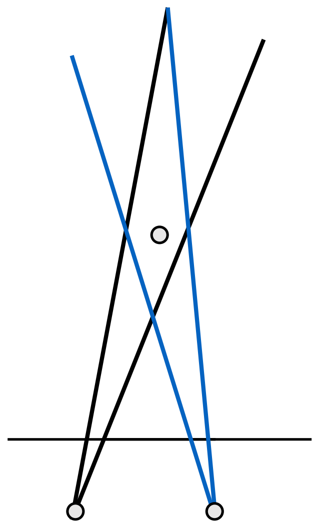
Choosing the Baseline

What's the optimal baseline?

- Too small: large depth error
- Too large: difficult search problem



Large Baseline



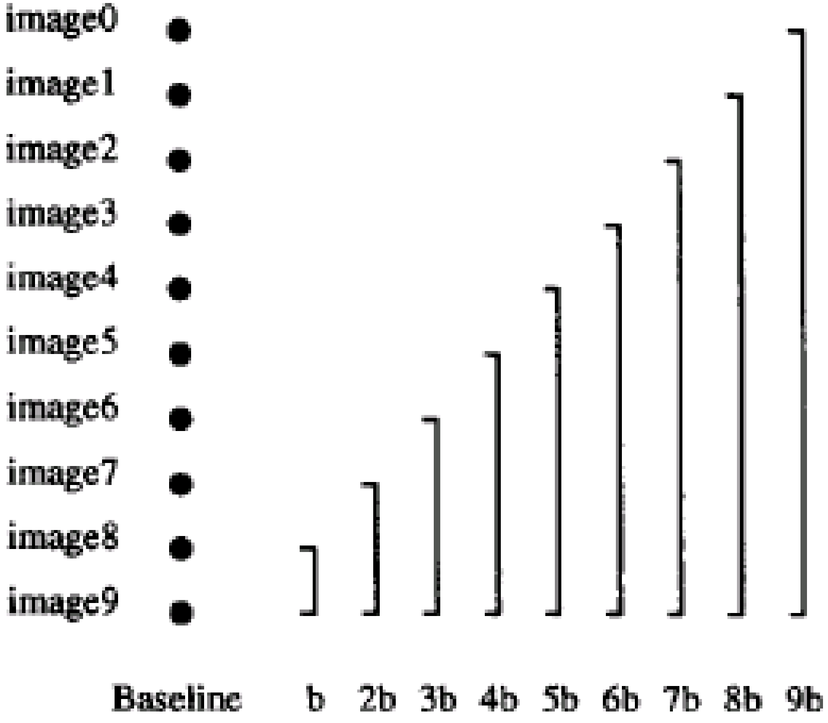
Small Baseline

Error in Z :

$$\epsilon_z = \frac{bf}{d} - \frac{bf}{d+\epsilon_d} = \frac{z^2\epsilon_d}{bf + z\epsilon_d} \approx \frac{z^2}{bf} \cdot \epsilon_d.$$

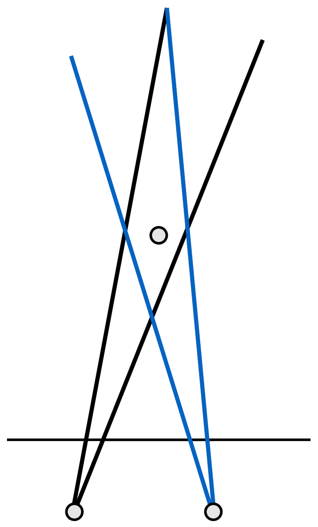
Effect of Baseline on Estimation

Figure 2: An example scene. The grid pattern in the background has ambiguity of matching.

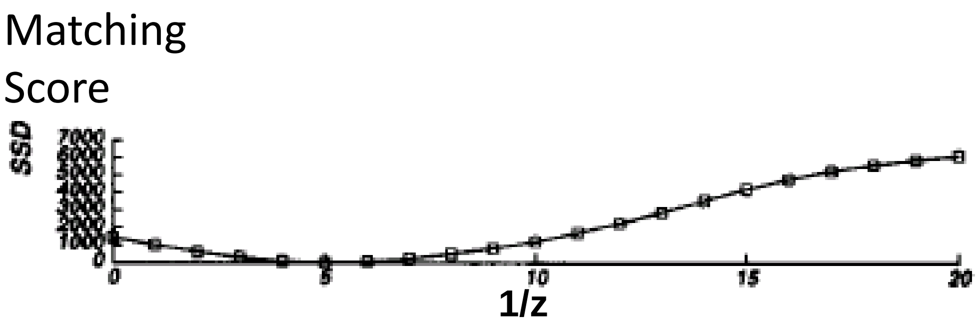


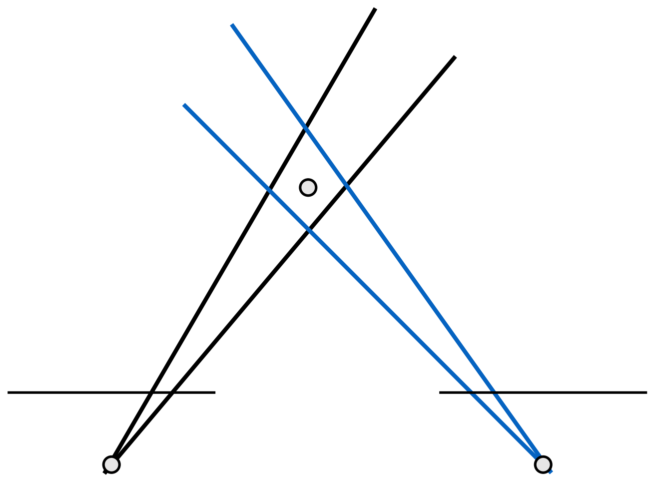
[Okutomi 1993]

Effect of Baseline on Estimation

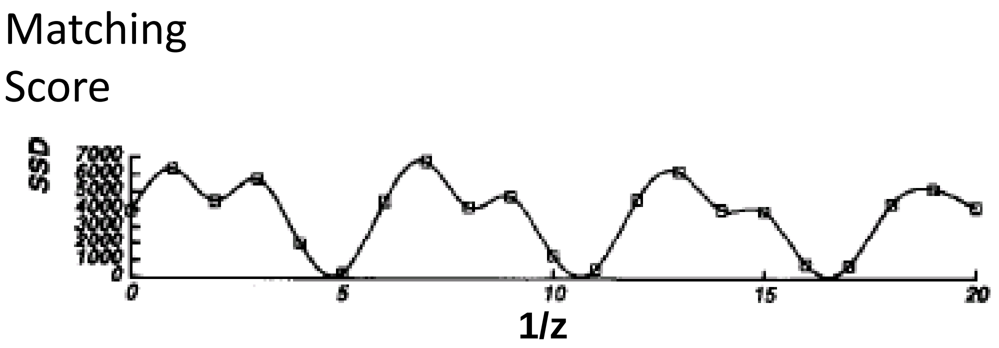


Small Baseline





Large Baseline



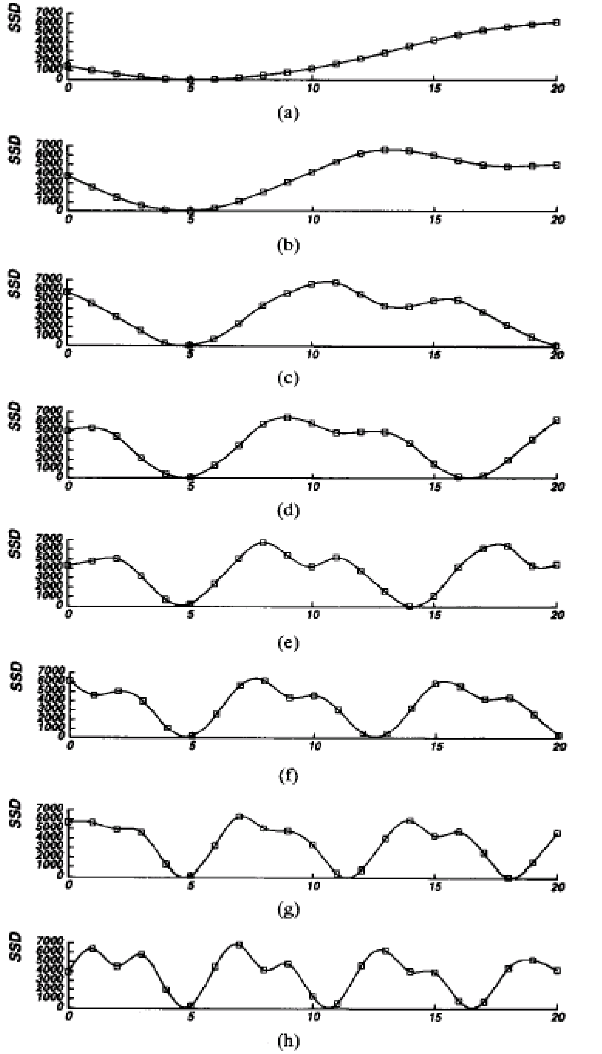


Fig. 5. SSD values versus inverse distance: (a) $B = b$; (b) $B = 2b$; (c) $B = 3b$; (d) $B = 4b$; (e) $B = 5b$; (f) $B = 6b$; (g) $B = 7b$; (h) $B = 8b$. The horizontal axis is normalized such that $8bF = 1$.

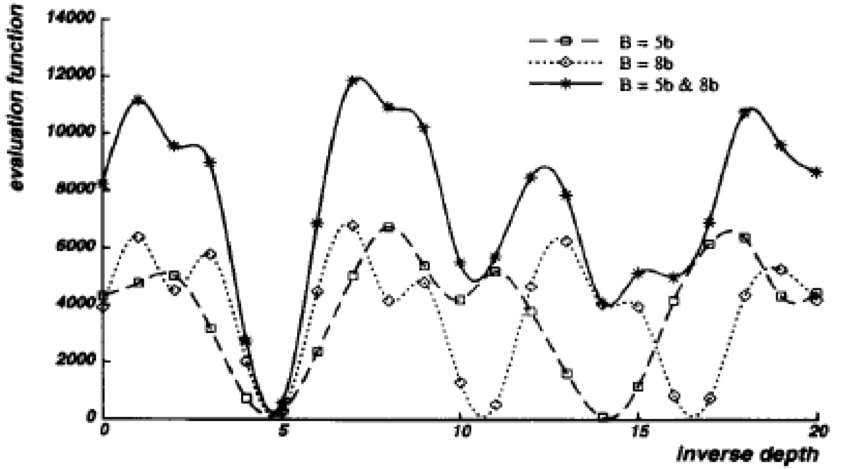


Fig. 6. Combining two stereo pairs with different baselines.

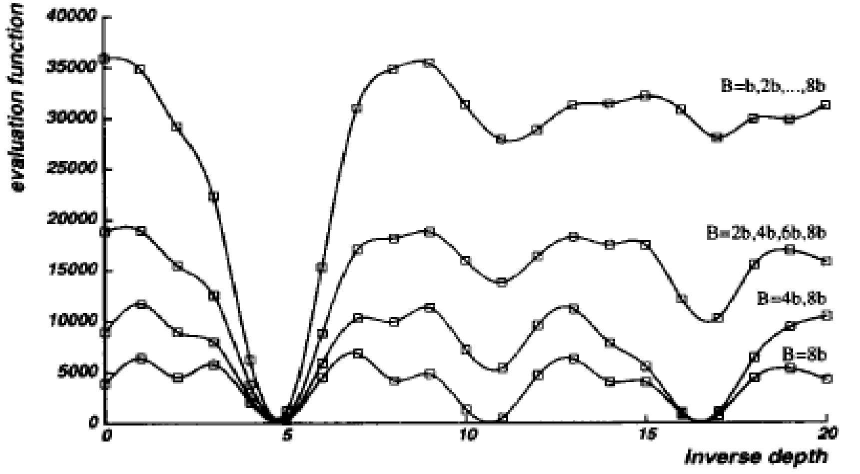


Fig. 7. Combining multiple baseline stereo pairs.

