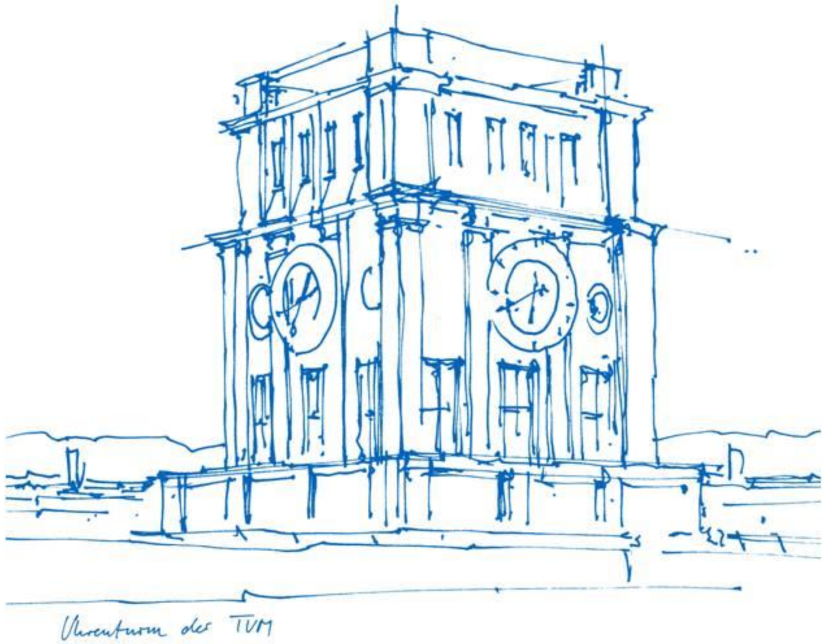
Fusion4D: Real-time Performance Capture of Challenging Scenes

Dou et al. 2016, Microsoft Research

Anna Baumeister

Technische Universität München

07. Mai 2019



Goal: Reconstruction of a live performance

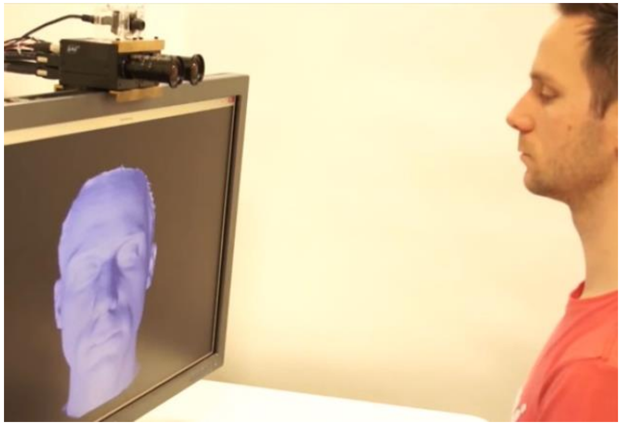
- With large frame-to-frame motion
- With drastic topology changes
- In real-time

Related Work

Real-time Non-rigid Reconstruction using an RGB-D Camera | Zollhöfer et al., 2014

- One RGB-D camera
- Template acquisition under rigid deformations
- Reconstruction by surface deformation under nonrigid transformations

