Recent Advances in 3D Computer Vision

SDF-Tracking

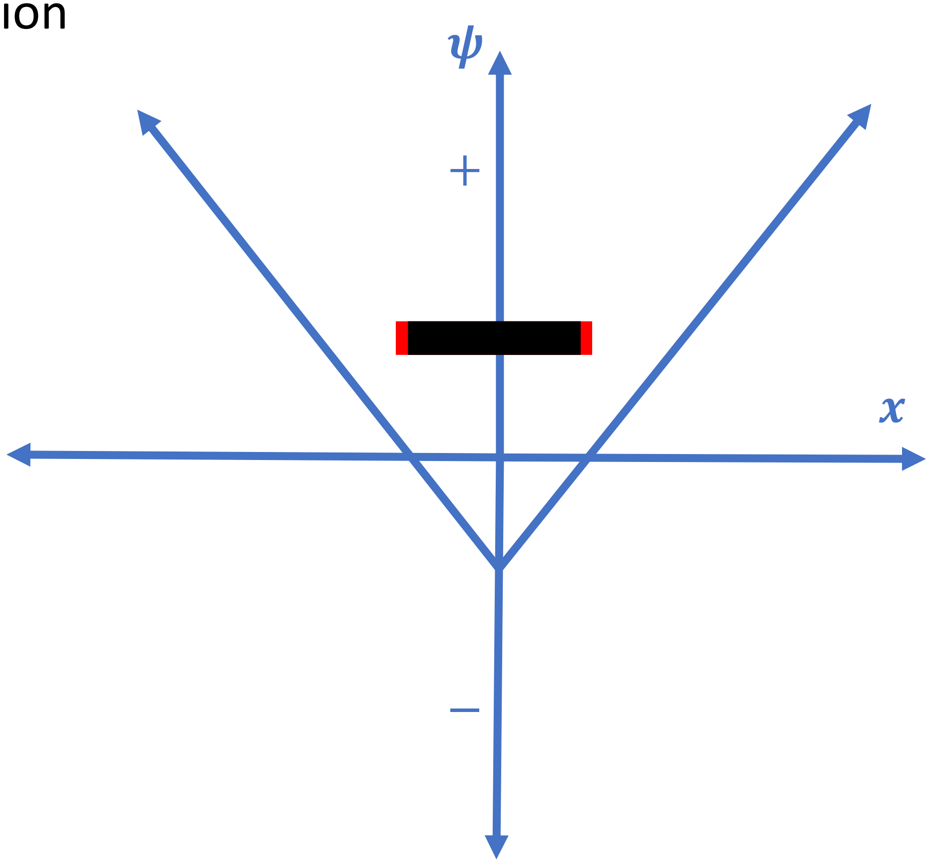
Leonardo Lerchenfeld

Real-Time Camera Tracking and 3D Reconstruction Using Signed Distance Functions

Erik Bylow, Jürgen Sturm, Christian Kerl, Fredrik Kahl and Daniel Cremers

https://www.youtube.com/watch?v=MzLdRFSrtuI

SDF-Tracking



Signed Distance Function (SDF)

Introduction and Problem Motivation

- **3D SLAM** is useful for
  - ➢ Robotics
  - ➢ Computer Vision
  - ➢ Architecture

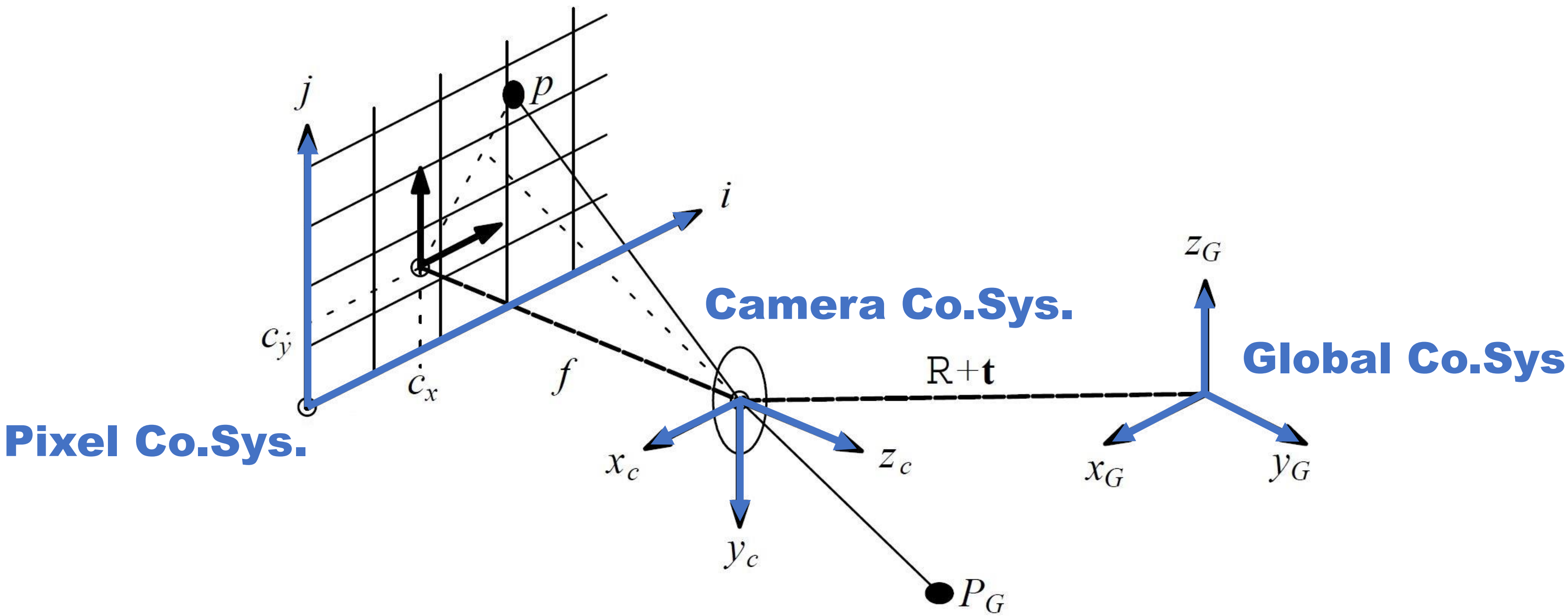
- **RGB-D sensors** output depth images
- **RGB-D SLAM** is accurate but no real-time
- **Kinect Fusion** in real-time but not as accurate