[Okutomi 1993]

Variable Baseline/Resolution Stereo

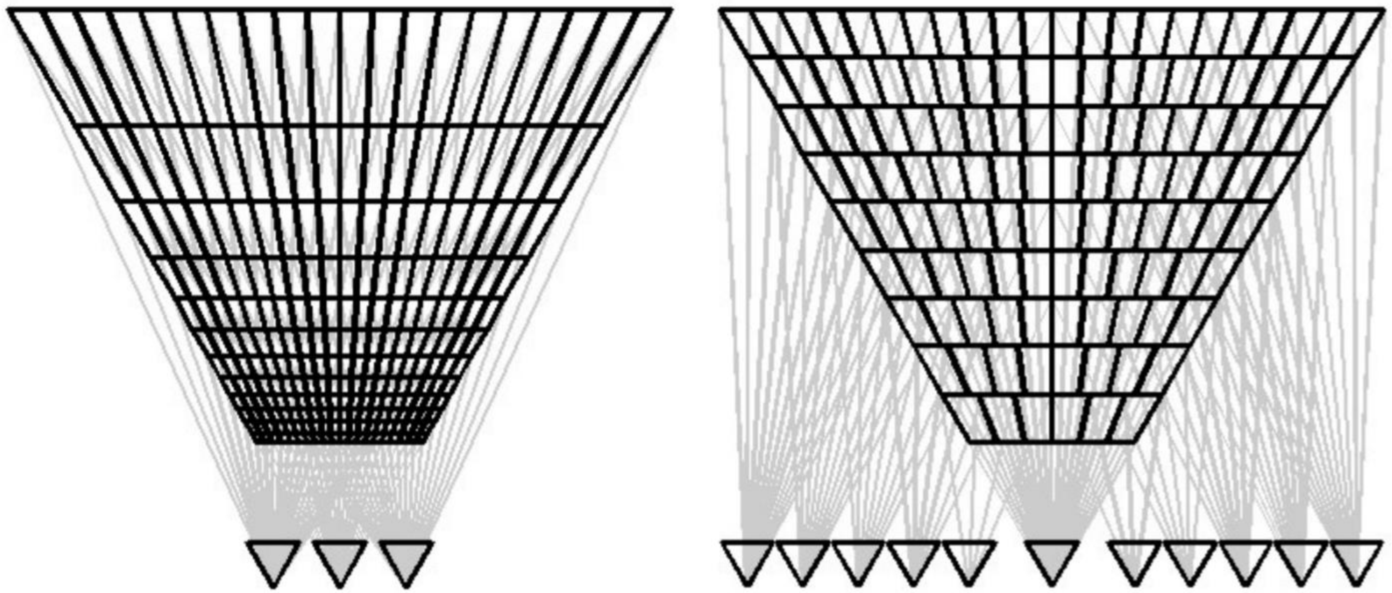


Figure 1. *Left*: Standard stereo. Note that the distance between depths increases quadratically. *Right*: Variable Baseline/Resolution Stereo. The distance between depths is held constant by increasing the baseline and selecting the appropriate resolution.

[Gallup et al]

Multiple Baseline Stereo

Basic Approach

- Choose a reference view
- Use your favorite stereo algorithm BUT
  - replace two-view SSD with SSD over all baselines
- Optimally chose a set of images to maintain a constant compute or error metric

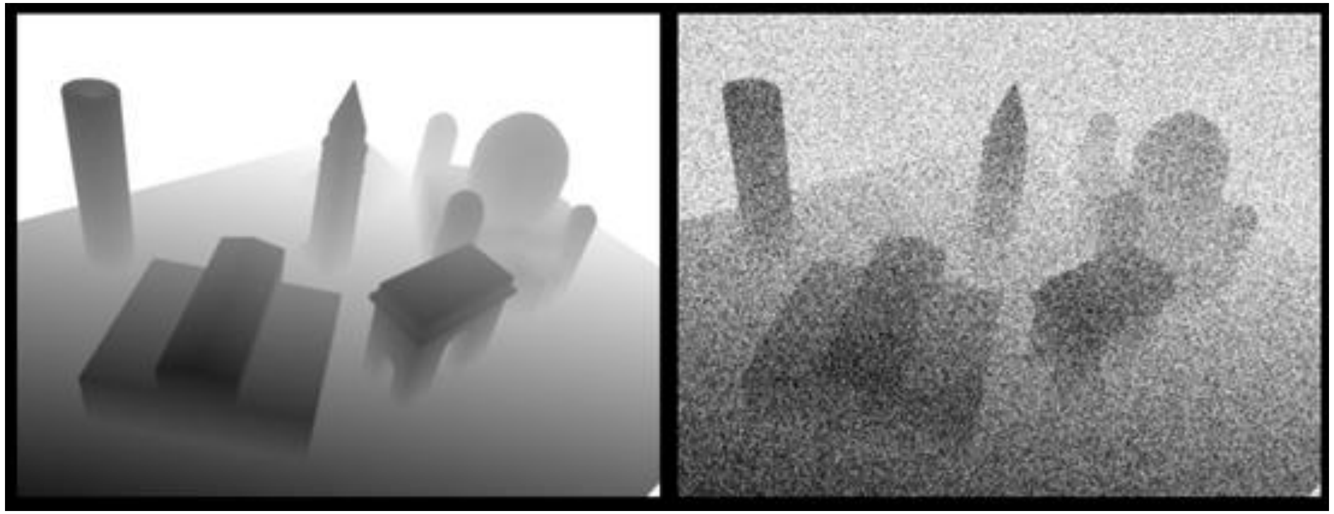
Limitations

- Which is the best reference view?
- Visibility: how to select which frames have scene co-visibility?
  [Kang, Szeliski, Chai, CVPR'01]

Image Modelling and Denoising

Estimating scene geometry with constraints

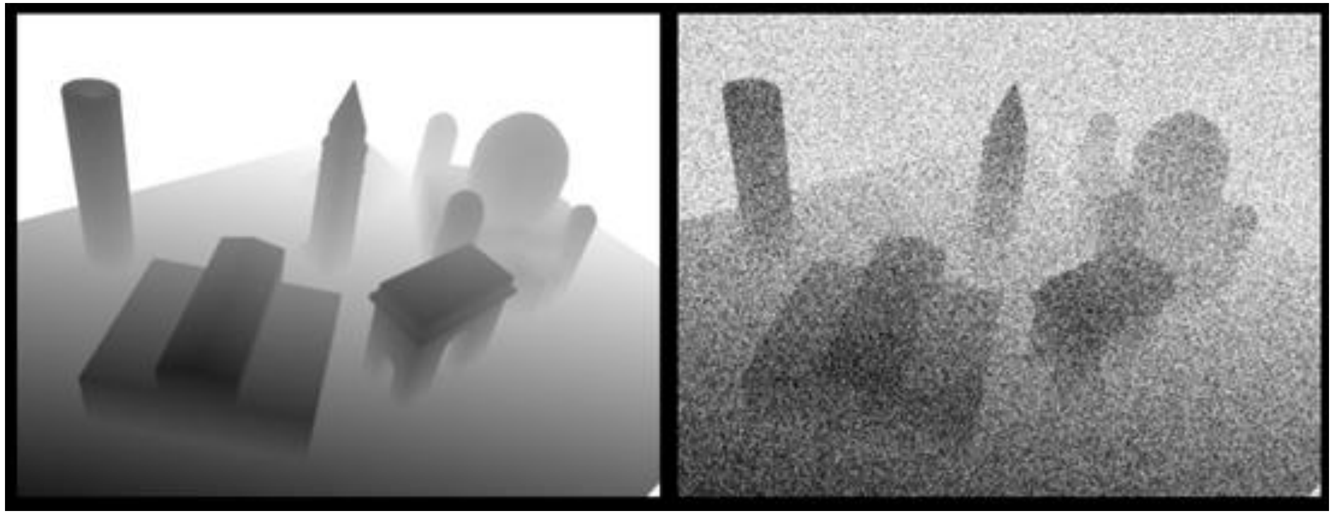
Denoising Data



(a) Model Depth $\mathcal{D}$ (b) $\mathcal{D} + \mathcal{N}(0, \mathbf{I}\sigma)$

Can we recover D given the noisy version?

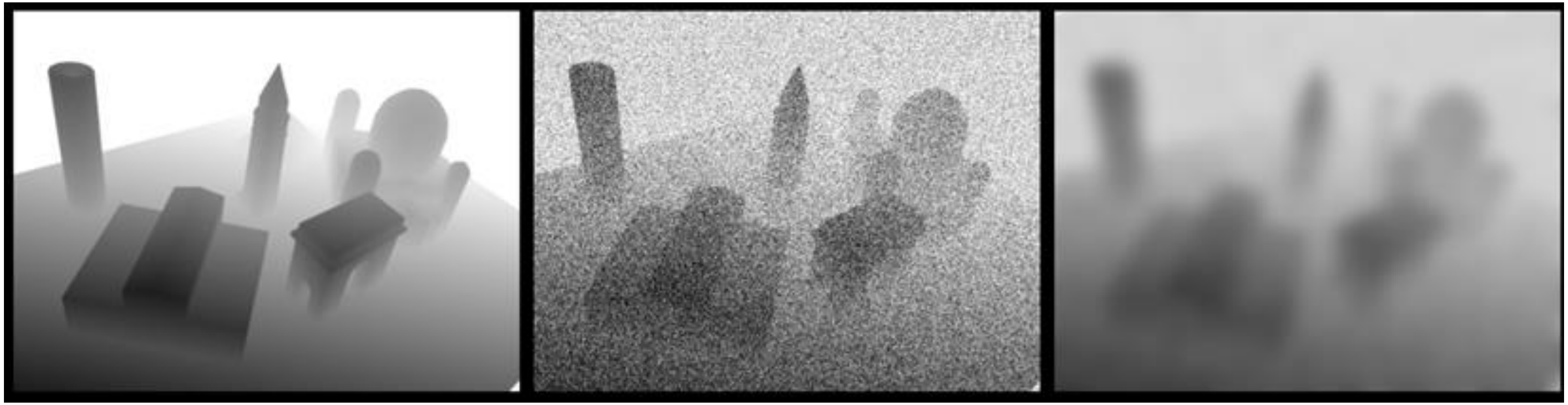
How to reduce the noise in the Depth Images?



(a) Model Depth $\mathcal{D}$ (b) $\mathcal{D} + \mathcal{N}(0, \mathbf{I}\sigma)$

Independent
Gaussian
Noise

Smoothing – e.g. apply image filtering?