- Real-Time
- Limitation: Fast motion, topology changes



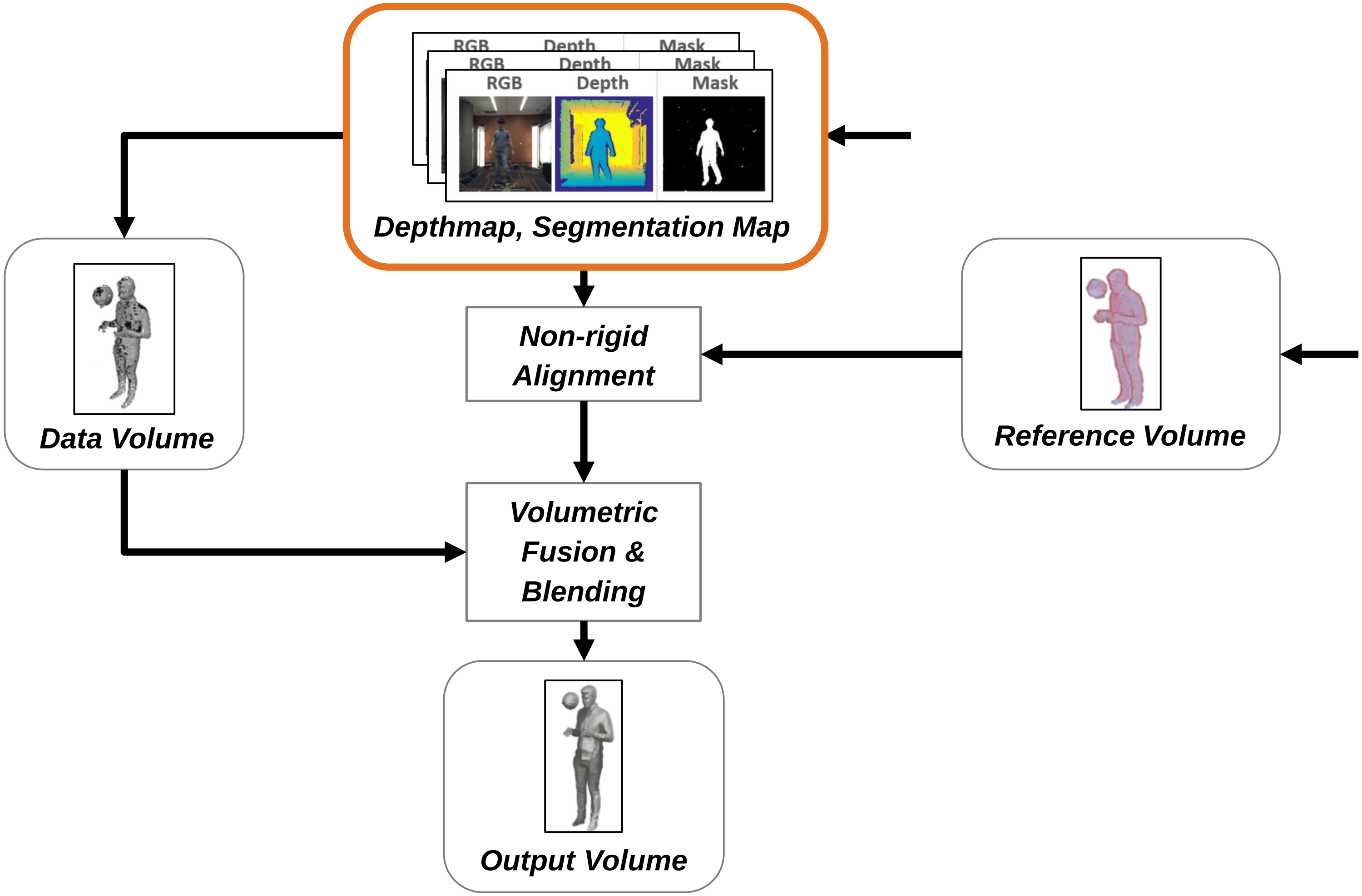
Related Work

DynamicFusion: Reconstruction and tracking of non-rigid scenes in real-time
Newcombe et al., 2015

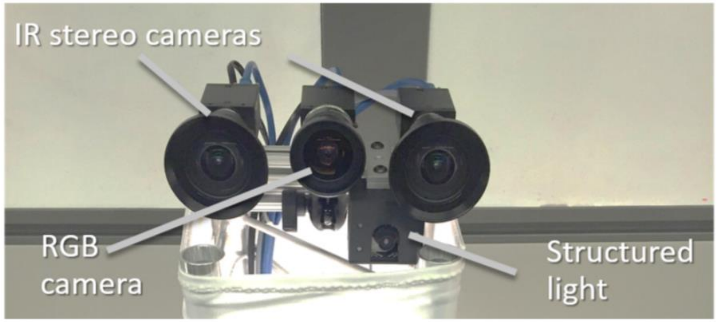
- One Kinect Sensor
- Nonrigid Volumetric Fusion