➔**SDF-Tracking**

Notation

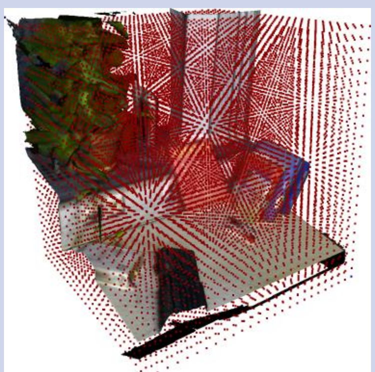
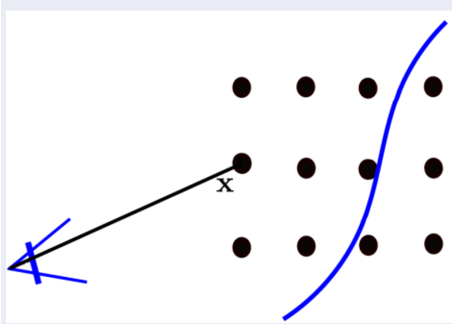
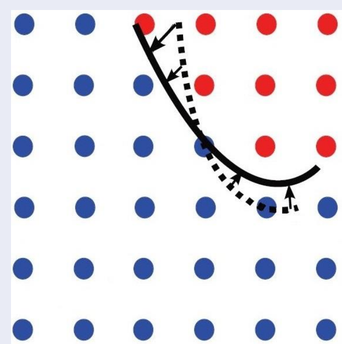
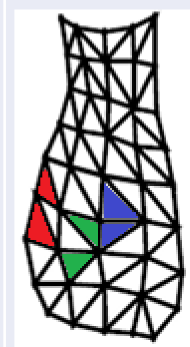
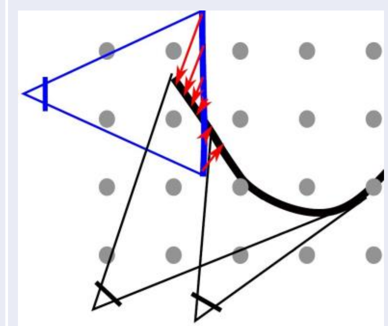
RGB-D Sensor	
Function for color images	$I_{RGB}\colon \mathbb{R}^2 \to \mathbb{R}^3$
Function for depth images	$I_d\colon \mathbb{R}^2 \to \mathbb{R}$
Camera and Global Coordinates	
3D point	$x \in \mathbb{R}^3$
Rotation of the camera	$R \in SO(3)$
And the translation	$\mathbf{t} \in \mathbb{R}^3$
from Camera to Global coordinates	$x_{ij}^{G} = Rx_{ij} + \mathbf{t}$
Camera Coordinates and the image plane coordinates	
focal length and the optical center	f_x, f_y, c_x, c_y

Coordinate Systems



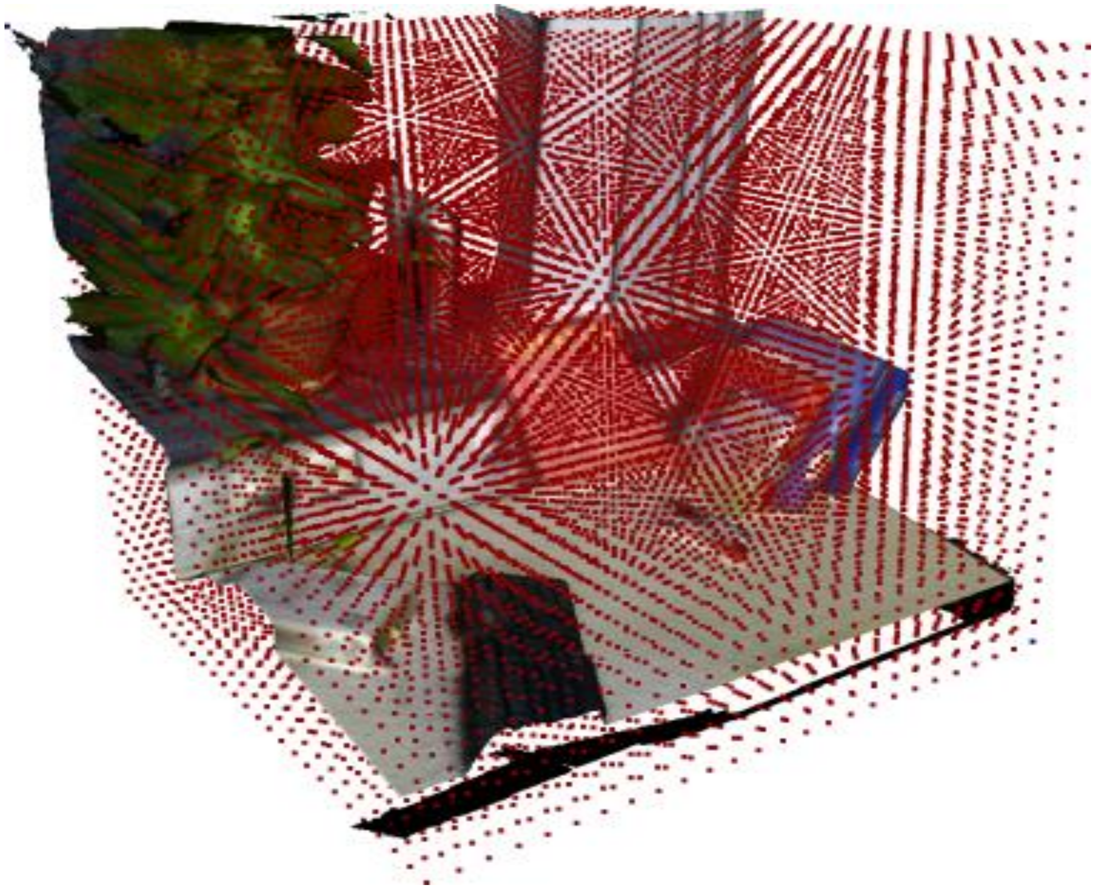
➔ Pixel coordinates	$\pi(\mathrm{x},\mathrm{y},\mathrm{z}) = \left(\frac{f_x x}{z} + c_x, \frac{f_y y}{z} + c_y\right)^T$
➔ Camera coordinates	$\rho(\mathrm{i},\mathrm{j},\mathrm{z}) = \left(\frac{(i - c_x)z}{f_x}, \frac{(j - c_y)z}{f_y}, z\right)^T$
➔ Global coordinates	$\mathrm{x}_{\mathrm{ij}}^{\mathrm{G}} = \mathrm{R}\mathrm{x}_{\mathrm{ij}} + \mathbf{t}$

