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

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

Robotics: Science and Systems Conference (RSS), 2013

Yusuf Ziya Güleray

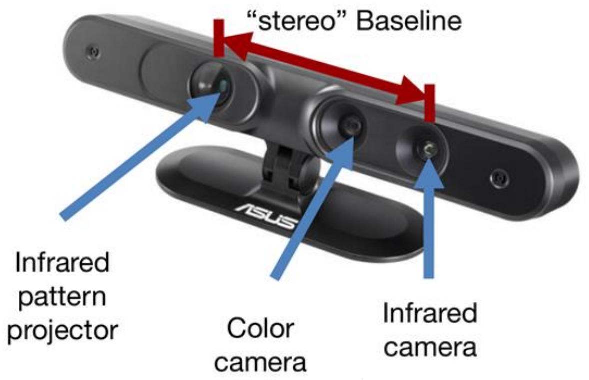
Munich, 5. October 2020

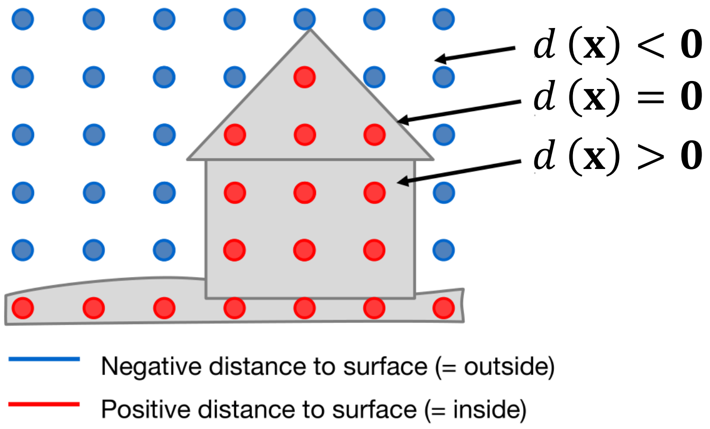
Goal and Constraints

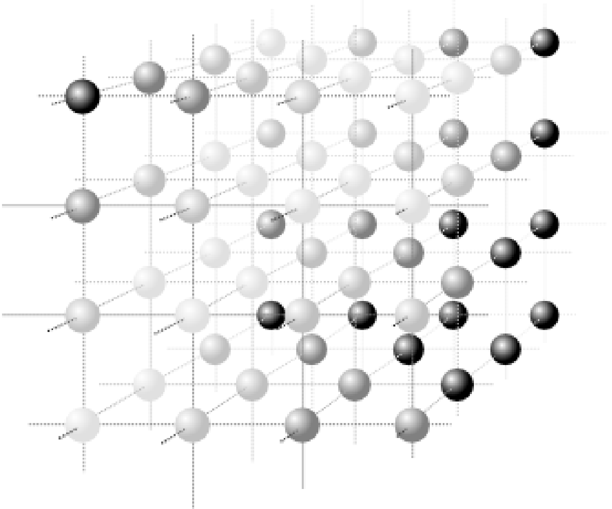
- **Real-time camera tracking** and **3D reconstruction**
- **Static** indoor environments using an **RGB-D** sensor
- **Real-Time** capable on a laptop with a Quadro GPU
- **Absolute metric** information and minimal **drift**
- Augmented reality applications: computer games, home decoration, and refurbishment measures