- Real Time
- Limitation: Fast motion, topology changes

The Fusion4D Pipeline



Raw Depth Acquisition and Preprocessing

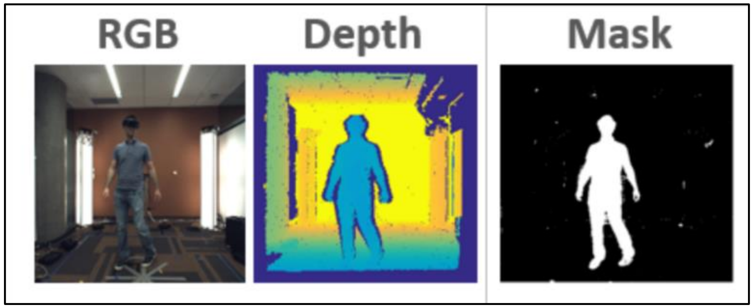


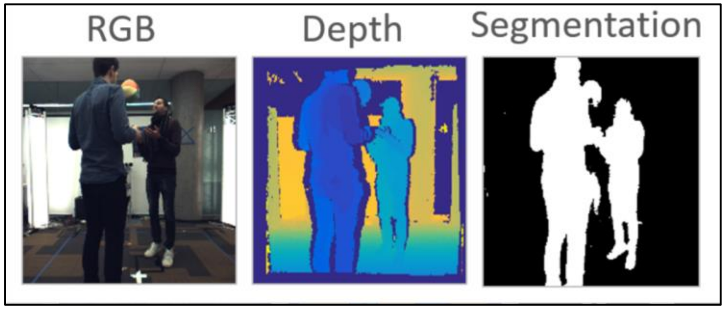
- 8 Depthmaps and RGB images
- No greenscreen, natural lighting