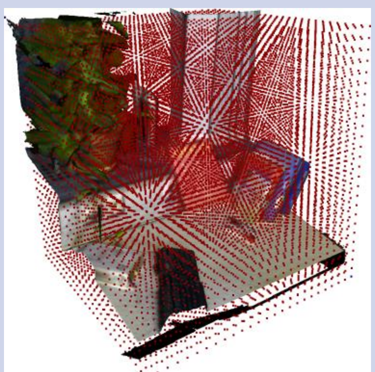
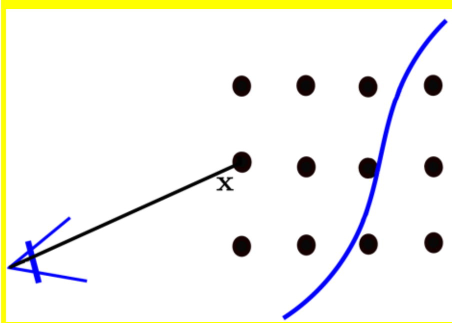
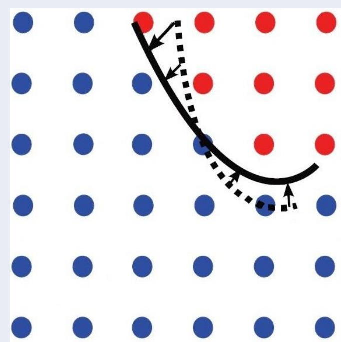
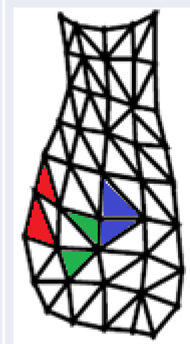
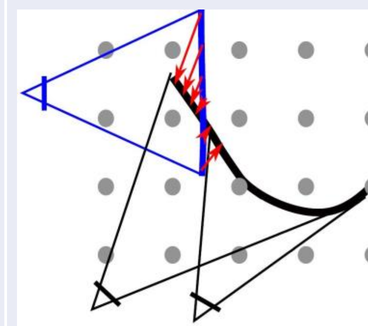
Approach

Representation of the SDF	Distance and Weighting Functions	Data Fusion and 3D Reconstruction	Color and Mesh	Camera Tracking
Voxel grids	Projective distance Weighting functions	Optimal SDF ψ	Surface Color	Rotation R Translation **t**
				

Representation of the SDF

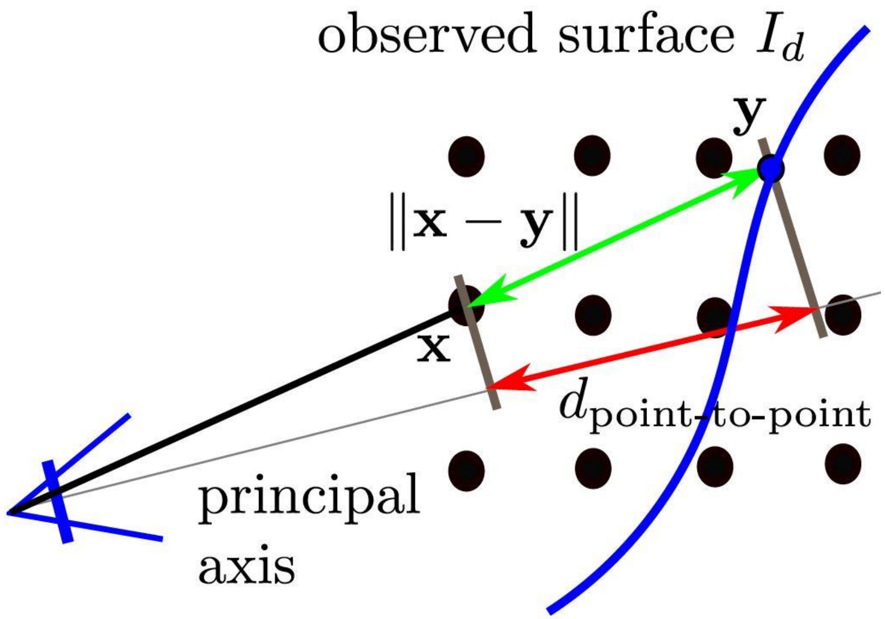
- Voxel grid of resolution *m*
- 6 channels
  - D: averaged **d**istance
  - W: sum of all **w**eights
  - R: **r**ed
  - G: **g**reen
  - B: **b**lue
  - W_c: **c**olor **w**eights
- $[0, \ldots, m-1]^3 \rightarrow \mathbb{R}$



Representation of the SDF	Distance and Weighting Functions	Data Fusion and 3D Reconstruction	Color and Mesh	Camera Tracking
Voxel grids	Projective distance Weighting functions	Optimal SDF ψ	Surface Color	Rotation R Translation $\mathbf{t}$
				