Fundamentals

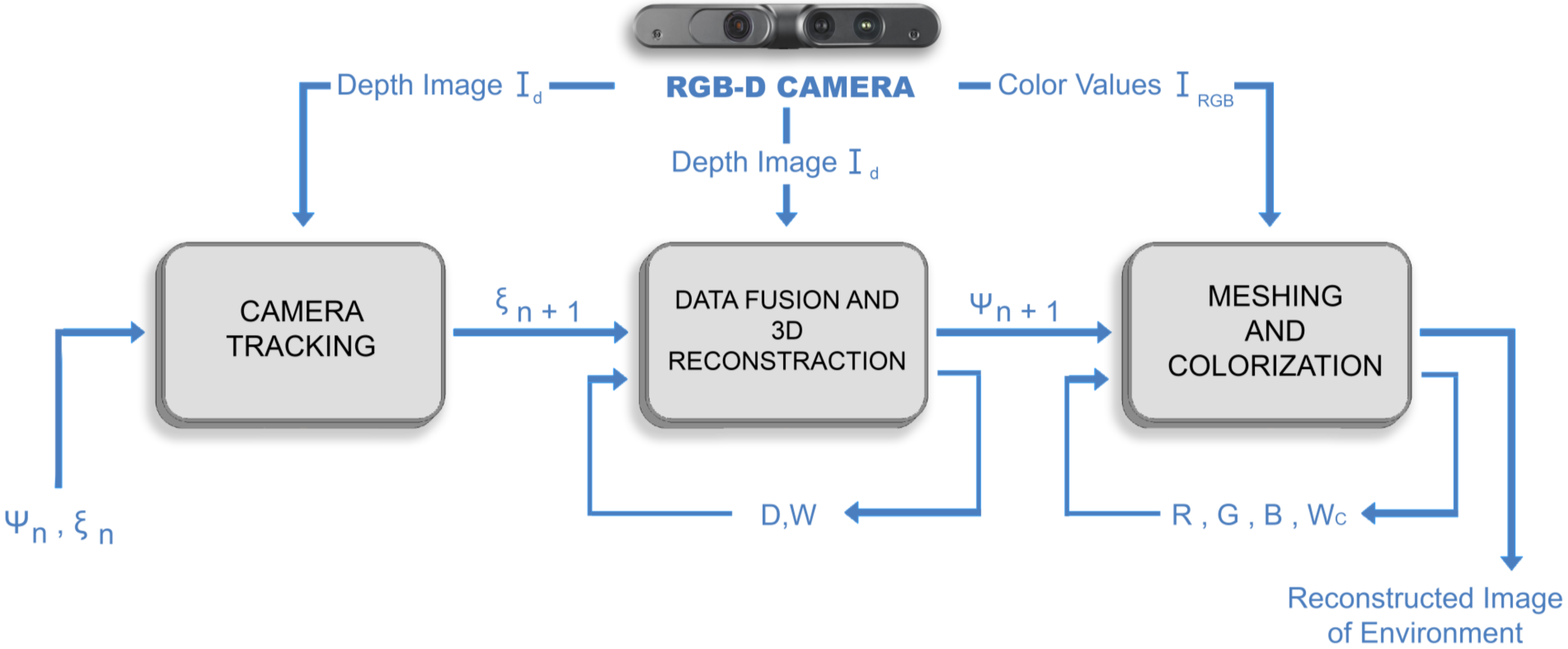
- **RGB-D Camera :** measures depth of every pixel (Asus Xtion Pro Live)
- **Voxel Grid :** Volumetric Pixel, represents a value on a regular grid in three-dimensional space
- **Signed distance function (SDF) :** Represents the distance to surface in a voxel grid

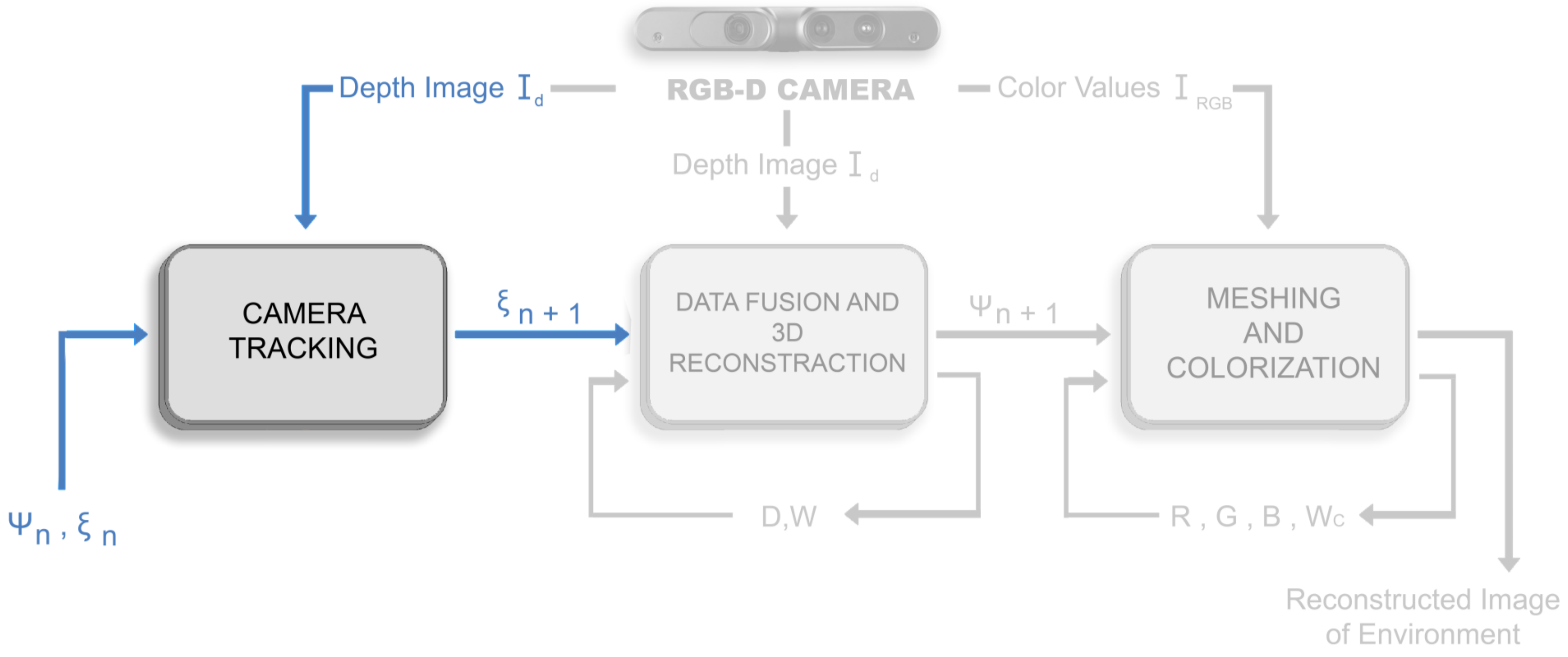






Approach


Depth Image I_d
RGB-D CAMERA
Color Values I_{RGB}
Depth Image I_d
CAMERA TRACKING
ξ_{n+1}
DATA FUSION AND 3D RECONSTRACTION
Ψ_{n+1}
MESHING AND COLORIZATION
Ψ_n, ξ_n
D,W
R , G , B , Wc
Reconstructed Image of Environment