(a) Model Depth $\mathcal{D}$ (b) $\mathcal{D} + \mathcal{N}(0, \mathbf{I}\sigma)$ Mean Filtered version of (b)

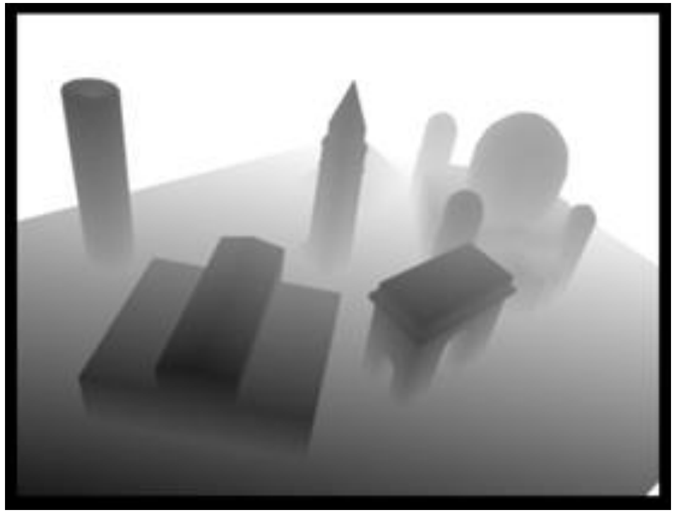
- Where are the sharp edges from the buildings?
- Mean filtering doesn't take into expected image structures

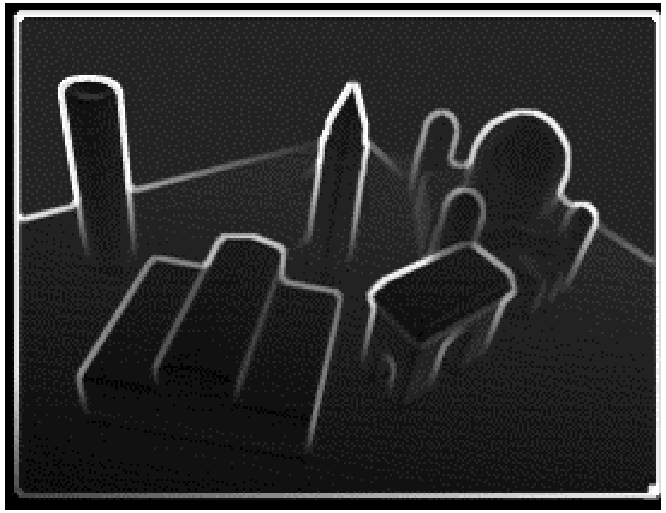
Gradients of Expected Depth Image

Depth Image

$\|\nabla \mathcal{D}(x,y)\|$

Model Depth $\mathcal{D}$

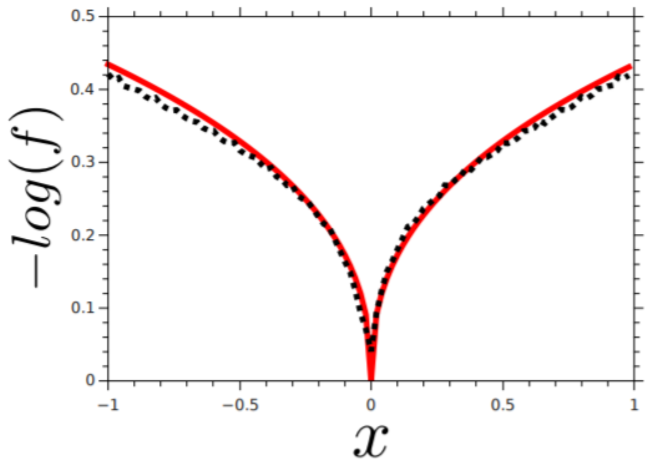




Mean Filtered version of Noisy Depth Image

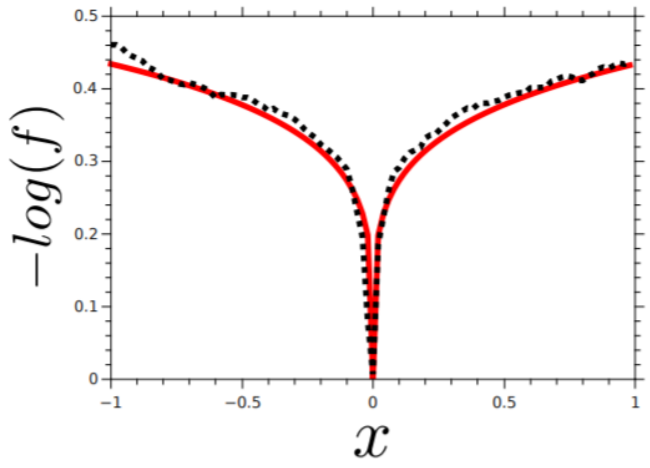
Priors: What do the images look like in gradient space?

- We can use statistics of image derivatives in *expected* data
- E.G what is the distribution of image gradients in a passive or depth image?



$$\alpha = 0.8$$

Histogram of –Log for the Image gradient dI/dx for visible light



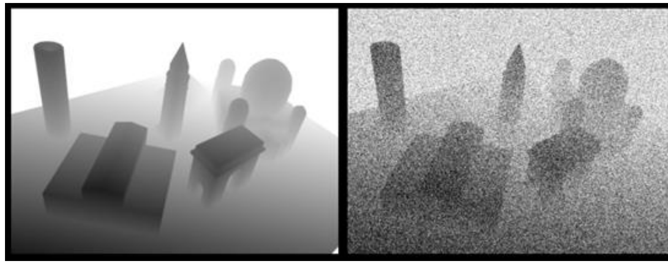
$$\alpha = 0.2$$

Histogram of –Log for the Image gradient dD/dx for Depth Image of the scene

Generalized Gaussian Distribution:

$$f(x) \propto \exp\left(-\frac{|x-\mu|^{\alpha}}{\alpha\sigma^{\alpha}}\right)$$

The probability of the depth image?



(a) Model Depth $\mathcal{D}$ (b) $\mathcal{D} + \mathcal{N}(0, \mathbf{I}\sigma)$

Data Term Likelihood assuming Independent Gaussian Noise:

$$p(g|\mathcal{D}) = \prod_{(x,y)\in\Omega} \frac{1}{\sqrt{2\pi\sigma}} \exp\left(-\frac{(g(x,y) - \mathcal{D}(x,y))^2}{2\sigma^2}\right)$$

Smoothness Prior w/ Gaussian Dist. Over 1st Order Image Gradients:

$$p(\mathcal{D}) = \prod_{(x,y)\in\Omega} \frac{1}{\sqrt{2\pi\nu}} \exp\left(-\frac{\|\nabla\mathcal{D}(x,y)\|^2}{2\nu^2}\right)$$

The probability of the depth image?

Recap Bayesian Inference given a distribution Likelihood and Prior:

$$p(\mathcal{D}|g) = \frac{p(g|\mathcal{D})p(\mathcal{D})}{p(g)} \quad \propto p(g|\mathcal{D})p(\mathcal{D})$$

Searching for the maximum a posteriori estimate Depth Image:

$$\hat{\mathcal{D}} = \underset{g}{\arg\max} \{p(\mathcal{D}|g)\}$$

$$\hat{\mathcal{D}} = \underset{g}{\arg\max} \{p(g|\mathcal{D})p(\mathcal{D})\}$$

$$\hat{\mathcal{D}} = \underset{g}{\arg\max} \left\{ \frac{1}{4\pi\mu\nu} \prod_{(x,y)\in\Omega} \exp\left(\frac{(g(x,y) - \mathcal{D}(x,y))^2}{\sigma^2} + \frac{|\nabla\mathcal{D}(x,y)|^2}{\nu^2} \right) \right\}$$

Depth Denoising by Energy Minimization

Transform to Energy, E(D), minimization problem using –Log:

$$\mathsf{E}(\mathcal{D}) \;=\; -\ln p(\mathcal{D}|g) \propto -\ln p(g|\mathcal{D}) - \ln p(\mathcal{D})$$

Energy, E(D):

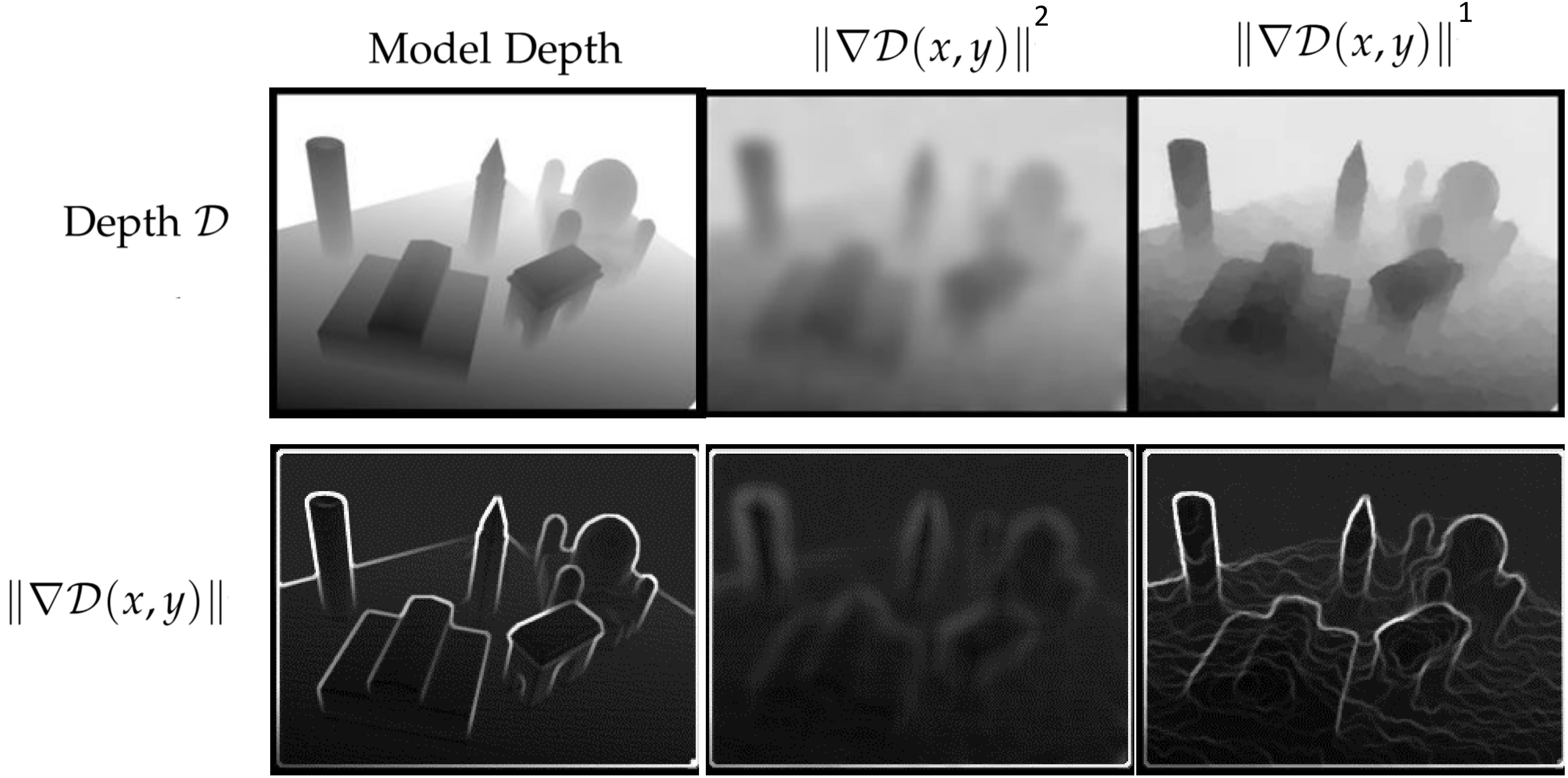
$$\hat{\mathcal{D}} \;=\; \min_{\mathcal{D}} \left\{ \sum_{(x,y)\in\Omega} \left(\frac{1}{2}((g(x,y) - \mathcal{D}(x,y))^2 + \frac{1}{2\lambda}\|\nabla\mathcal{D}(x,y)\|^2 \right) \right\}$$

Where λ combines factors relating to the variances ν^2 , σ^2.

Solve this minimization problem using:
→ Gradient Descent, Quasi Newton Methods or Discrete Optimization techniques

[Newcombe, 2013]

Comparison of Image Priors in Denoising



Correspondence as Global Energy Minimization

- This denoising approach can be directly applied to correspondence (e.g. depth):

$$E(d) = \underbrace{E_d(d)}_{\text{Match Cost (Data Term)}} + \lambda \underbrace{E_s(d)}_{\text{Smoothness Cost (Priors over solution space)}}$$

Want each pixel to find a good match in the other image

Bias the search towards realistic solutions

- Many matching costs and Priors see [Scharstein & R. Szeliski] [D. Scharstein, Middlebury Stereo Evaluation, www.middlebury.edu/stereo] :
- Can be computationally expensive
  - There are reasonable alternatives to Full Global Optimization, see [Hirshmüller CVPR05]

Scene Reconstruction

From Image space depth to complete geometric models

Problem: How to Combine Depth Images into a Complete Model?