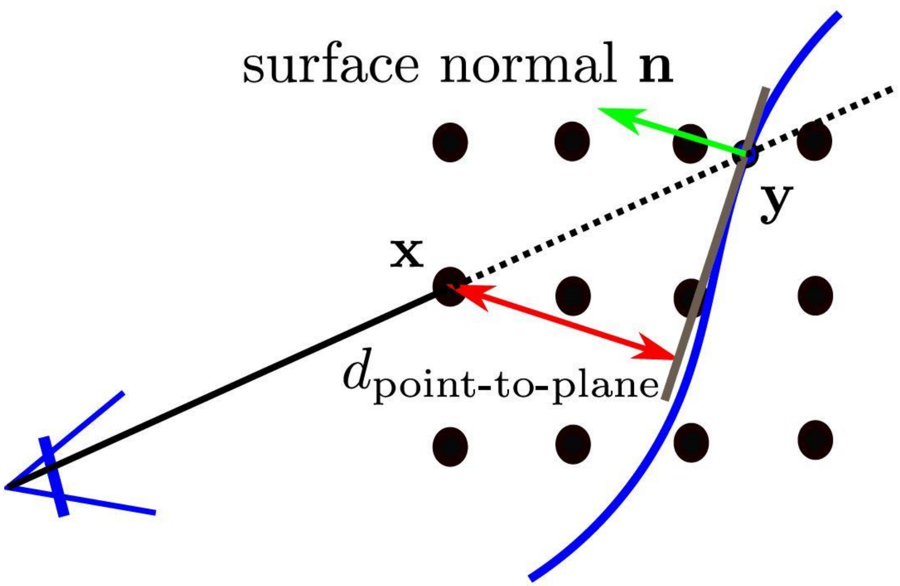
Projective Distances

Point-To-Point



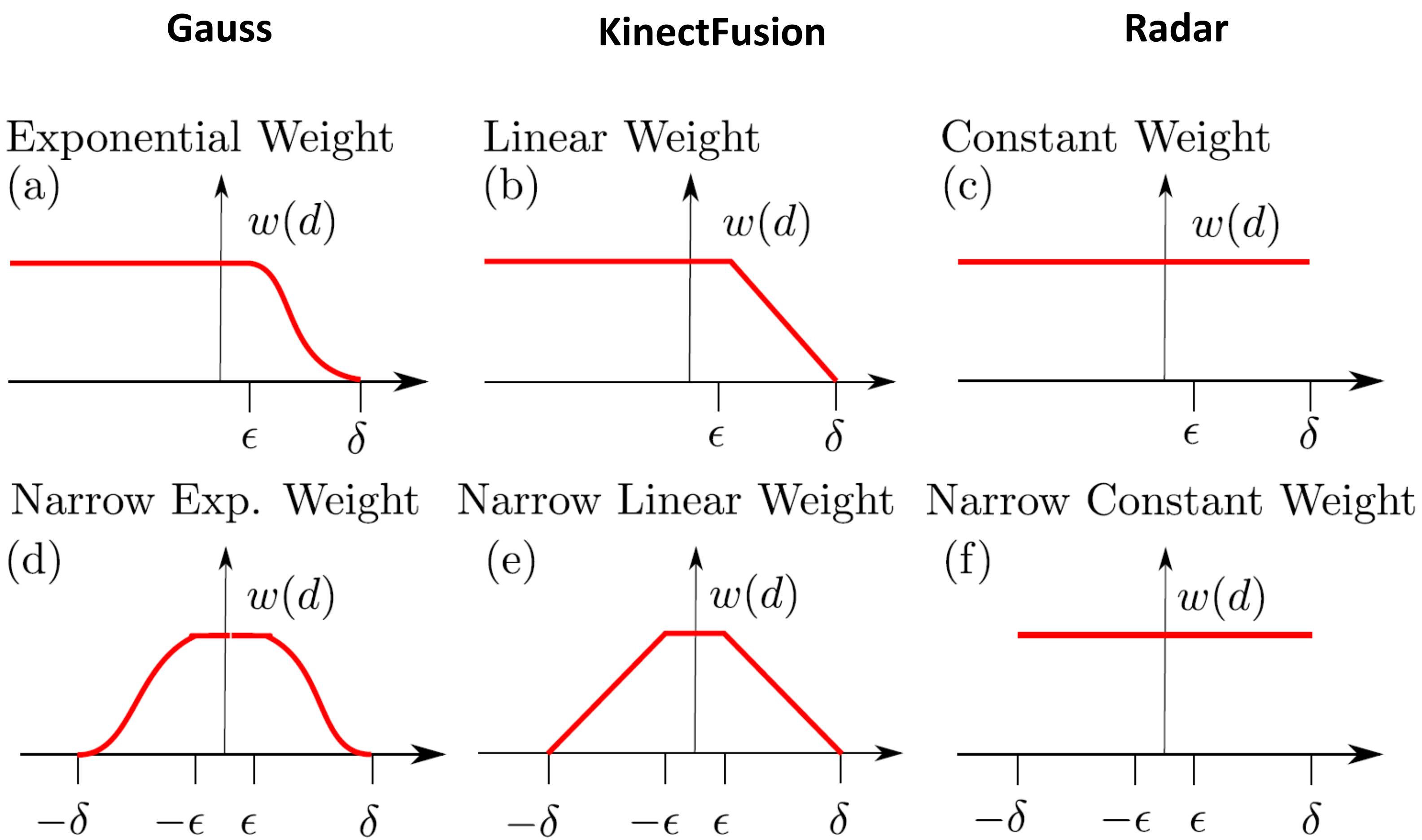
$$d_{point-to-point}(\boldsymbol{x}) = z - I_d(i,j)$$

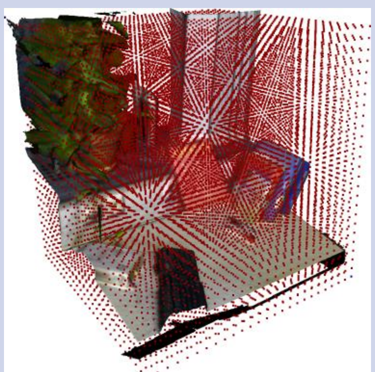
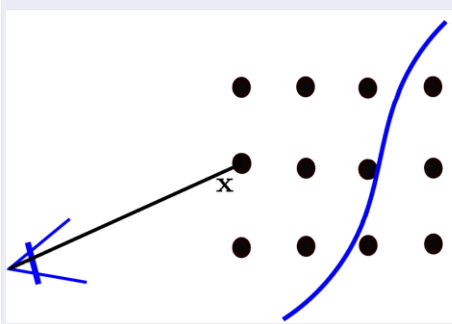
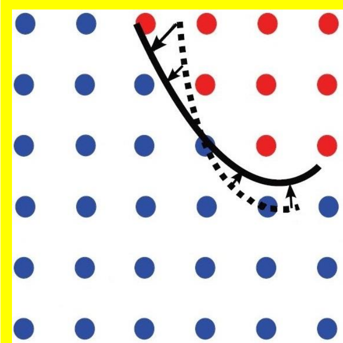
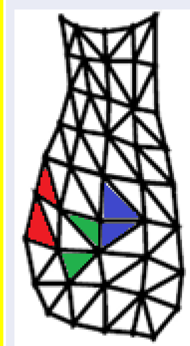
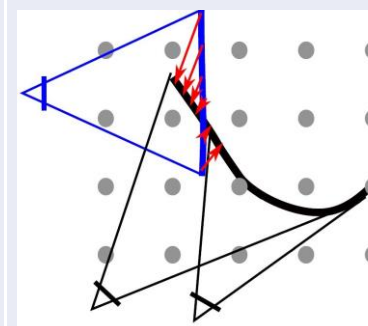
Point-To-Plane



$$d_{point-to-plane}(\boldsymbol{x}) = (\boldsymbol{y} - \boldsymbol{x})^T \boldsymbol{n}(i,j)$$

Weighting Functions



Representation of the SDF	Distance and Weighting Functions	Data Fusion and 3D Reconstruction	Color and Mesh	Camera Tracking
Voxel grids	Projective distance Weighting functions	Optimal SDF ψ	Surface Color	Rotation R Translation **t**
				