RGB-D CAMERA
Depth Image I_d
Color Values I_{RGB}
Depth Image I_d
CAMERA TRACKING
ξ_{n+1}
DATA FUSION AND 3D RECONSTRACTION
Ψ_{n+1}
MESHING AND COLORIZATION
Ψ_n , ξ_n
D,W
R , G , B , Wc
Reconstructed Image of Environment

Camera Tracking

Finding the pose that fits the depth image to the SDF best

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

$$\mathbf{x}_{ij} = I_d(i, j)$$

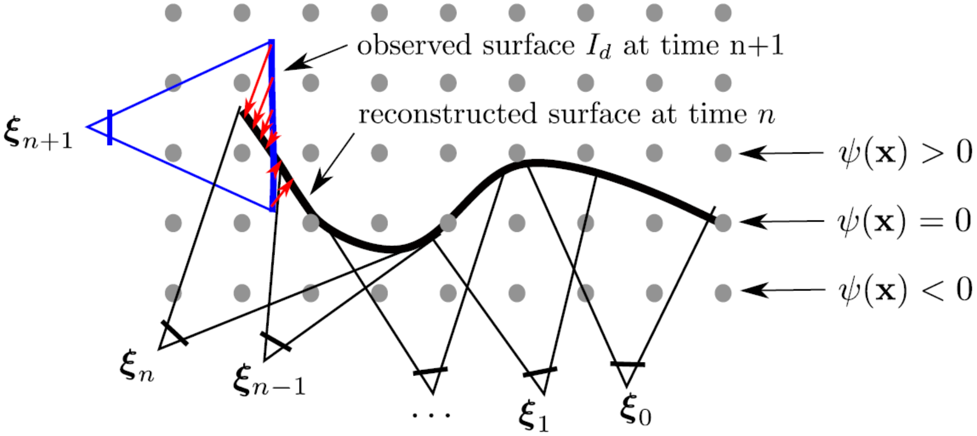
$$\mathbf{x}_{ij}^G = R\mathbf{x}_{ij} + \mathbf{t}$$

$$\{\mathbf{x} \mid \psi(\mathbf{x}) = 0\}$$

$$p(I_d \mid R, \mathbf{t}) \propto \prod_{i,j} \exp\left(-\psi\left(R\mathbf{x}_{ij} + \mathbf{t}\right)^2\right)$$

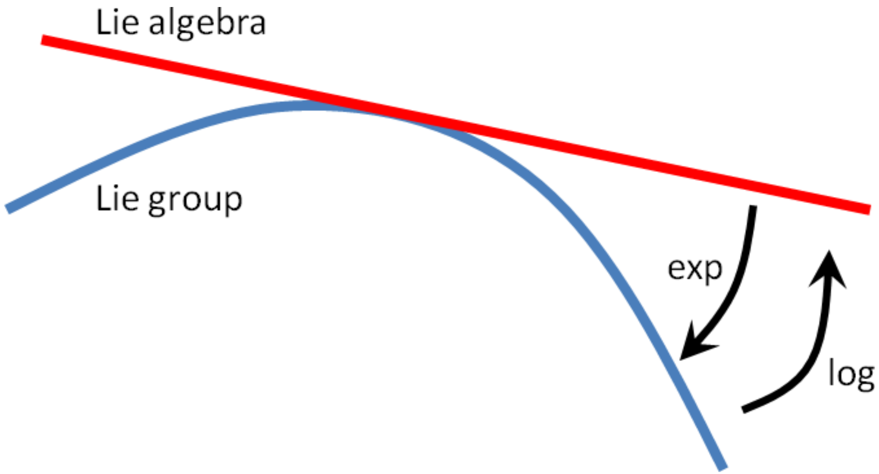
$$(R^*, \mathbf{t}^*) = \arg\max_{R,\mathbf{t}} p(I_d \mid R, \mathbf{t})$$

$$E(R, \mathbf{t}) = \sum_{i,j} \psi\left(R\mathbf{x}_{ij} + \mathbf{t}\right)^2$$



Camera Tracking Cont.