(a) Measurement (b) 2 Frames (c) 30 Frames (d) 100 Frames (e) Complete model

[Extracted from KinectFusion. Newcombe et al, 2011]

Merging depth maps

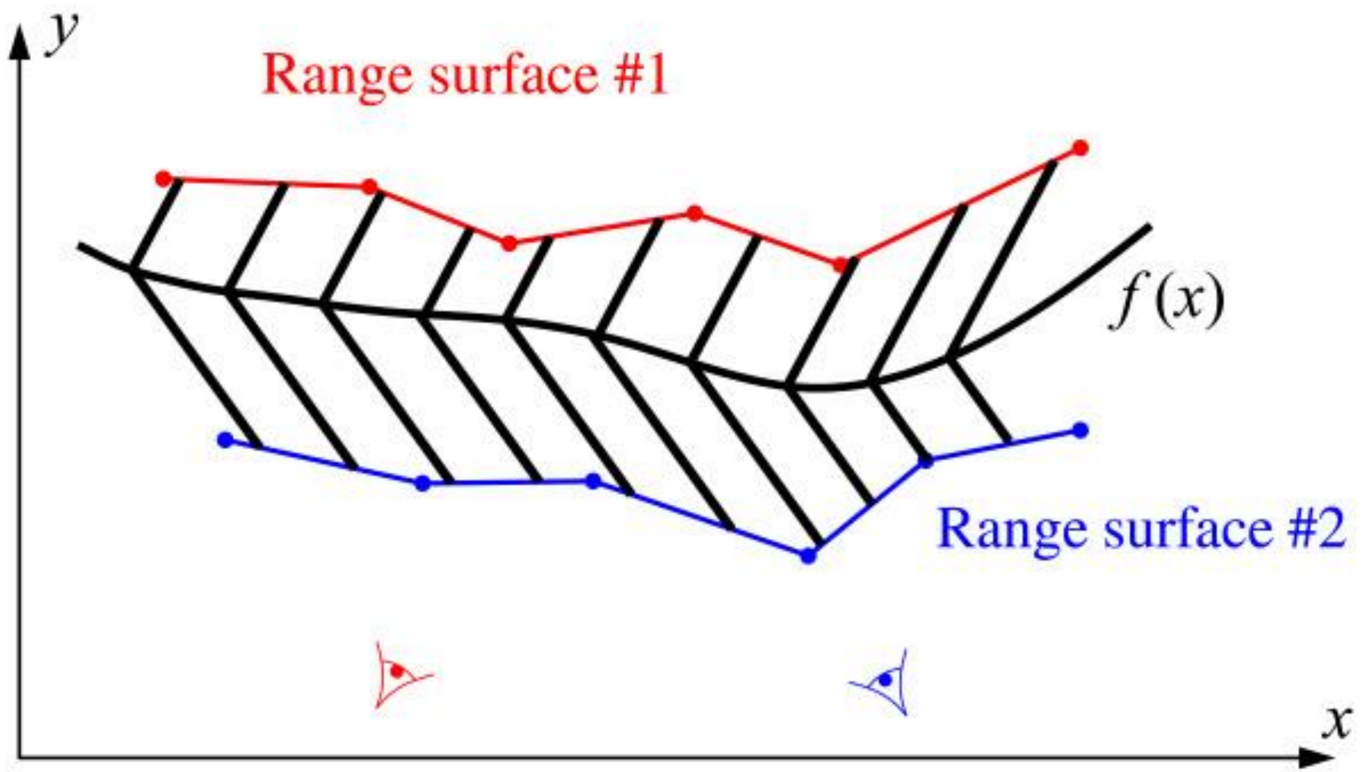
Depth map 1 Depth map 2 Combination (Union)

Reconstructed
Surfaces

- Naïve combination (union) produces artifacts
- Better solution: find “average” surface
  - ➔ Surface that minimizes sum (of squared) distances to the depth maps

[From Curless & Levoy, 1996]

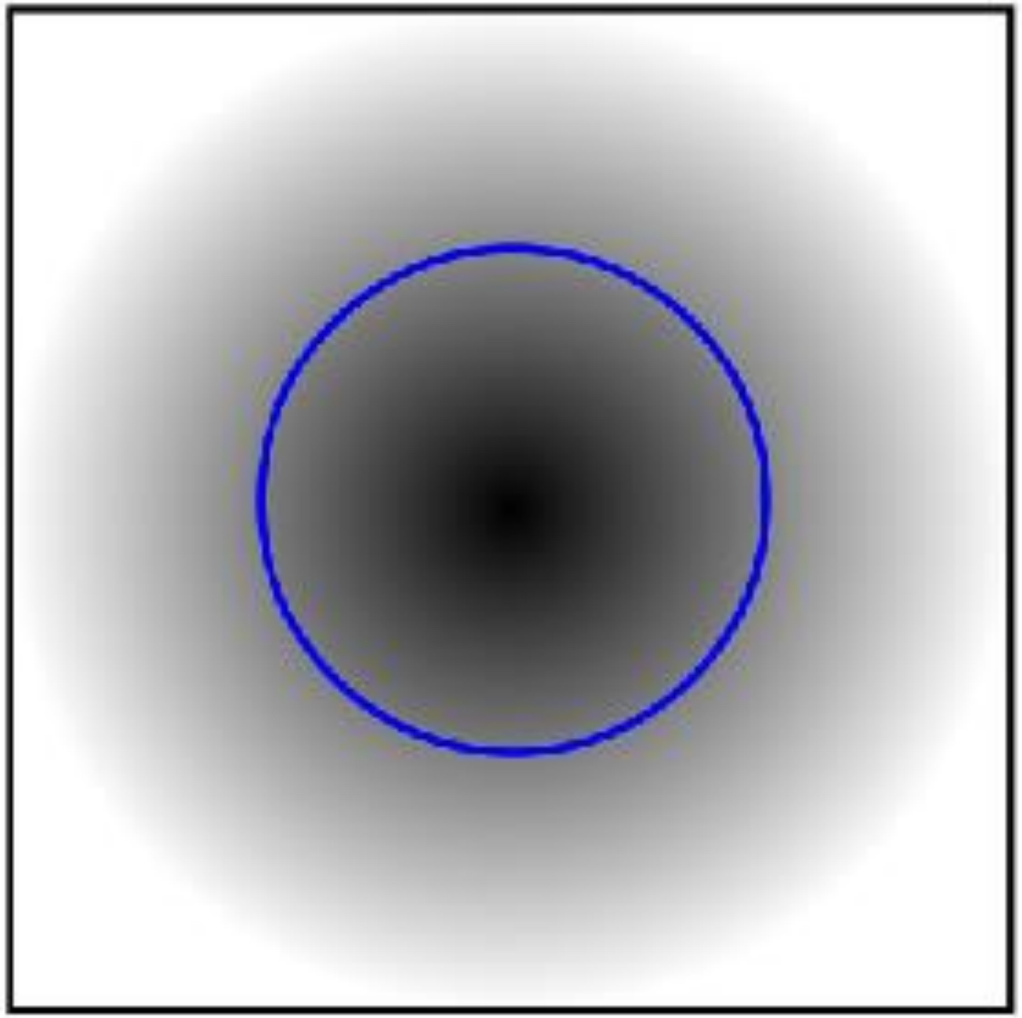
Least squares surface solution



$$E(f) = \sum_{i=1}^{N} \int d_i^2(x, f)\,dx$$

[Slide from Seitz, UW CSEP576]

Representing Geometry Implicitly

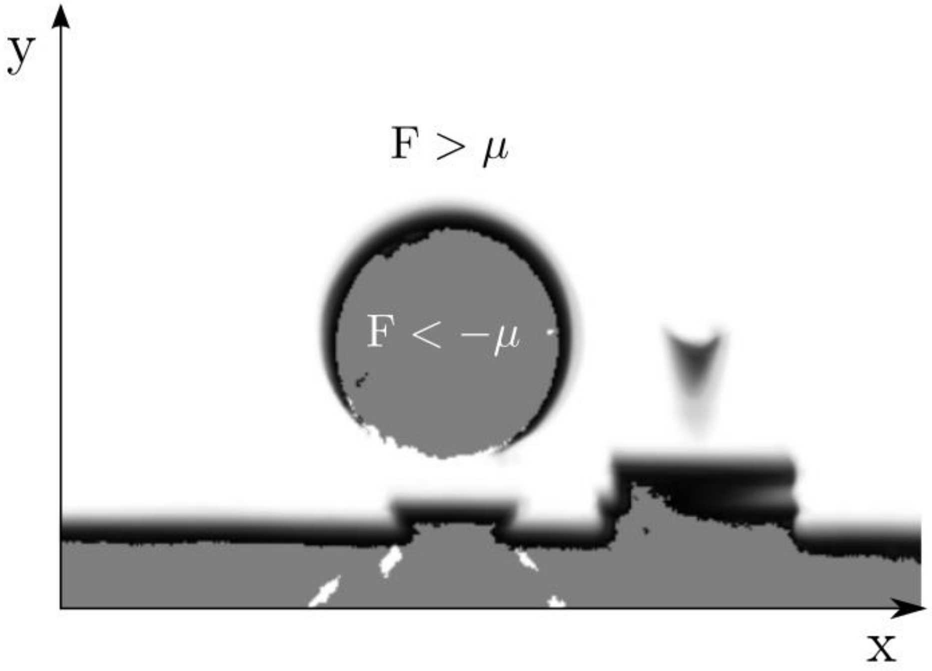


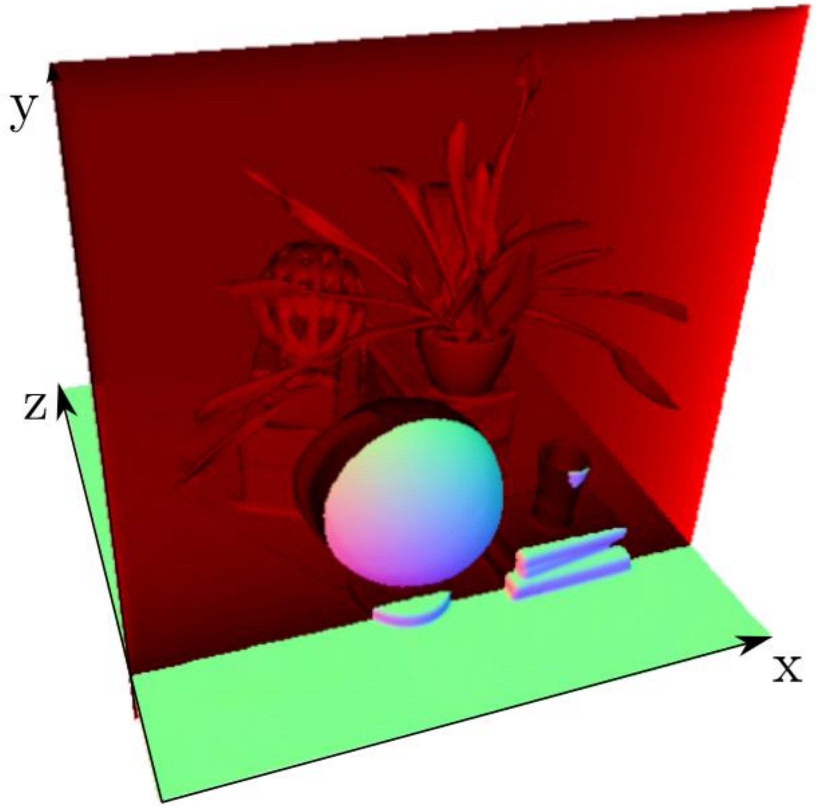
Signed Distance Functions

Example: Truncated Signed Distance Function (TSDF)

[Newcombe, 2015]