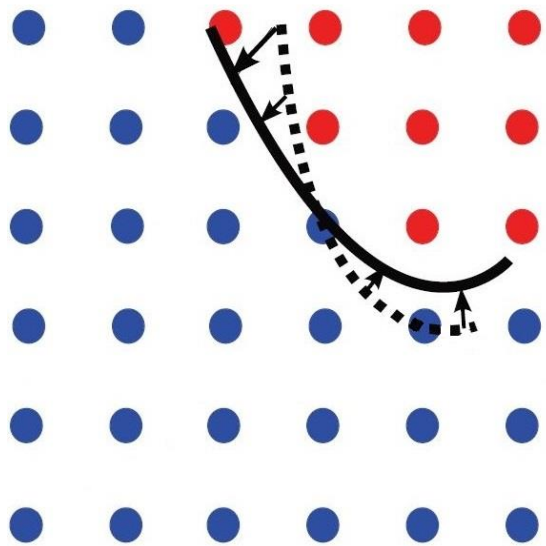
Data Fusion and 3D Reconstruction

SDF

$$\psi: \mathbb{R}^3 \rightarrow \mathbb{R}$$

Error Function

$$L(\psi) = \sum_{i=1}^{n} \frac{1}{2} w_i (\psi - d_i)^2$$



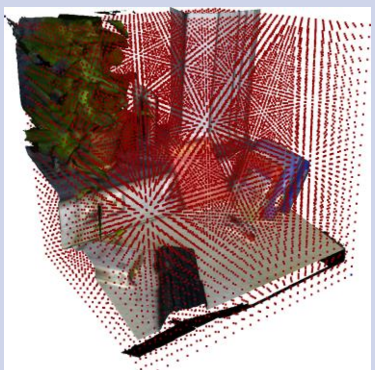
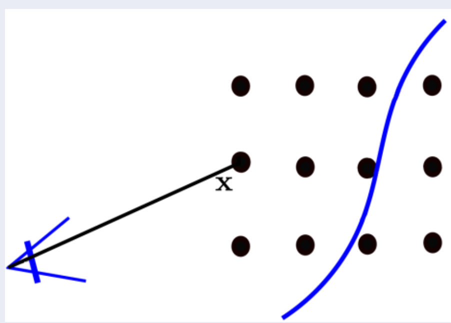
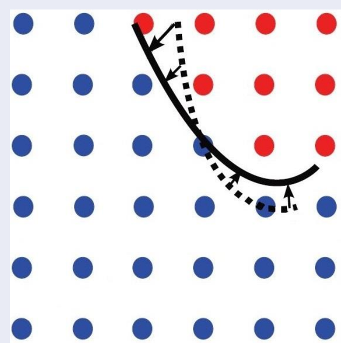
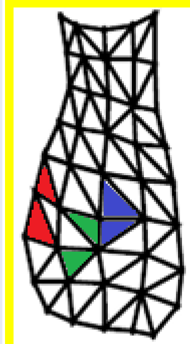
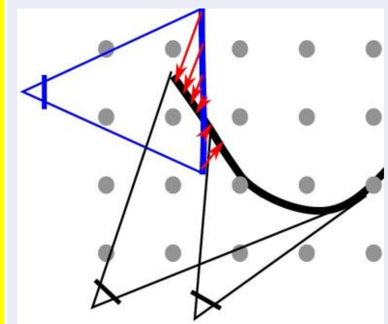
Optimal SDF

$$\psi = \frac{\sum_{i=1}^{n} w_i d_i}{\sum_{i=1}^{n} w_i}$$

Running Average

$$D \leftarrow \frac{WD + w_{n+1} d_{n+1}}{W + w_{n+1}}$$

$$W \leftarrow W + w_{n+1}$$

Representation of the SDF	Distance and Weighting Functions	Data Fusion and 3D Reconstruction	**Color and Mesh**	Camera Tracking
Voxel grids	Projective distance Weighting functions	Optimal SDF ψ	Surface Color	Rotation R Translation **t**
				

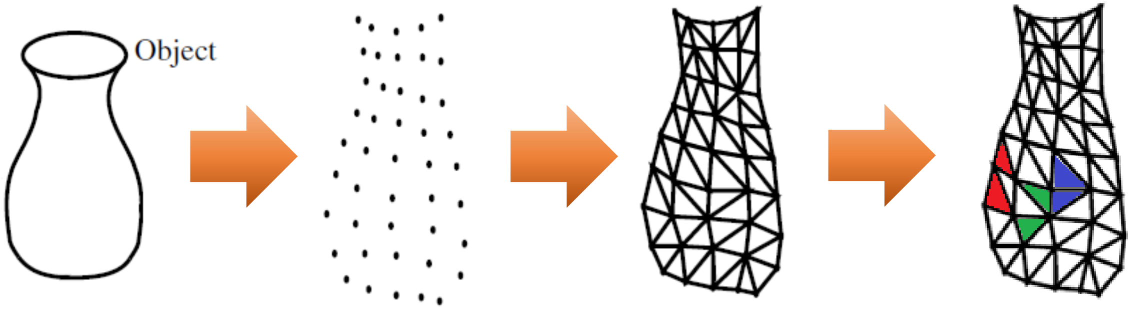
Color and Mesh

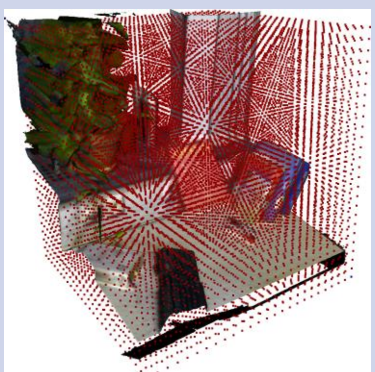
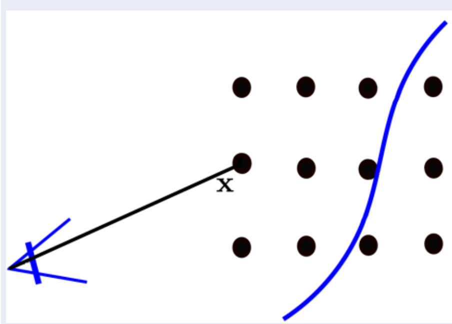
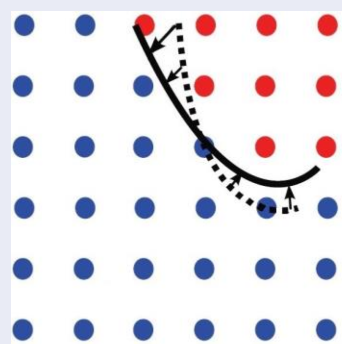
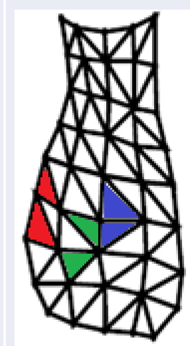
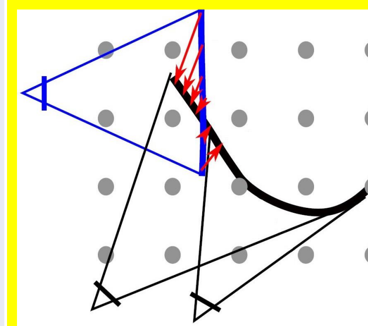
$$(r, g, b)^T = I_{RGB}(i, j)$$

$$G \leftarrow \frac{W_c G + w_c^{n+1} g}{W_c + w_c^{n-1}}$$

$$w_c^{n+1} = w_{n+1} cos\theta$$

Marching cubes algorithm



Representation of the SDF	Distance and Weighting Functions	Data Fusion and 3D Reconstruction	Color and Mesh	**Camera Tracking**
Voxel grids	Projective distance Weighting functions	Optimal SDF ψ	Surface Color	Rotation R Translation $\mathbf{t}$
				