- Lie Group $\mathrm{so}(3)$ is a minimal representation
- Optimization using Lie Group becomes **unconstrained**



Lie Algebra $SO(3)$

$$R \in GL(3)$$
$$R^T R = I$$
$$\det(R) = 1$$

Lie Group $so(3)$

$$\{\, \hat{w} \mid w \in R^3 \}$$

Rigid body motion parameters in $\mathrm{so}(3)$: $\boldsymbol{\xi} = (\omega_1, \omega_2, \omega_3, v_1, v_2, v_3)$

$$E(\xi)) = \sum_{i,j} \psi\left(R\mathbf{x}_{ij} + \mathbf{t}\right)^2 = \sum_{i,j} \psi_{ij}(\xi)^2$$

$$\psi_{ij}(\xi) = \psi\left(Rx_{ij} + t\right)$$

Camera Tracking Cont.

$$\psi(\xi) \approx \psi(\xi^{(k)}) + \nabla\psi(\xi^{(k)})^{\top}(\xi - \xi^{(k)})$$

$$E_{\text{approx}}(\xi) = \sum_{i,j} \left(\psi_{ij}(\xi^{(k)}) + \nabla\psi_{ij}(\xi^{(k)})^{\top}(\xi - \xi^{(k)})\right)^2$$

$$\frac{\mathrm{d}}{\mathrm{d}\xi} E_{\text{approx}}(\xi) = 0$$

$$\sum_{i,j} \psi_{ij}(\xi^{(k)})\nabla\psi_{ij}(\xi^{(k)}) + \nabla\psi_{ij}(\xi^{(k)})\nabla\psi_{ij}(\xi^{(k)})^{\top}(\xi - \xi^{(k)}) = 0$$

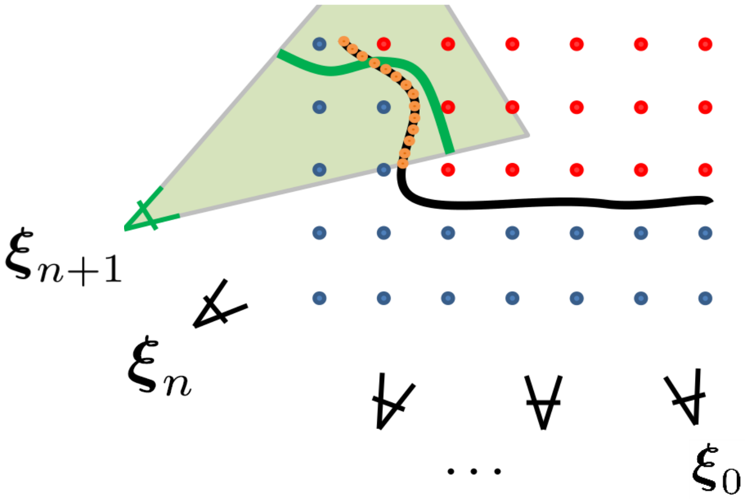
$$A \coloneqq \sum_{i,j} \nabla\psi_{ij}(\xi^{(k)})\nabla\psi_{ij}(\xi^{(k)})^{\top} \in R^{6\times 6}, \qquad b \coloneqq \sum_{i,j} \psi_{ij}(\xi^{(k)})\nabla\psi_{ij}(\xi^{(k)}) \in R^{6\times 1}$$