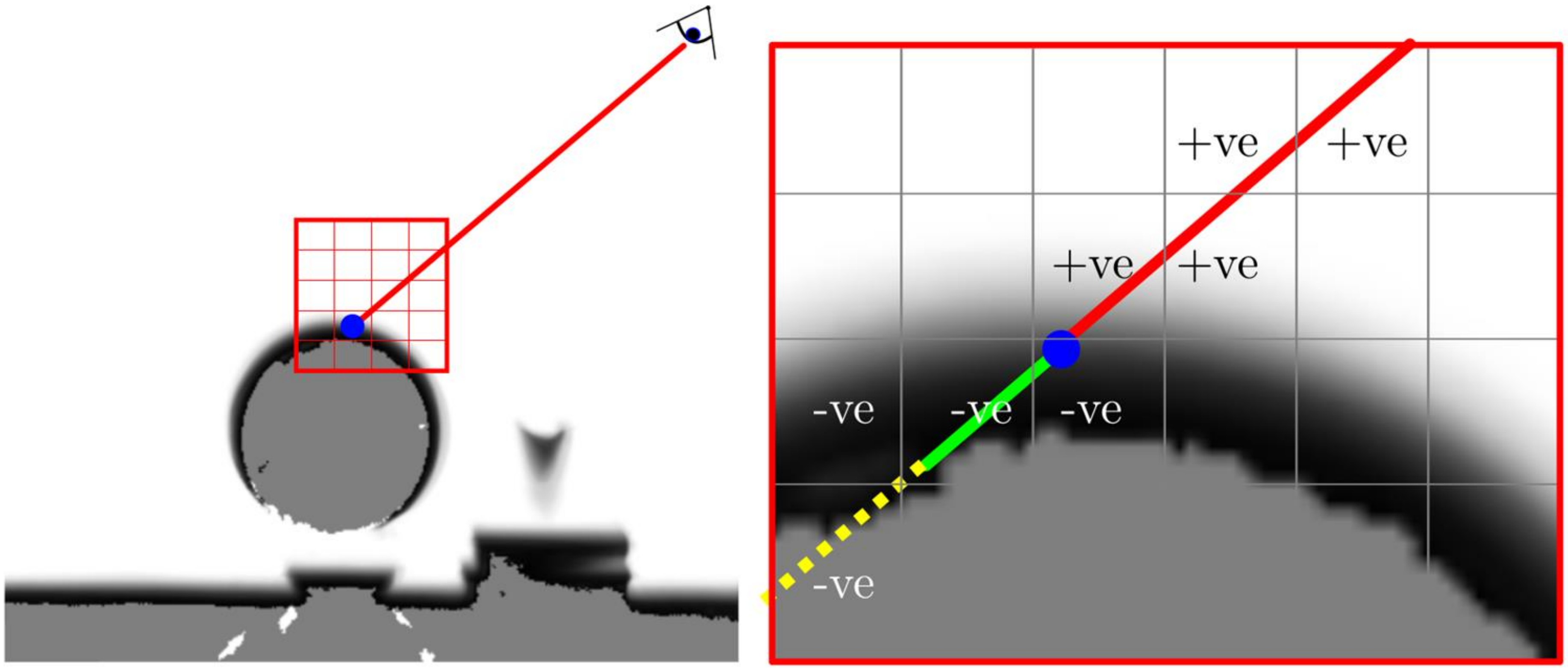
Representing Scenes with TSDF



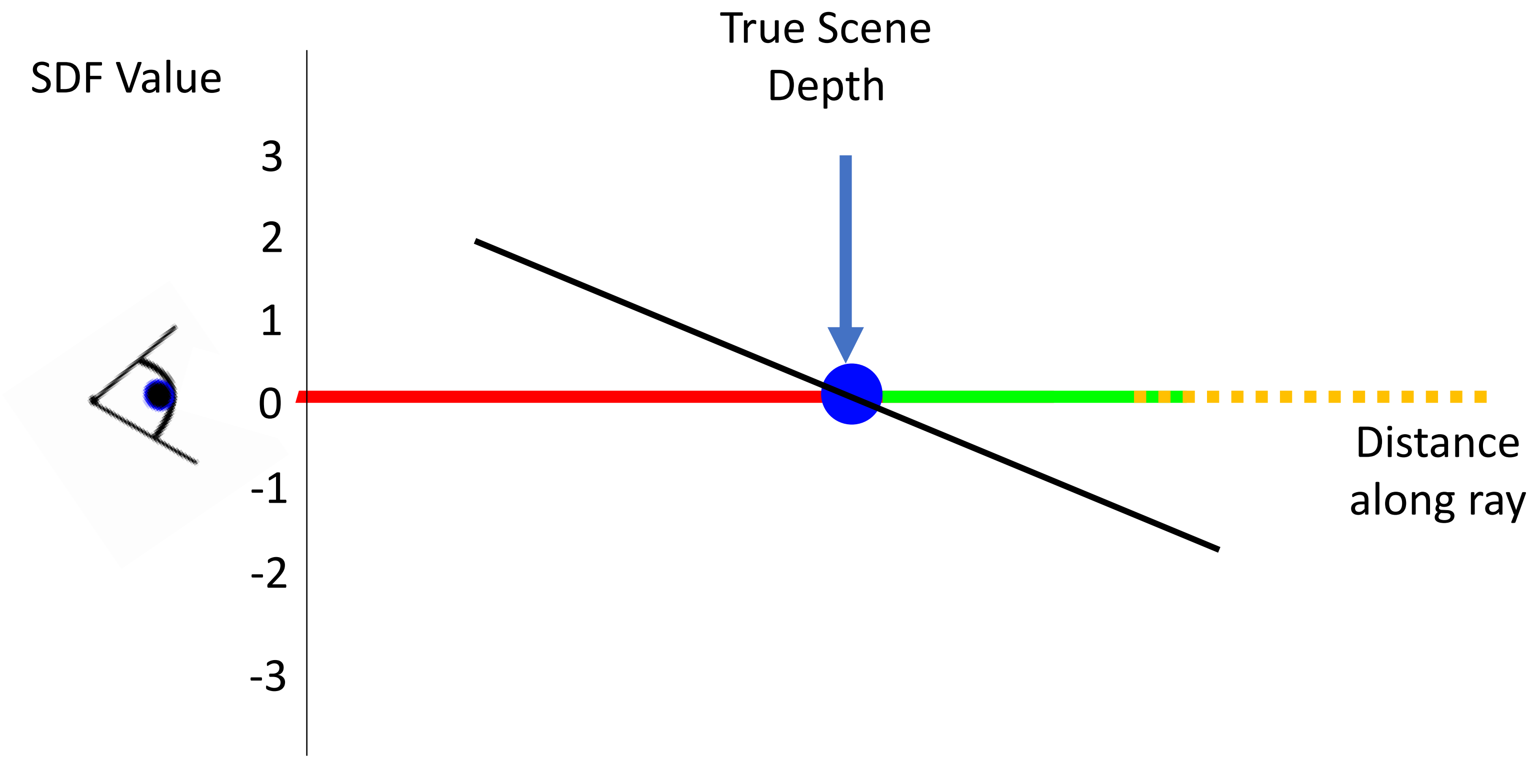


[KinectFusion, Newcombe et al, 2011]