Camera Tracking

$$(x, y, z)^T = \rho(i, j, I_d(i, j))$$ Camera coordinates

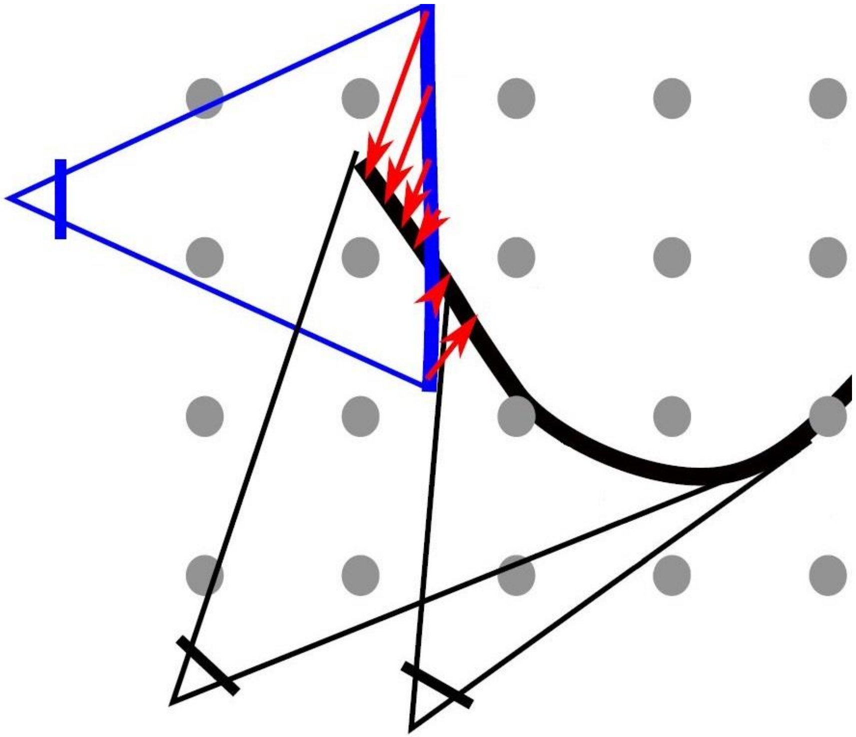
$$x_{ij}^G = Rx_{ij} + t$$ Global coordinates

$$\psi\colon \mathbb{R}^3 \to \mathbb{R}$$

$$\psi_{ij}(\xi) = \psi\left(\mathbf{R}x_{ij} + \mathbf{t}\right)$$

Twist Coordinates:

$$\xi = (\omega_1, \omega_2, \omega_3, v_1, v_2, v_3)$$



Gauss-Newton Method

$$E(\boldsymbol{\xi}) = E(\boldsymbol{R}, \boldsymbol{t}) = \sum_{i,j} \psi_{ij}(\boldsymbol{\xi})^2$$

$$\psi(\boldsymbol{\xi}) \approx \underbrace{\psi\left(\boldsymbol{\xi}^{\boldsymbol{k}}\right)}_{1\text{x}1} + \underbrace{\nabla\psi\left(\boldsymbol{\xi}^{\boldsymbol{k}}\right)^T}_{1\text{x}6} \underbrace{\left(\boldsymbol{\xi} - \boldsymbol{\xi}^{\boldsymbol{k}}\right)}_{6\text{x}1}$$

$$\sum_{i,j} \psi_{ij}(\boldsymbol{\xi})^2 \approx \sum_{i,j} \psi_{ij}\left(\boldsymbol{\xi}^{\boldsymbol{k}}\right)^2 + 2\psi_{ij}\left(\boldsymbol{\xi}^{\boldsymbol{k}}\right) \nabla\psi_{ij}\left(\boldsymbol{\xi}^{\boldsymbol{k}}\right)^T \left(\boldsymbol{\xi} - \boldsymbol{\xi}^{\boldsymbol{k}}\right) + \left(\nabla\psi_{ij}\left(\boldsymbol{\xi}^{\boldsymbol{k}}\right)^T \left(\boldsymbol{\xi} - \boldsymbol{\xi}^{\boldsymbol{k}}\right)\right)^2$$

$$\frac{d}{d\boldsymbol{\xi}} E_{approx}(\boldsymbol{\xi}) = \sum_{i,j} 2\psi_{ij}\left(\boldsymbol{\xi}^{\boldsymbol{k}}\right) \nabla\psi_{ij}\left(\boldsymbol{\xi}^{\boldsymbol{k}}\right) + 2\left(\nabla\psi_{ij}\left(\boldsymbol{\xi}^{\boldsymbol{k}}\right) \nabla\psi_{ij}\left(\boldsymbol{\xi}^{\boldsymbol{k}}\right)^T \left(\boldsymbol{\xi} - \boldsymbol{\xi}^{\boldsymbol{k}}\right)\right) = 0$$