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

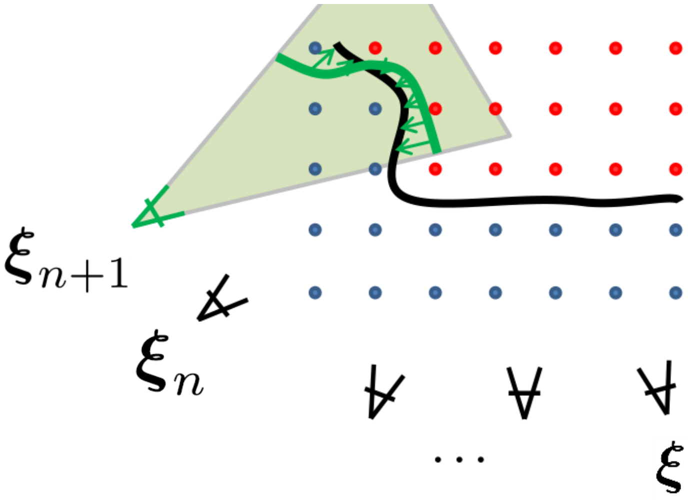
$$\xi^{(k+1)} = \xi^{(k)} - A^{-1}b$$

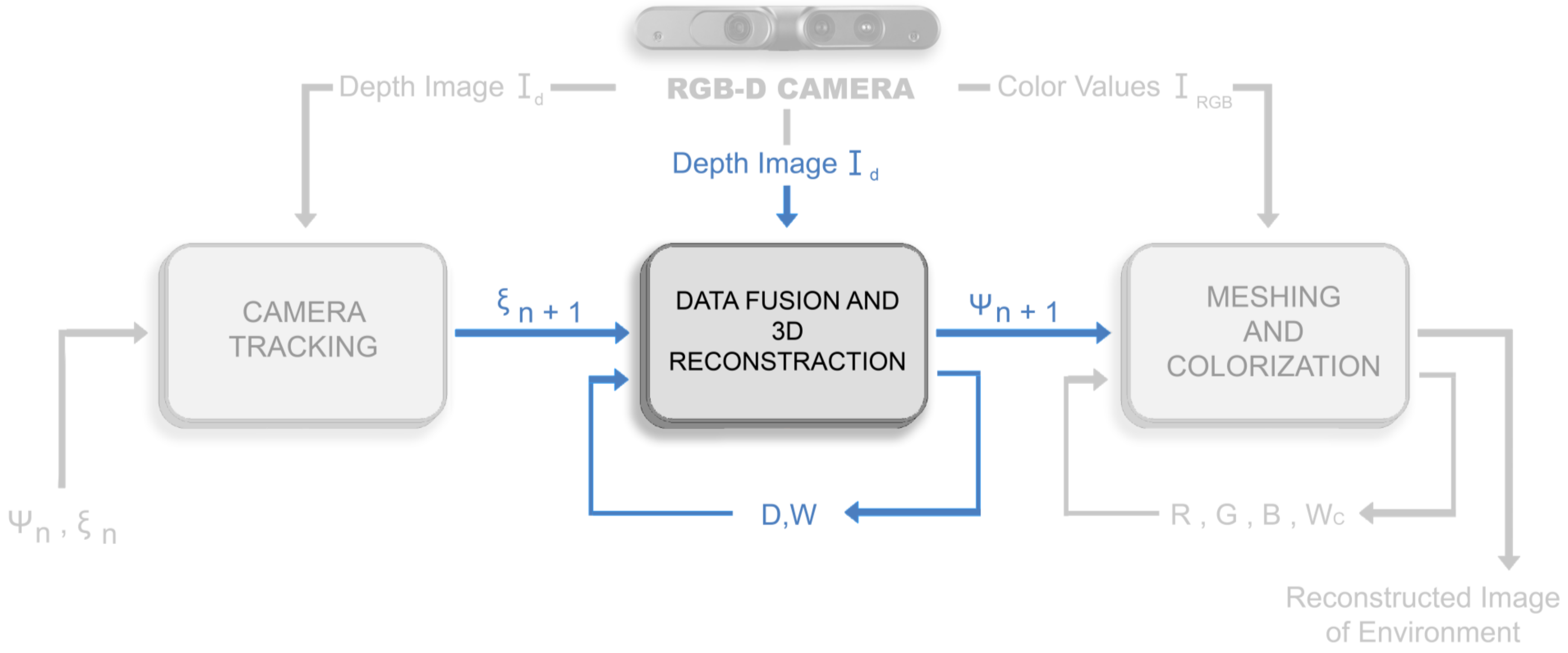
Camera Tracking Cont.

KinectFusion generates a **synthetic depth image** from SDF and aligns it using ICP



Here SDF is used **directly** during minimization





Depth Image I_d
RGB-D CAMERA
Color Values I_{RGB}
Depth Image I_d
CAMERA TRACKING
ξ_{n+1}
DATA FUSION AND 3D RECONSTRACTION
Ψ_{n+1}
MESHING AND COLORIZATION
D,W
R , G , B , Wc
Ψ_n , ξ_n
Reconstructed Image of Environment

SDF - Distance and Weighting Functions

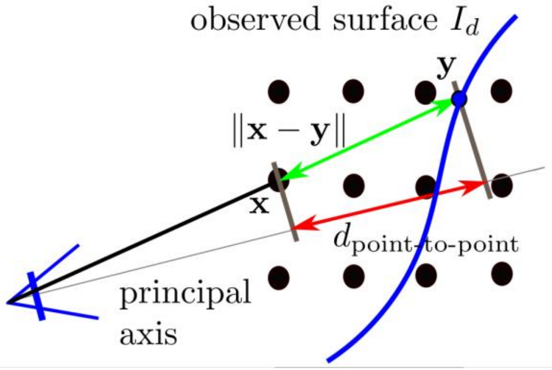
- Depth image : distance to the surface for **each pixel**
- SDF : distance to the surface from **each voxel**

How to integrate the new **depth images** to the **SDF**?

Computing the true distance of each voxel is not feasible, it has to be approximated:

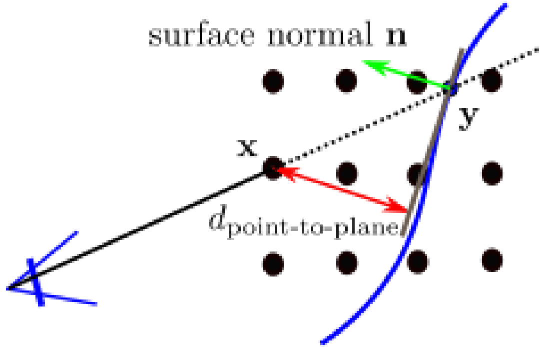
1) Projective Point-To-Point

$$\mathbf{x} = (x, y, z)^\top$$
$$(i, j)^\top = \pi(\mathbf{x})$$
$$d_{\text{point-to-point}}(\mathbf{x}) := z - I_d(i, j)$$

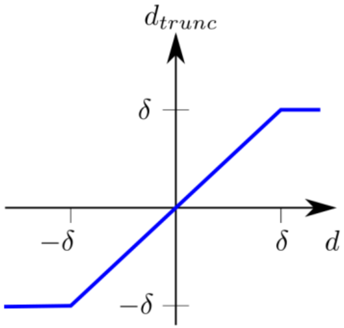


2) Projective Point-To Plane

$$d_{\text{point-to-plane}}(\mathbf{x}) := (\mathbf{y} - \mathbf{x})^\top \mathbf{n}(i, j)$$