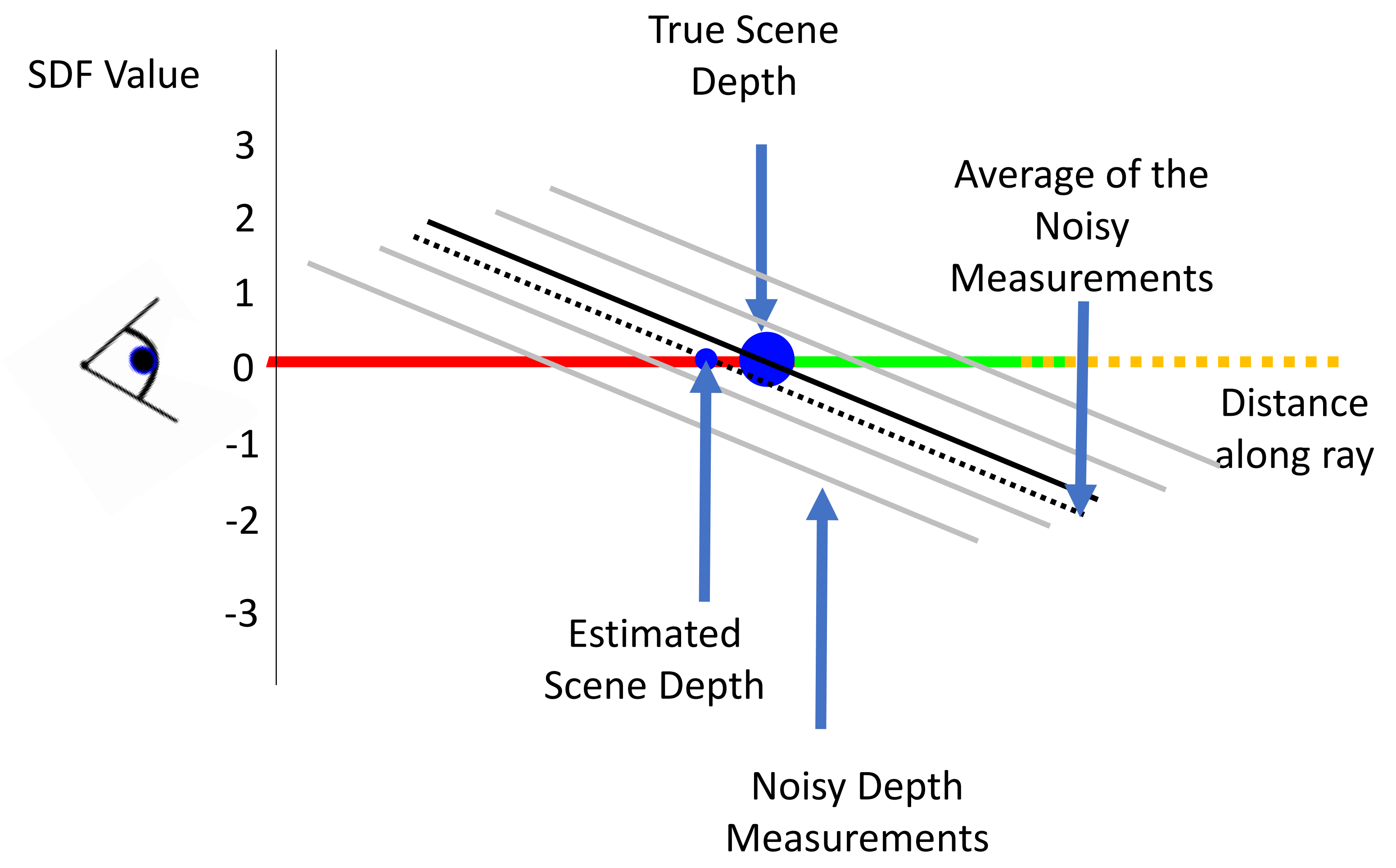
A Single Ray Observation in TSDF



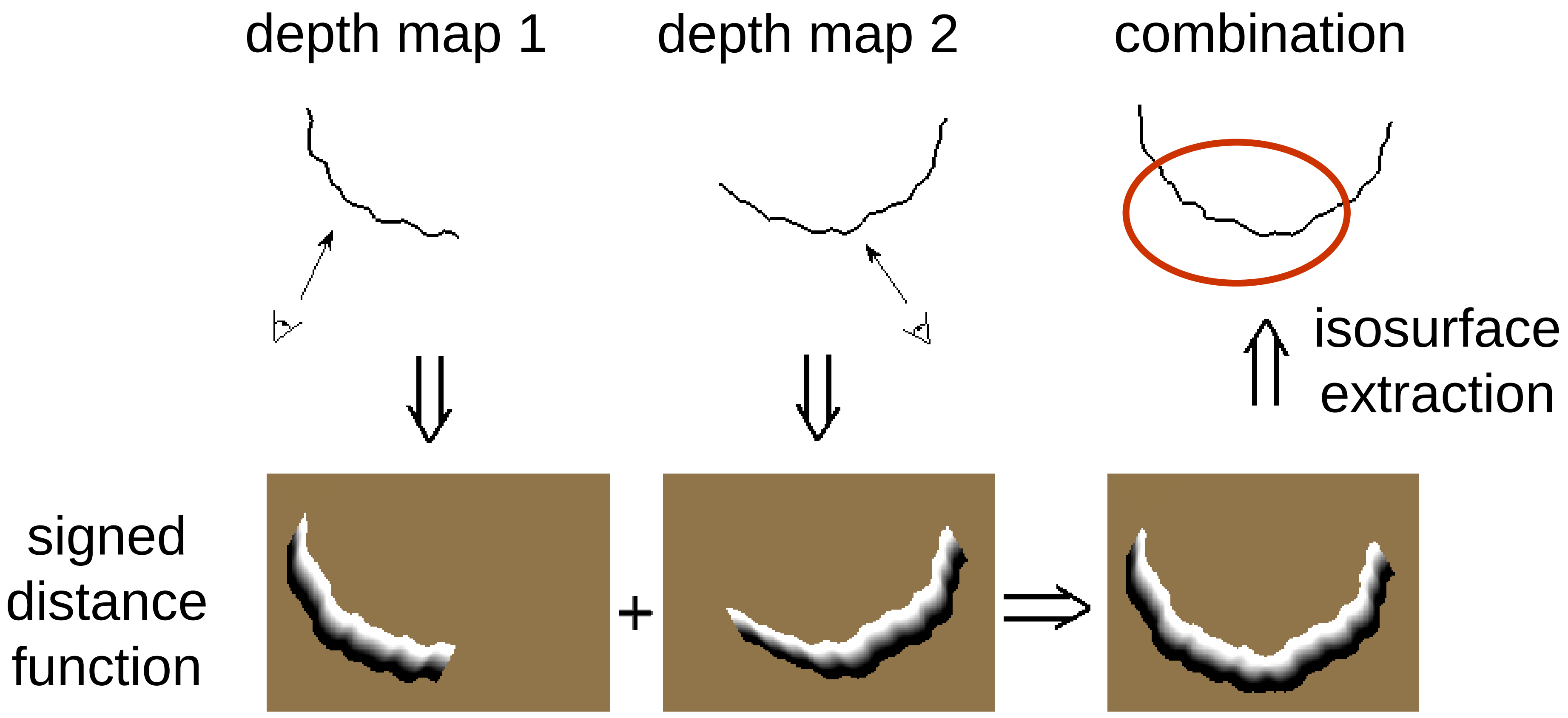
Ray Observations in TSDF



Fusing Noisy Ray Observations in TSDF



VRIP [Curless & Levoy 1996]



Merging Depth Maps: Temple Model

input image

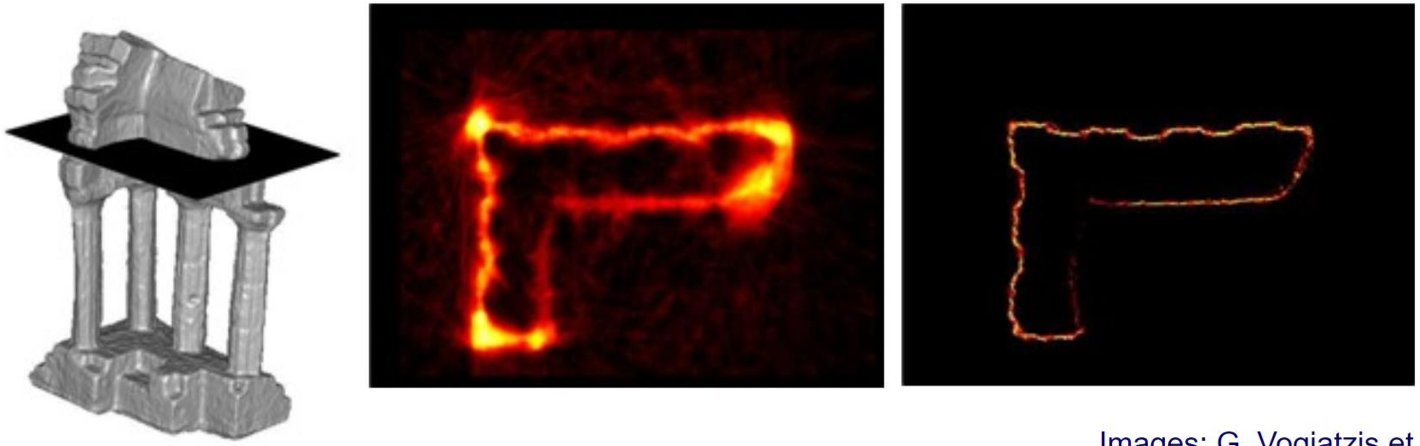
317 images
(hemisphere)

ground truth model

Goesele, Curless, Seitz, 2006

Global Photometric Volume Optimization

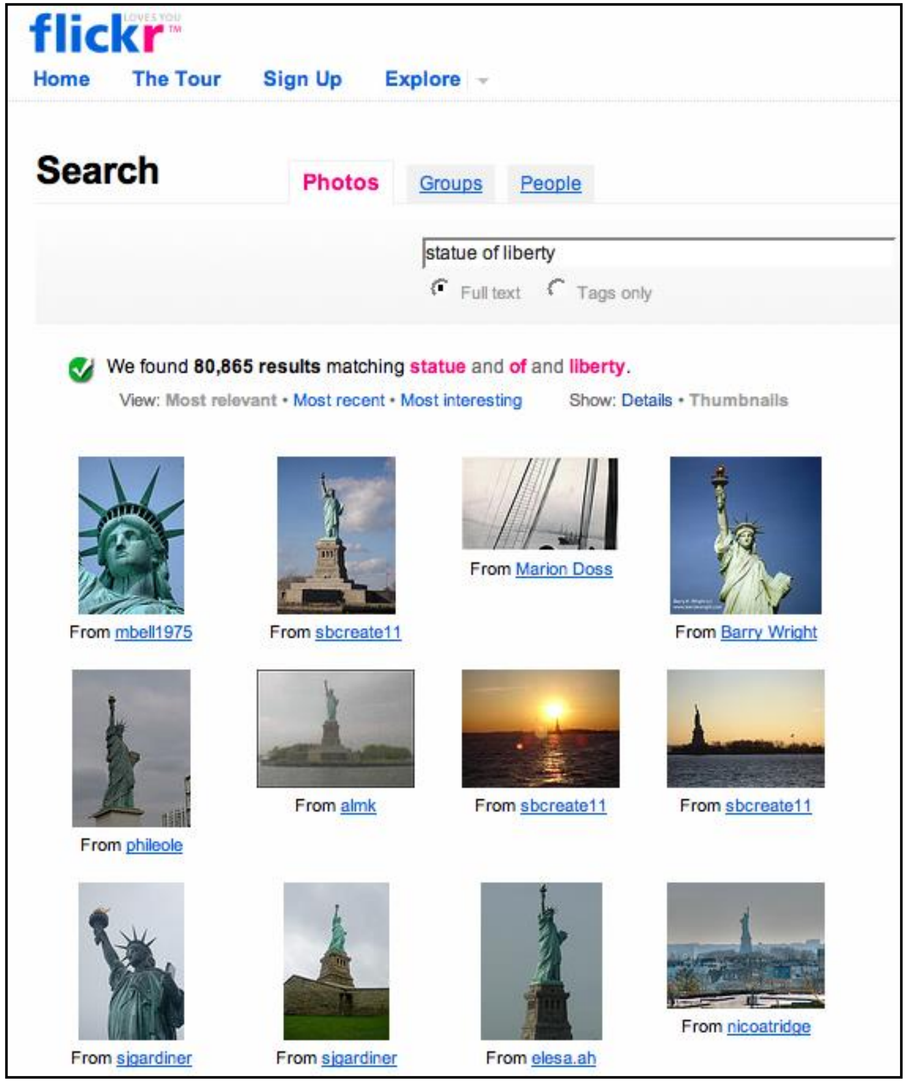
- Instead of fusing noisy depth maps into a volume
- Compute the photo-metric data-term for co-visible pairs of frames
- Integrate the photo-metric costs into a single (3D) voxel volume
- Define the total cost function with a surface regularization term
- Minimize the Global cost of the 'surface' that passes through the volume



Images: G. Vogiatzis et al.

Application: Multi-view stereo from Internet Collections

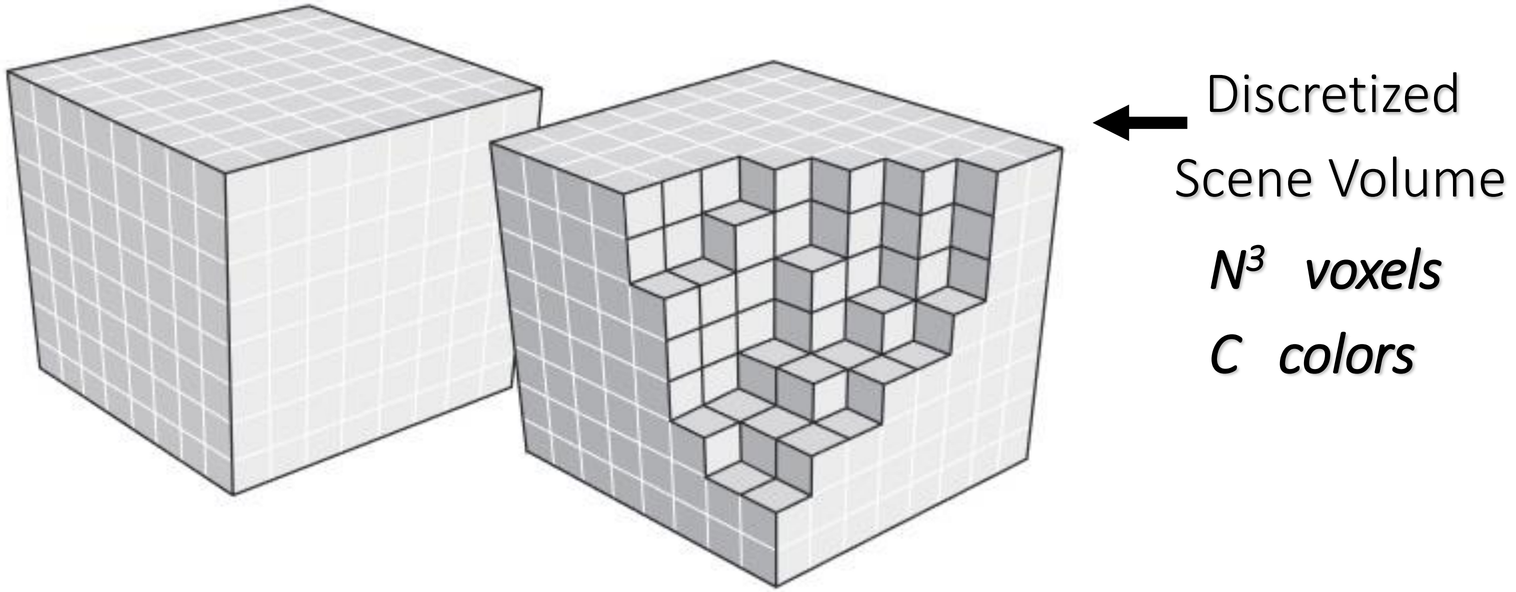
[Goesele, Snavely, Curless, Hoppe, Seitz, ICCV 2007]

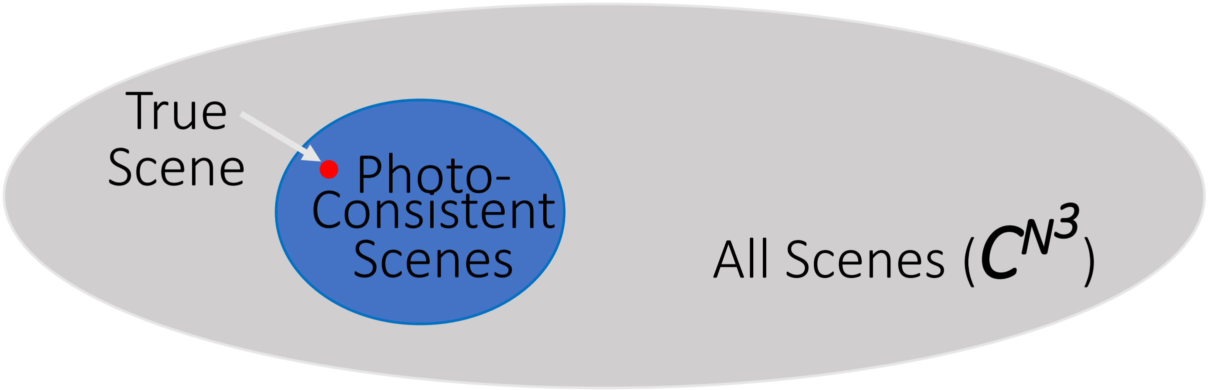


Voxel Coloring Algorithms [Seitz & Dyer]