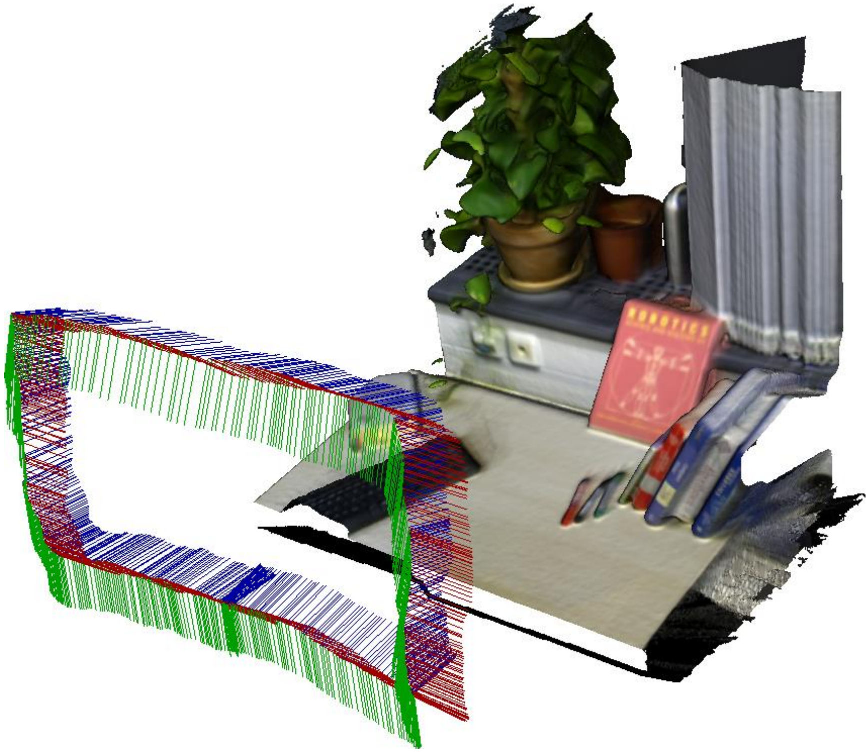
$$\boldsymbol{b} + \boldsymbol{A}\left(\boldsymbol{\xi} - \boldsymbol{\xi}^{\boldsymbol{k}}\right) = \boldsymbol{0}$$

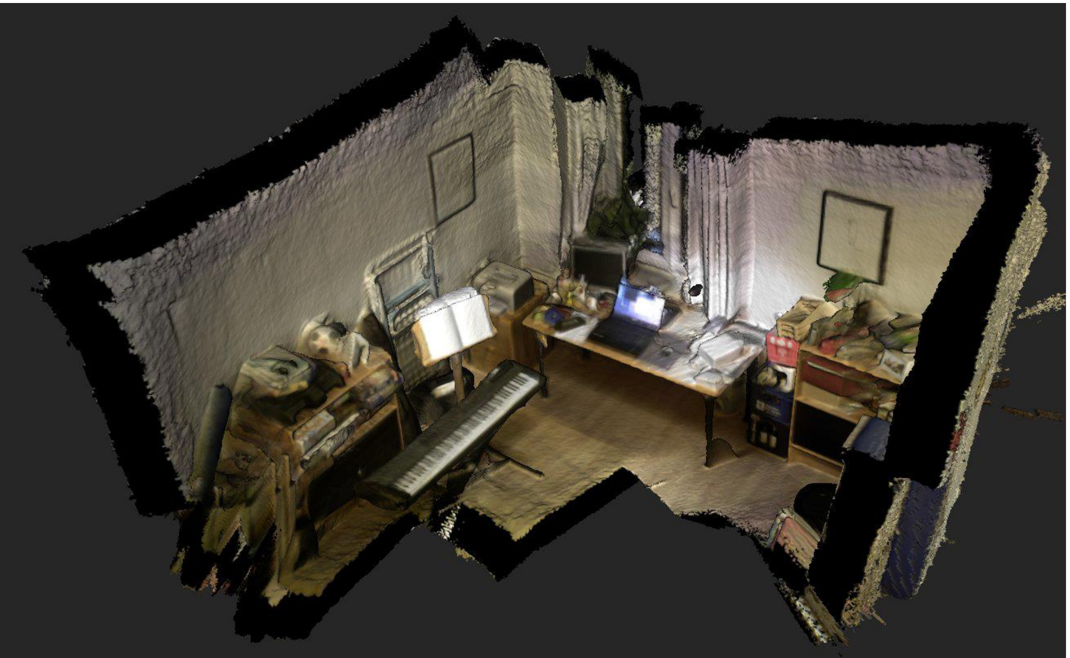
$$\boldsymbol{\xi}^{\boldsymbol{k+1}} = \boldsymbol{\xi}^{\boldsymbol{k}} - \boldsymbol{A}^{-1}\boldsymbol{b}$$

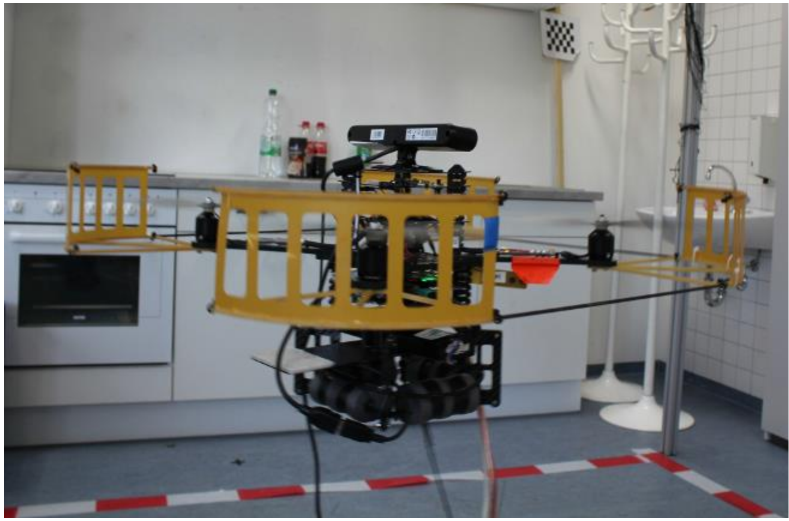
Experimental Results

Experimental Results

- Almost drift-free for small scenes
- Architects can use it for planning
- Navigation of robots







Parameter Study

- Truncation δ = 0.3m
- Weighting function
- Absolute trajectory error in m

Dataset	F1 Teddy		F1 Desk	
	RMSE	Max	RMSE	Max
Exp. Weight	0.088 m	**0.213 m**	**0.038 m**	0.088 m
Linear Weight	**0.083 m**	0.285 m	**0.038 m**	0.089 m
Constant Weight	0.093 m	0.242 m	0.040 m	0.089 m
Narrow Exp.	0.170 m	0.414 m	**0.038 m**	**0.083 m**
Narrow Linear	0.382 m	0.688 m	0.044 m	0.085 m
Narrow Constant	0.379 m	0.694 m	0.044 m	0.209 m

Method	Res.	Teddy	Desk	Desk2	Household	Floor	360	Room	Plant
KinFu	256	0.156	0.057	0.42	0.064	Failed	0.913	Failed	0.598
*Plane	256	**0.072**	0.087	0.078	0.053	0.811	0.533	0.163	0.047
*Point	256	0.086	0.038	**0.061**	**0.039**	0.641	0.420	0.121	0.047
KinFu	512	0.337	0.068	0.635	0.061	Failed	0.591	0.304	0.281
*Plane	512	0.101	0.059	0.623	0.053	0.640	0.206	0.105	**0.041**
*Point	512	0.08	**0.035**	0.062	0.04	**0.567**	**0.119**	**0.078**	0.043
RGB-D SLAM		0.111	0.026	0.043	0.059	0.035	0.071	0.101	0.061