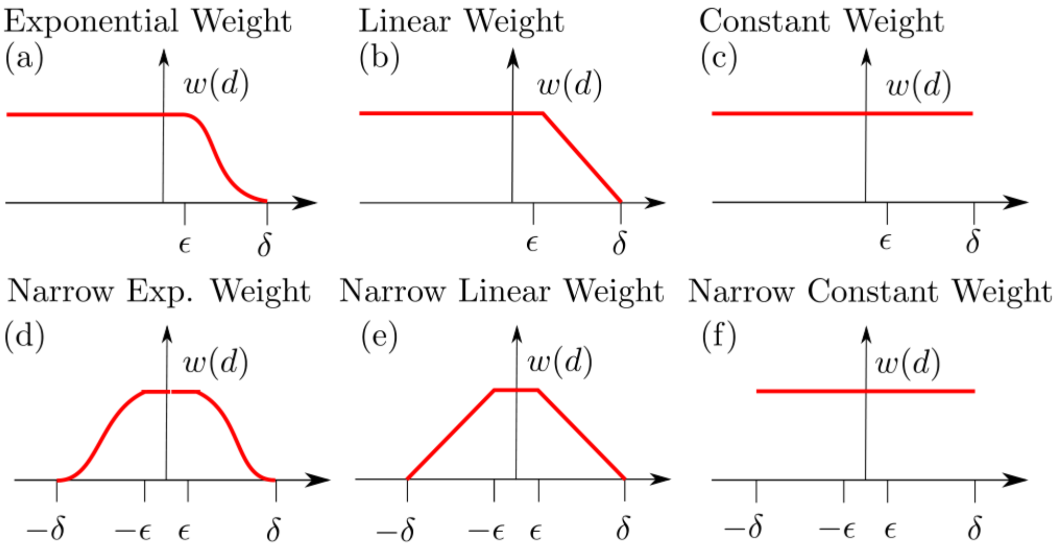
SDF - Truncation and Weighting



Projective distances are **truncated**

- only the small band near the **zero-crossing** is relevant
- **large distances** values are highly **erroneous**

Observations are **weighted** according to their **confidence**



Data Fusion and 3D Reconstruction

Given a sequence of approximated distance measurements and the weights for voxel cell, find the best possible estimate for $\psi(x)$.

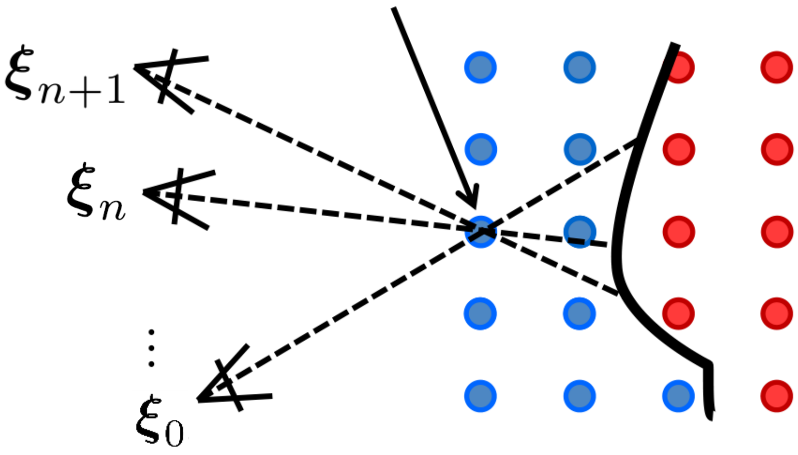
$$p(d_1, w_1, \ldots, d_n, w_n \mid \psi) \propto \prod_{i=1}^{n} \exp\left(-\frac{1}{2} w_i (\psi - d_i)^2\right)$$