- The space of possible Volumetric Scene Reconstructions

- These Approaches obtain voxel Coloring that ‘generate’ the observed images

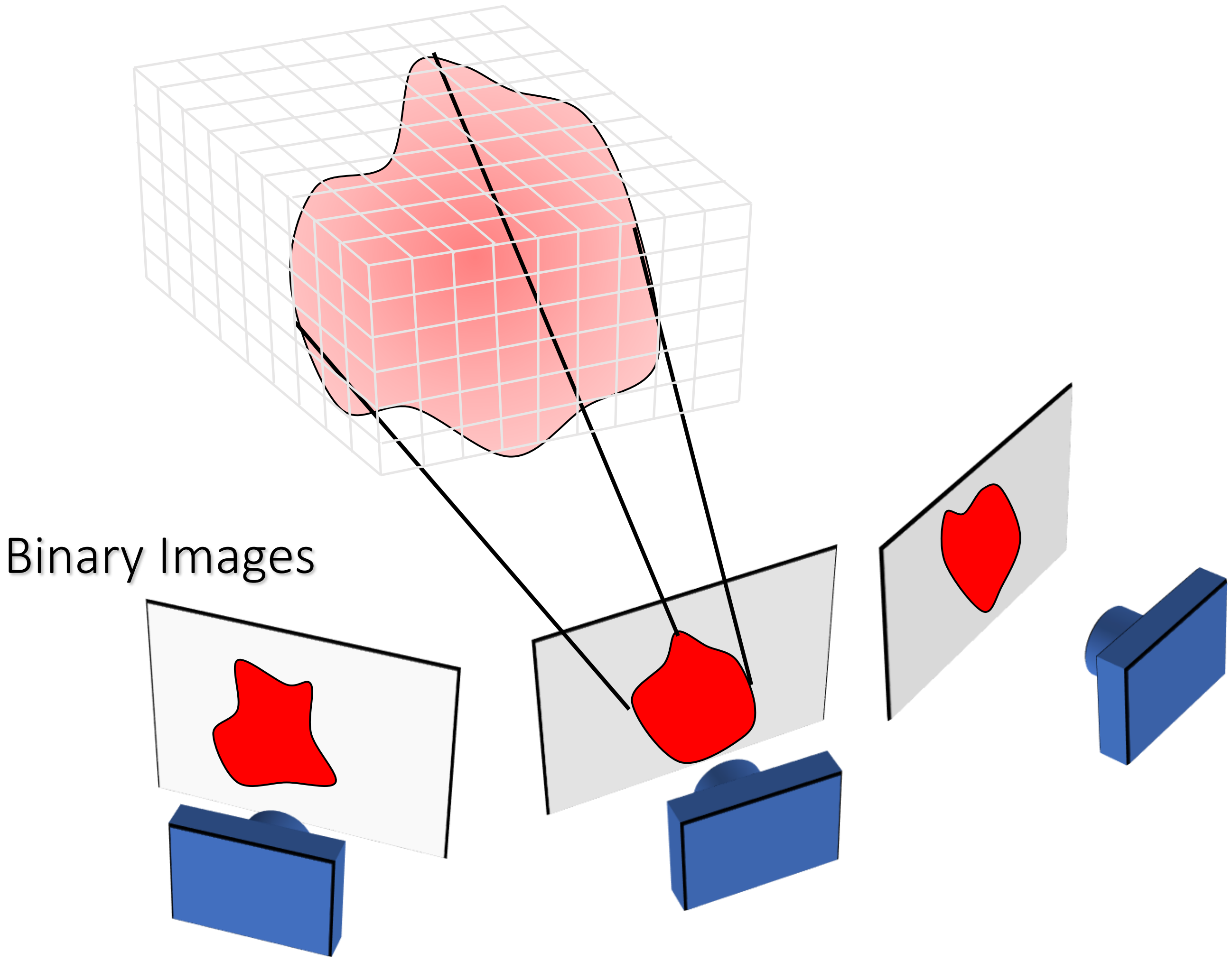




[Slide from Steve Seitz]

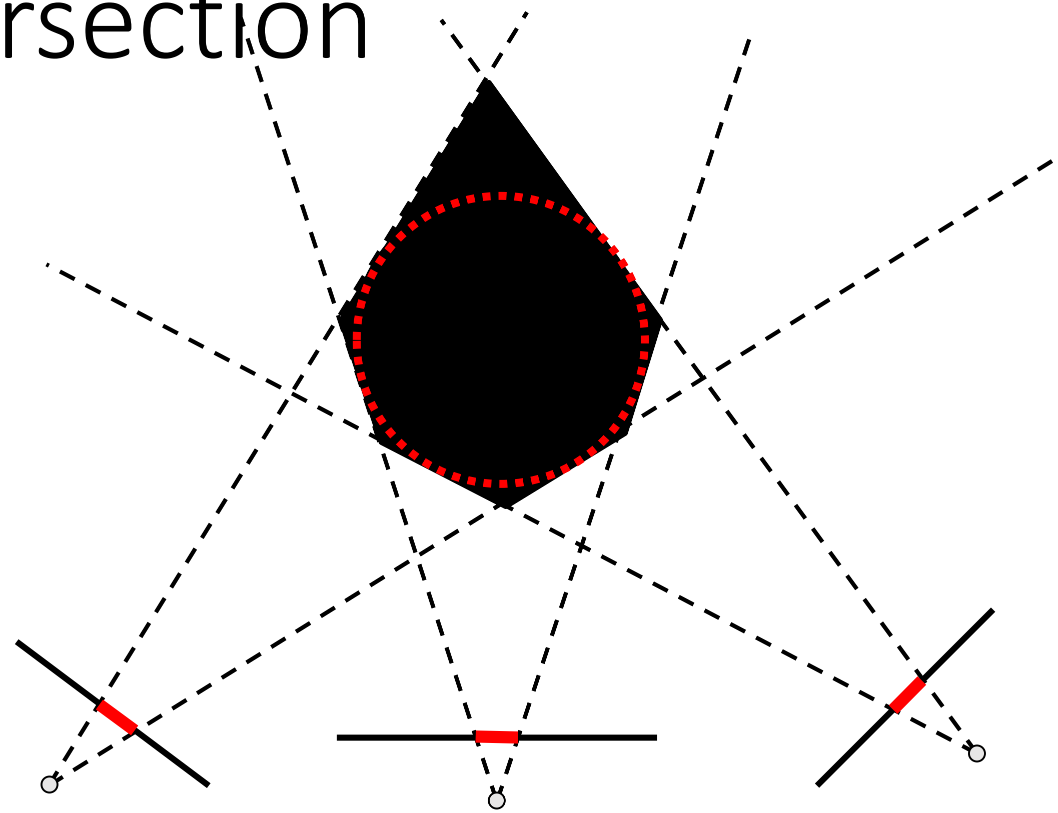
Example: Reconstruction from Silhouettes (**C=2**)

- *Back-project* each silhouette
- Intersect back-projected volumes
- How can we get Shape Silhouettes?



[Slide from Steve Seitz]

Volume intersection

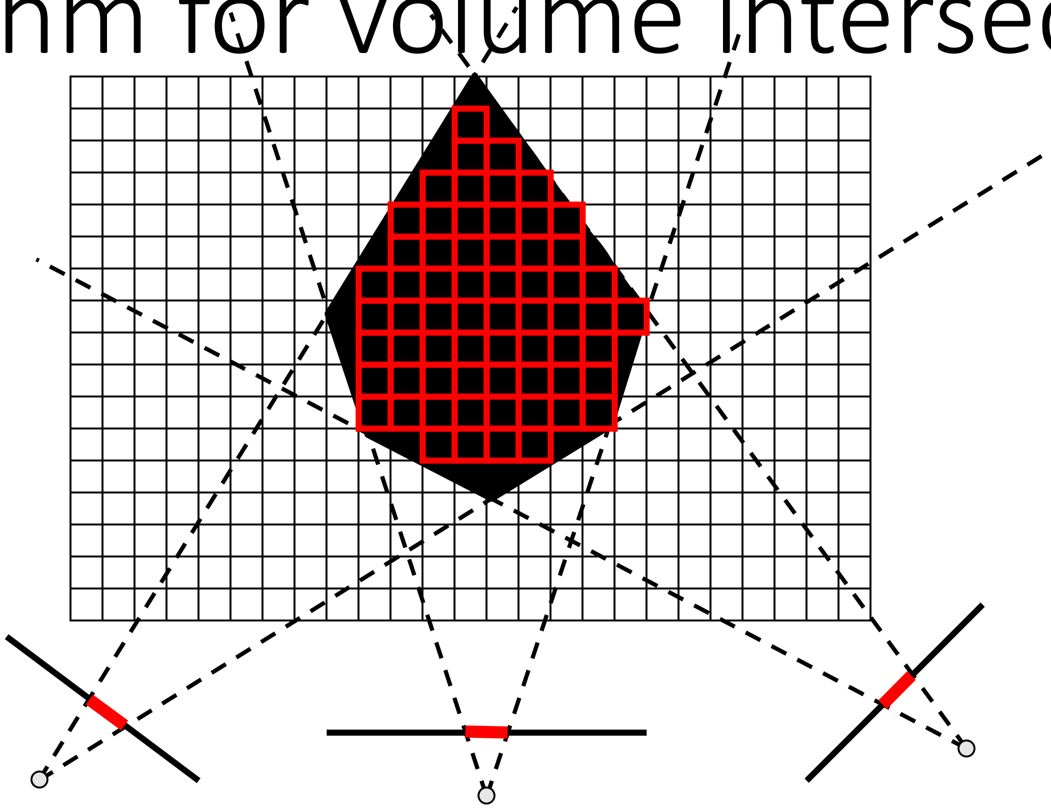


Reconstruction Contains the True Scene

- But is generally not the same
- In the limit (all views) get *visual hull*

[Slide from Steve Seitz]

Voxel algorithm for volume intersection



Color voxel black if on silhouette in every image

- O(?), for M images, N^3 voxels
- Don't have to search 2^{N^3} possible scenes!