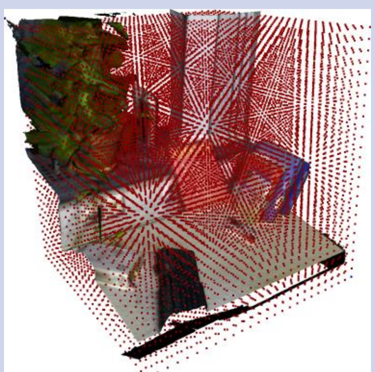
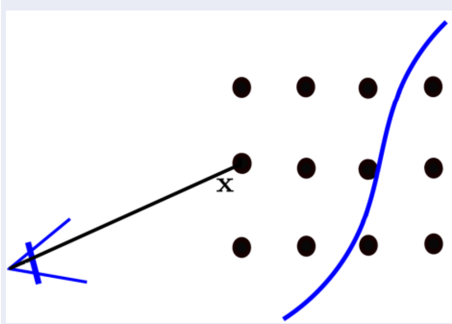
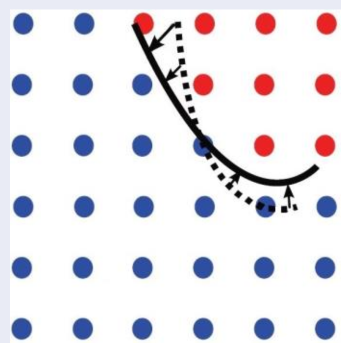
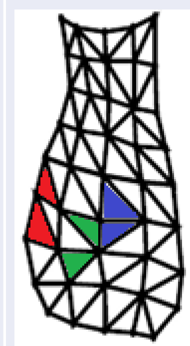
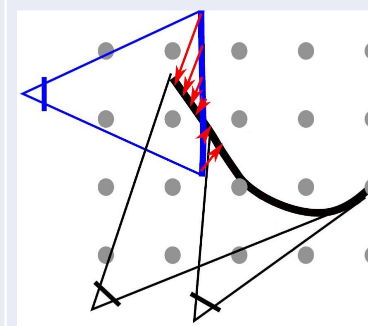
- Point-To-Point metric better
- Similar performance compared to RGB-D SLAM
- Clearly outperforms KinectFusion

Runtime and Memory Consumption

Res.	RAM on the GPU		
	SDF	Color grid	
512	1 GB	2 GB	
256	128 MB	256 MB	
	Runtime		Comparison
	Pose optimization **O(i*j)**	Data fusion **O(m^3)**	
512	31.1 ms	21.6 ms	
256	19.4 ms	3.7 ms	= 23 ms
KinFu (256)			20 ms
RGB-D SLAM			100 – 250 ms

- For m=256, in real-time on 30 fps
- Speed as KinFu
- Faster than RGB-D SLAM

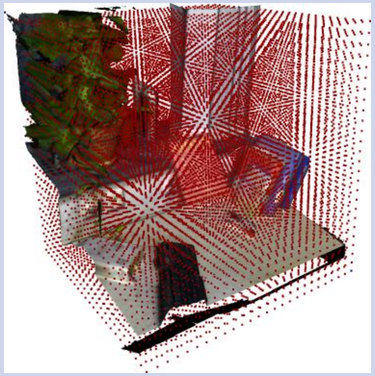
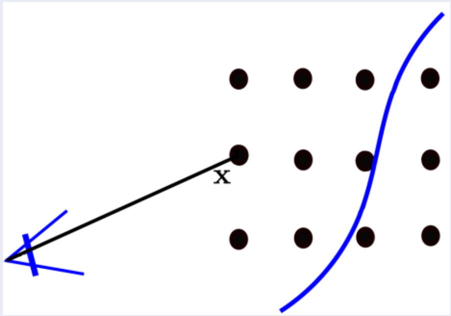
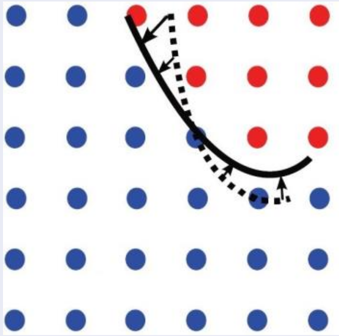
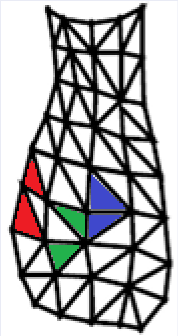
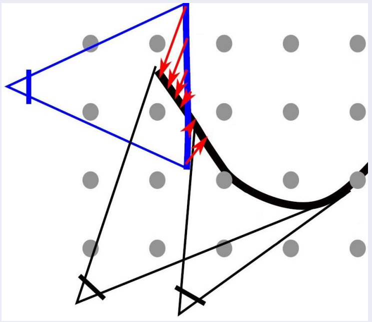
Summary

Representation of the SDF	Distance and Weighting Functions	Data Fusion and 3D Reconstruction	Color and Mesh	Camera Tracking
Voxel grids	Projective distance Weighting functions	Optimal SDF ψ	Surface Color	Rotation R Translation $\mathbf{t}$
				

Discussion

Advantages +	Drawbacks -
RGB-D SLAM: feature extraction & bundle adjustment	
Very accurate	No real-time
Kinect Fusion: virtual camera & ICP	
Real time	Not as accurate
	Memory consumption
SDF-Tracking: direct estimation on SDF	
Real-time	Memory consumption
Very accurate	RGB values are not used for tracking

Questions?

Representation of the SDF	Distance and Weighting Functions	Data Fusion and 3D Reconstruction	Color and Mesh	Camera Tracking
Voxel grids	Projective distance Weighting functions	Optimal SDF ψ	Surface Color	Rotation R Translation **t**
				

References

- E. Bylow, J. Sturm, C. Kerl, F. Kahl, D. Cremers. **Real-Time Camera Tracking and 3D Reconstruction Using Signed Distance Function**. In Robotics: Science and Systems, 2013
- E. Bylow, C. Olsson, F. Kahl (2014). **Robust camera tracking by combining color and depth measurements**. In 2014 22nd international conference on pattern recognition (ICPR).
- B. Curless and M. Levoy. **A volumetric method for building complex models from range images**. In SIGGRAPH, 1996.
- C. Steger, M. Ulrich, C. Wiedemann. **Machine Vision Algorithms and Applications**. WILEY-VCH Verlag, 2008. Pages 182-183
- S. Osher and R. Fedkiw. **Level Set Methods and Dynamic Implicit Surfaces**. Springer, 2003. Page 20