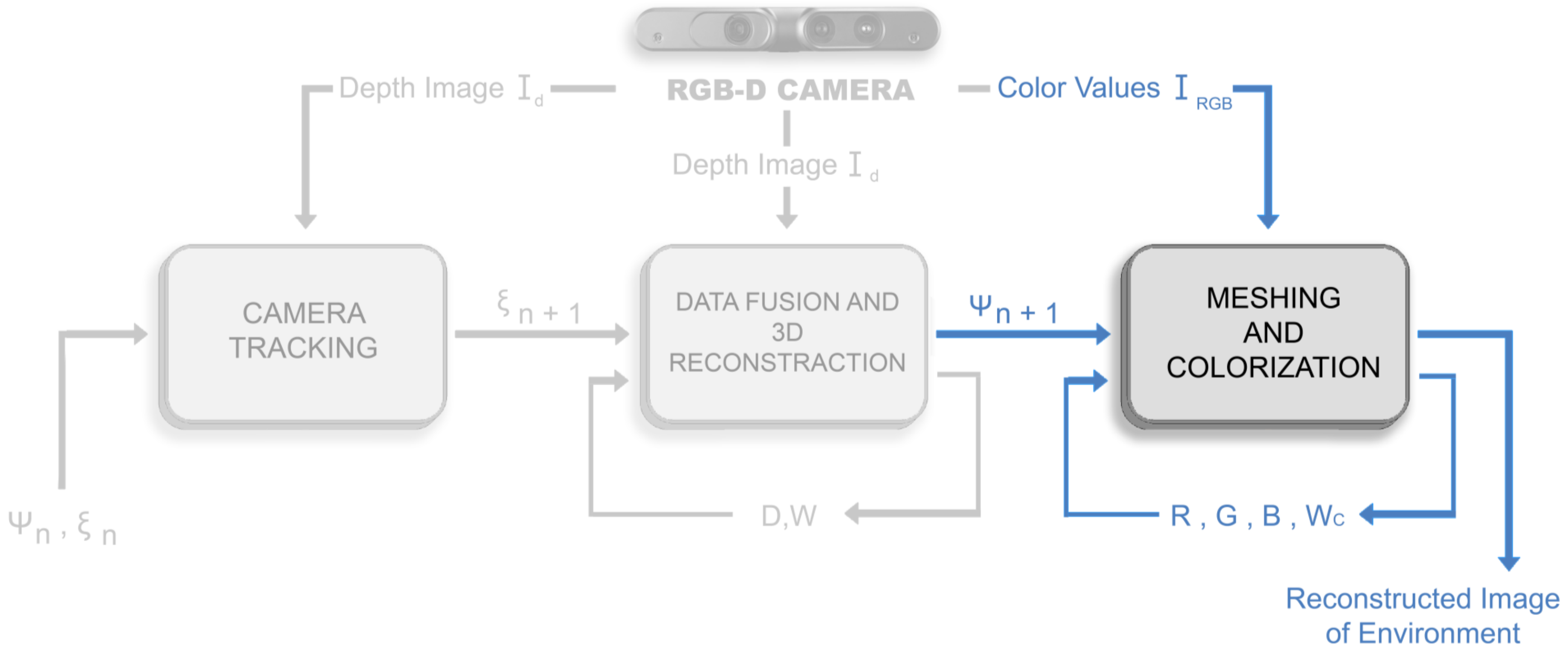
$$L(\psi) = \sum_{i=1}^{n} \frac{1}{2} w_i (\psi - d_i)^2$$

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

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

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

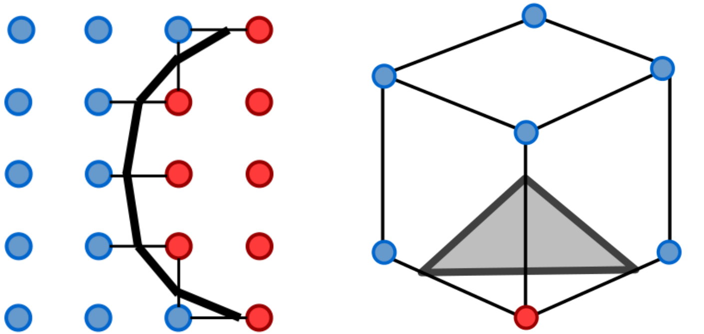




Depth Image I_d
RGB-D CAMERA
Color Values I_{RGB}
Depth Image I_d
CAMERA TRACKING
ξ_{n+1}
DATA FUSION AND 3D RECONSTRACTION
Ψ_{n+1}
MESHING AND COLORIZATION
Ψ_n, ξ_n
D,W
R , G , B , Wc
Reconstructed Image of Environment

Meshing and Colorization

Marching cubes: Creates a triangle mesh from the zero-crossings in the signed distance function.



Colorization : Texture represented with the channels R, G, B and W for the voxels that are sufficiently close to the surface.

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

$$R \leftarrow \frac{W_c R + w_c^{n+1} r}{W_c + w_c^{n+1}}$$

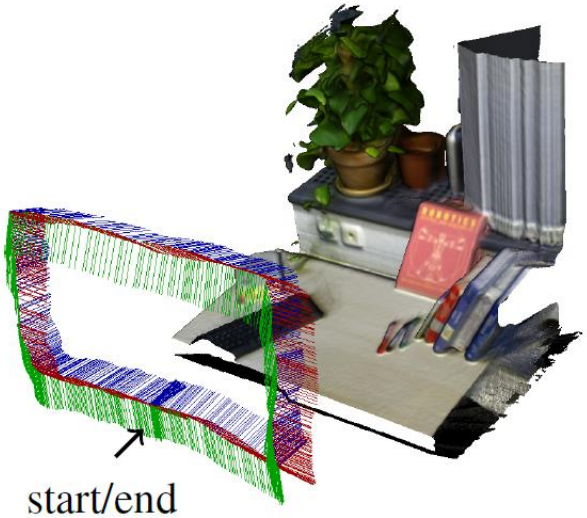
$$G \leftarrow \frac{W_c G + w_c^{n+1} g}{W_c + w_c^{n+1}}$$

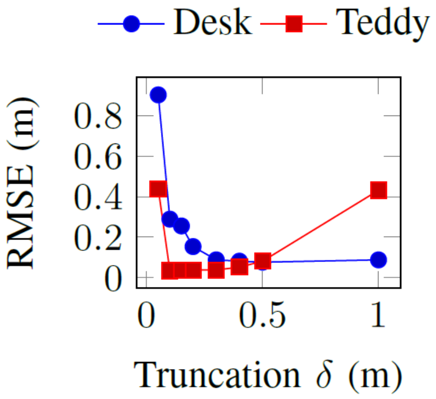
$$B \leftarrow \frac{W_c B + w_c^{n+1} b}{W_c + w_c^{n+1}}$$