- *Useful for reconstructions from Green Screen*

[Slide from Steve Seitz]

Properties of Volume Intersection

Pros

- Easy to implement, fast
- Accelerated via octrees [Szeliski 1993]

Cons

- No concavities
- Reconstruction is not photo-consistent
- Requires identification of silhouettes

More General Cases (Color images, general cameras):

- Voxel Coloring [Seitz and Dyer]
- Space Carving [Kutulakos and Seitz]

Applications of Direct Methods: Real-Time Mapping and Tracking

Using Passive and RGB-D sensors

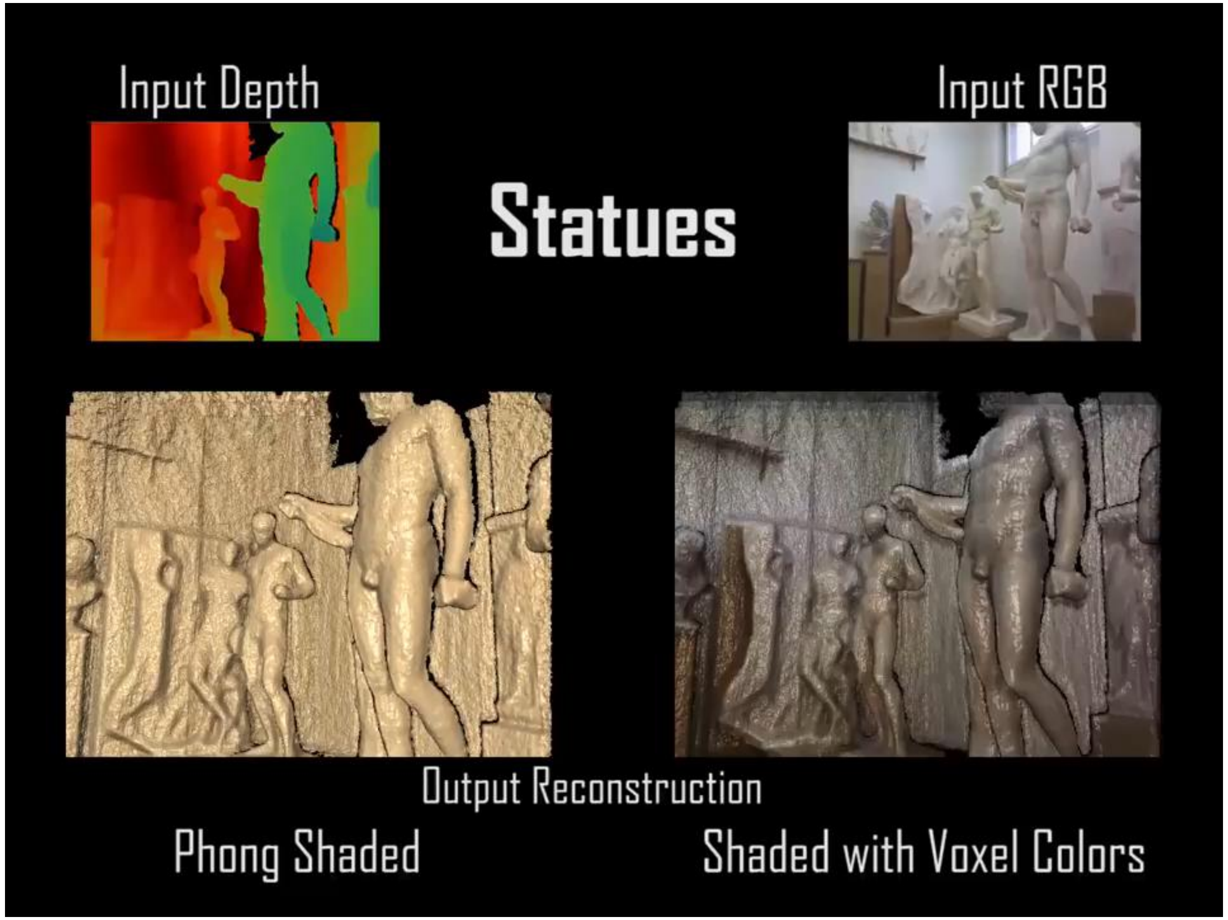
KinectFusion: Dense Surface Tracking and Mapping in Real-Time

- Uses an RGB-D Sensor
- First Dense SLAM System
- Interleaves:
  1. TSDF Fusion (Map)
  2. Projective ICP (Track)
- Efficient to implement on GPU Compute Architecture
- Memory for Scene is O(N^3)

Newcombe, Izadi et al

Real-Time 3D Reconstruction at Scale using Voxel Hashing

- Extends KinectFusion methods to work over very large volumes
- Very Efficient <O(N^3) Memory!



[Niesner, Zollhofer, Izadi, Stamminger]

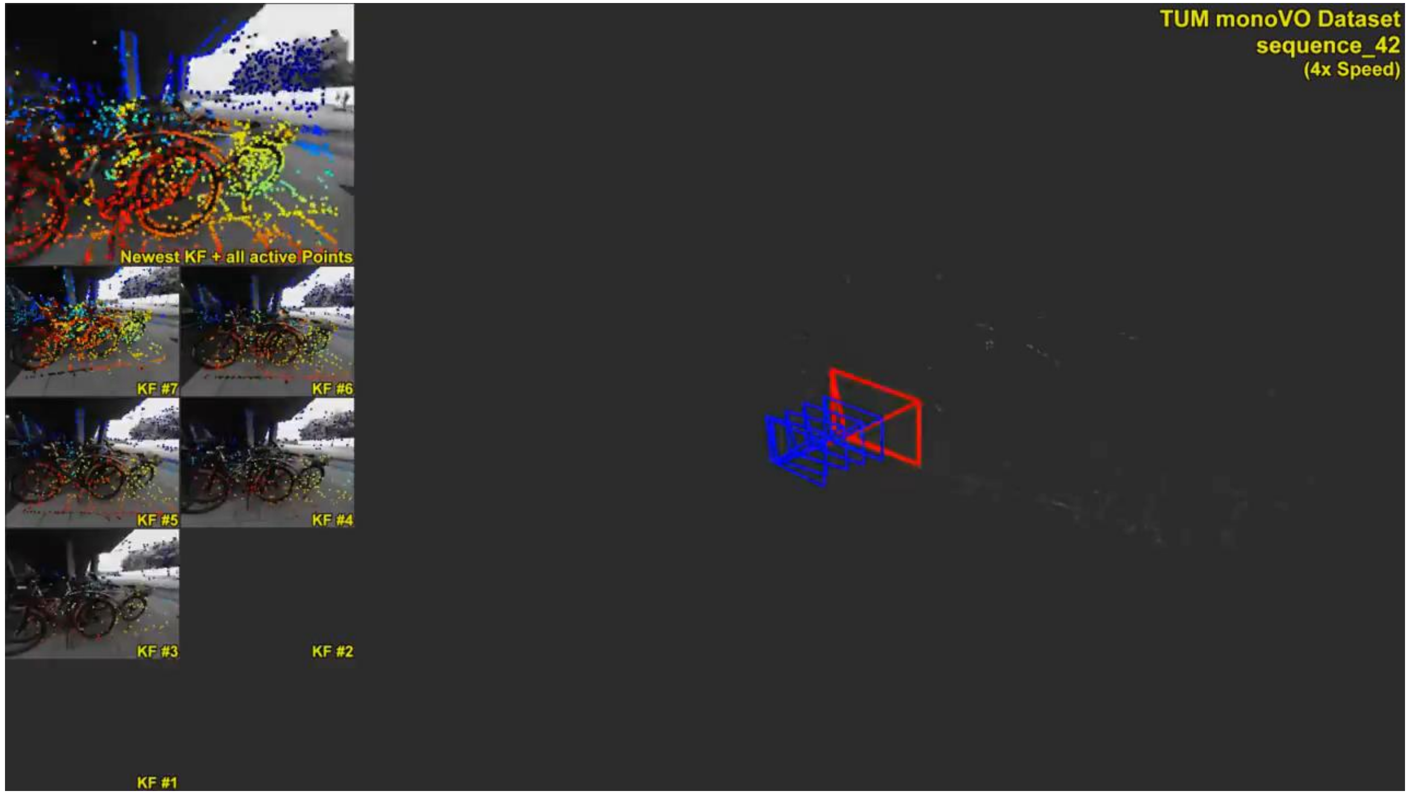
ElasticFusion: Dense SLAM Without A Pose Graph

- Uses Surfel scene representation
- RGB-D dense tracking
- Enables Loop Closure with a Deformation Graph

[Whelan et al]

DSO: Direct Sparse Odometry

- Passive Mono Camera
- Full Direct Formulation:
  - Jointly optimizes scene geometry & Camera Motion
- Generative model for accounting for Image Brightness changes
- Works across many more indoor/Outdoor Scenes

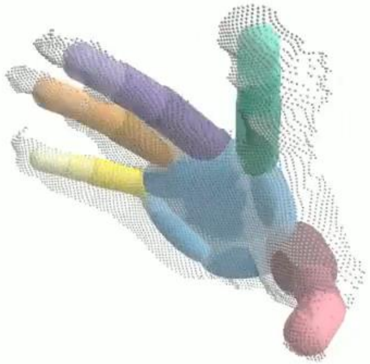


[Engel, Koltun, Cremers]

DART: Dense **Articulated** Real-Time Tracking

- Uses RGB-D Sensor
- Tracking only systems
- Tracks any Piece-wise rigid Articulated Object Model
- *Applications in Hand, Human, Robot and Object Tracking*

Experimental Results



Subject	1	2	3	4	5	6
FORTH	35.4	19.8	27.3	26.3	16.6	46.2
ICP-PSO	9.3	24.1	14.4	13.4	11.0	20.0
DART asym.	32.0	34.4	47.4	21.3	19.1	35.6
DART symm.	14.1	12.0	24.7	14.4	12.6	26.8

[Schmidt et al]

DynamicFusion: Reconstruction and Tracking of Non-rigid Scenes in Real-time

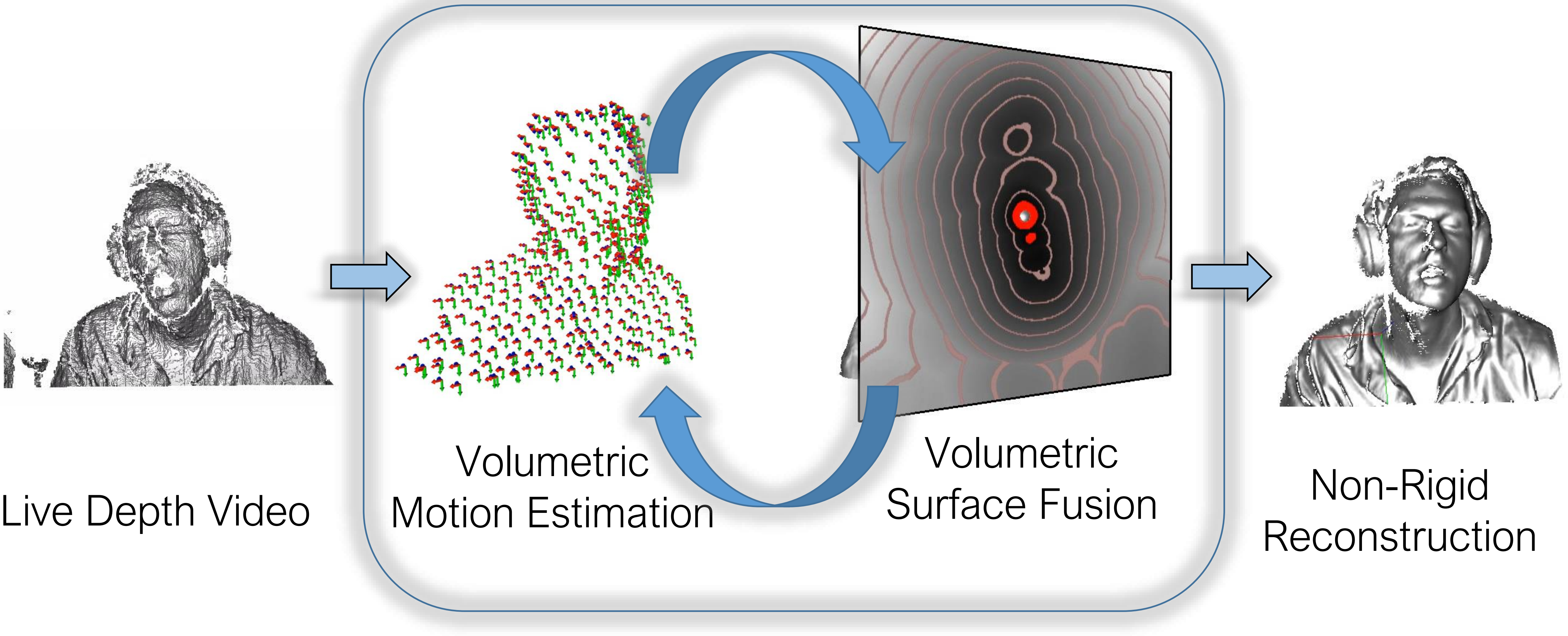
- Uses RGB-D Sensor
- Generalizes KinectFusion
- Non-rigid Scene Motion Representation

Live RGB-D Image

Real-time Non-Rigid Reconstruction

[Newcombe, Fox, Seitz]

DynamicFusion: Reconstruction and Tracking of Non-rigid Scenes in Real-time



[Newcombe, Fox, Seitz]