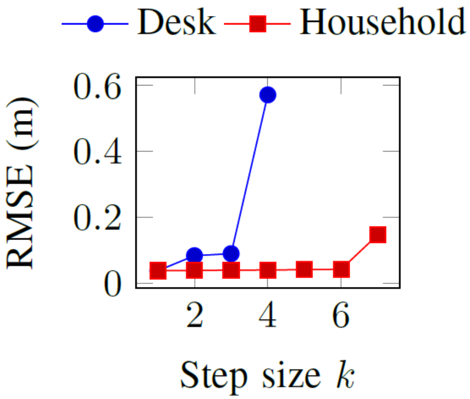
$$w_c^{n+1} = w_{n+1} \cos\theta$$

Results

Method	Res.	Teddy	F1 Desk	F1 Desk2	F3 Household	F1 Floor	F1 360	F1 Room	F1 Plant	F1 RPY	F1 XYZ
KinFu	256	0.156 m	0.057m	0.420 m	0.064 m	Failed	0.913 m	Failed	0.598 m	0.133 m	0.026 m
KinFu	512	0.337 m	0.068 m	0.635 m	0.061 m	Failed	0.591 m	0.304 m	0.281 m	0.081 m	0.025 m
Point-To-Plane	256	**0.072 m**	0.087 m	0.078 m	0.053 m	0.811 m	0.533 m	0.163 m	0.047 m	0.047 m	0.029 m
Point-To-Plane	512	0.101 m	0.059 m	0.623 m	0.053 m	0.640 m	0.206 m	0.105 m	**0.041 m**	**0.042 m**	0.026 m
Point-To-Point	256	0.086 m	0.038 m	**0.061 m**	**0.039 m**	0.641 m	0.420 m	0.121 m	0.047 m	0.047 m	**0.021 m**
Point-To-Point	512	0.080 m	**0.035 m**	0.062 m	0.040 m	**0.567 m**	**0.119 m**	**0.078 m**	0.043 m	**0.042 m**	0.023 m
RGB-D SLAM		0.111 m	0.026 m	0.043 m	0.059 m	0.035 m	0.071 m	0.101 m	0.061 m	0.029 m	0.013 m







Results

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

	Duration per Frame (ms)
Proposed Alg.	23 ms
KinFu	20 ms
RGB-D SLAM	100 - 250 ms

	Duration for pose optimization	Duration for data fusion
m = 256	19.4 ms	3.7 ms
m = 512	31.1 ms	21.6 ms

	SDF size on RAM	Color grid size on RAM
m = 256	128 MB	256 MB
m = 512	1 GB	2 GB

Conclusion

- **Absolute metric** information and minimal **drift**
  - ➢ Ideal for home decoration and refurbishment measures
- Highly efficient : **Real-Time** capable on a laptop with a Quadro GPU
- Outperforms ICP-based methods such as KinFu
- Comparable performance with bundle adjustment with significantly less computation
  - ➢ Fails in cases where only co-planar surfaces are

Supplementary Documents

1. B. Curless and M. Levoy. A volumetric method for building complex models from range images. In SIGGRAPH,1996.
2. R.A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A.J. Davison, P. Kohli, J. Shotton, S. Hodges, and A.W. Fitzgibbon. KinectFusion: Real-time dense surface mapping and tracking. In ISMAR, pages 127–136, 2011
3. KinectFusion Implementation in the Point Cloud Library (PCL). http://svn.pointclouds.org/pcl/trunk/
4. F. Endres, J. Hess, N. Engelhard, J. Sturm, D. Cremers, and W. Burgard. An evaluation of the RGB-D SLAM system. In ICRA, May 2012
5. J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers. A benchmark for the evaluation of RGB-D SLAM systems. In IROS, 2012. https://vision.in.tum.de/data/datasets/rgbd-dataset

Sensor RGB& Depth& Microphone*2

Depth Image Size VGA (640x480) : 30 fps
QVGA (320x240): 60 fps

Resolution SXGA (1280*1024)

Field of View 58° H, 45° V, 70° D (Horizontal, Vertical, Diagonal)

Distance of Use Between 0.8m and 3.5m

Power Consumption Below 2.5W

Interface USB2.0/ 3.0

Platform Intel X86 & AMD

OS Support Win 32/64 : XP , Vista, 7, 8
Linux Ubuntu 10.10: X86,32/64 bit
Android(by request)

Software Software development kits(OpenNI SDK bundled)

Programming Language C++/C# (Windows)
C++(Linux)
JAVA

Operation Environment Indoor

Dimensions 18 x 3.5 x 5

Related Work

- A volumetric method for building complex models from range images [Curless and Levoy, 1996]
  - Represent **distance to surface** in a voxel grid
  - Data fusion of depth images with SDF

- KinectFusion: Real-time dense surface mapping and tracking [Newcombe et al., 2011]
  - Generate synthetic depth image from SDF
  - Iterative closest point (ICP) between current and synthetic image