- Active stereo setup
- Two NIR cameras plus random IR dot pattern projector

Holoportation: Virtual 3D Teleportation in Real-time | Orts-Escolano et al. 2016

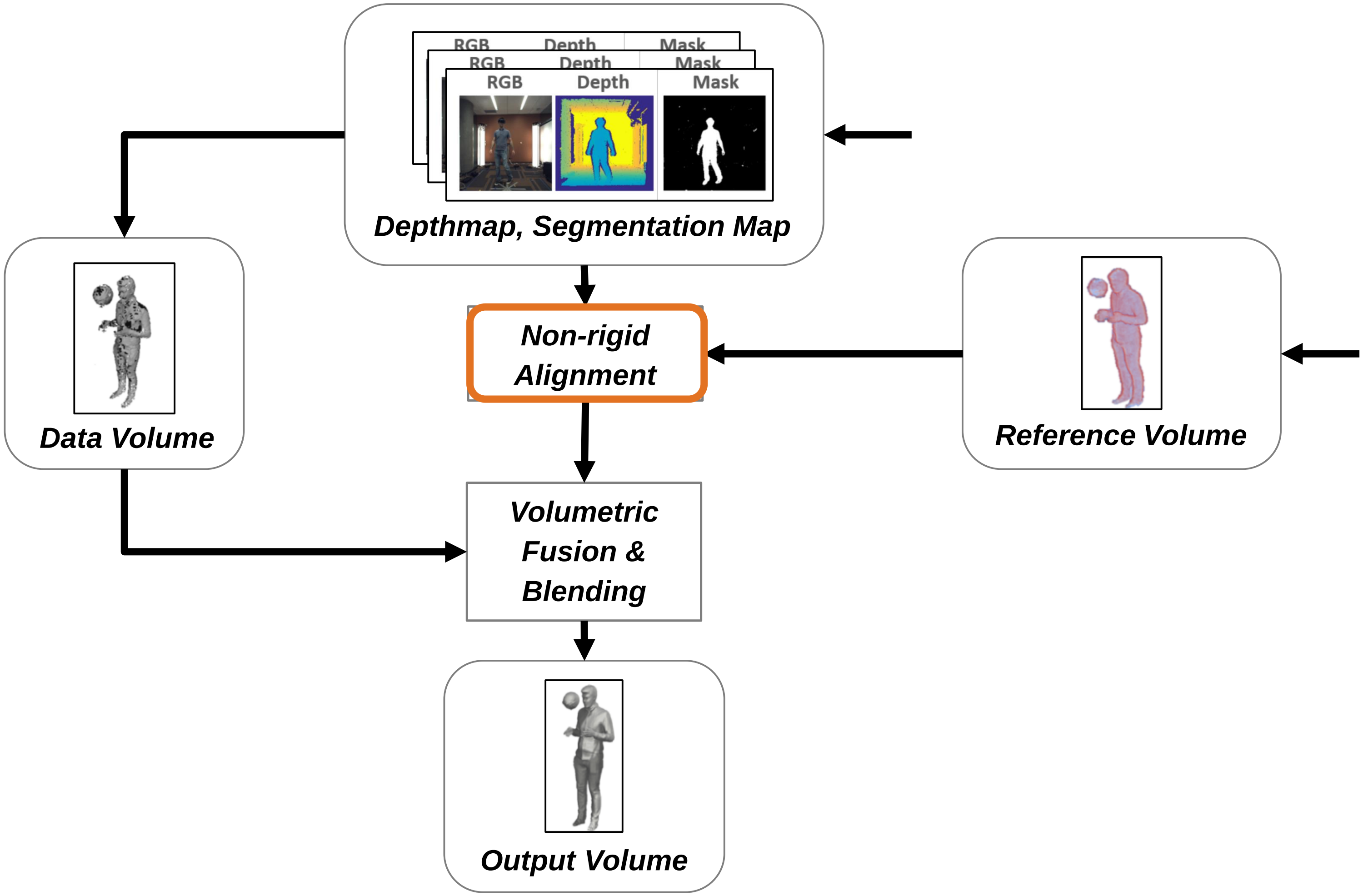
Raw Depth Acquisition and Preprocessing



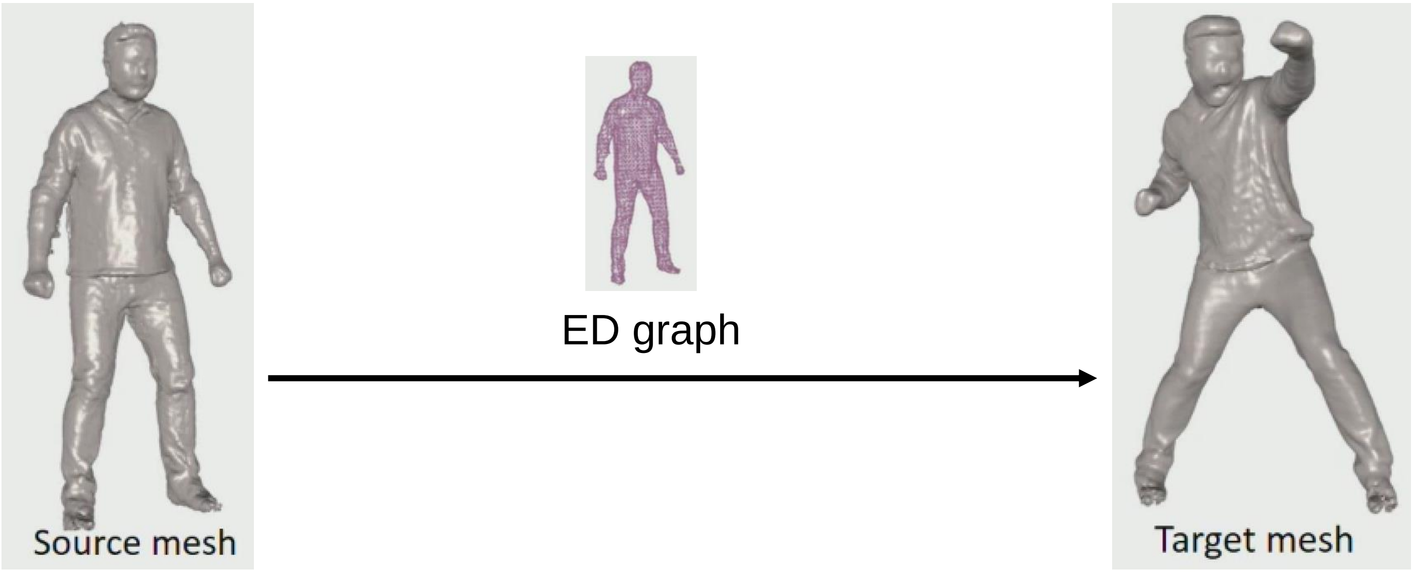


- Background Model of the empty scene for each camera
- Real time foreground/background segmentation for each frame

The Fusion4D Pipeline



Embedded Deformation Model



- The source mesh is represented by a set of uniformly sampled ED nodes
- Each ED node represents a weighted average of the surrounding vertices
- The nonrigid deformation of the entire mesh is modelled by the deformation of the ED nodes

Energy Function

$$E(G) = \lambda_{\text{data}} E_{\text{data}}(G) + \lambda_{\text{hull}} E_{\text{hull}}(G) + \lambda_{\text{corr}} E_{\text{corr}}(G) + \lambda_{\text{rot}} E_{\text{rot}}(G) + \lambda_{\text{smooth}} E_{\text{smooth}}(G)$$

E_{data}: Penalizes misalignments between the deformed model and the data

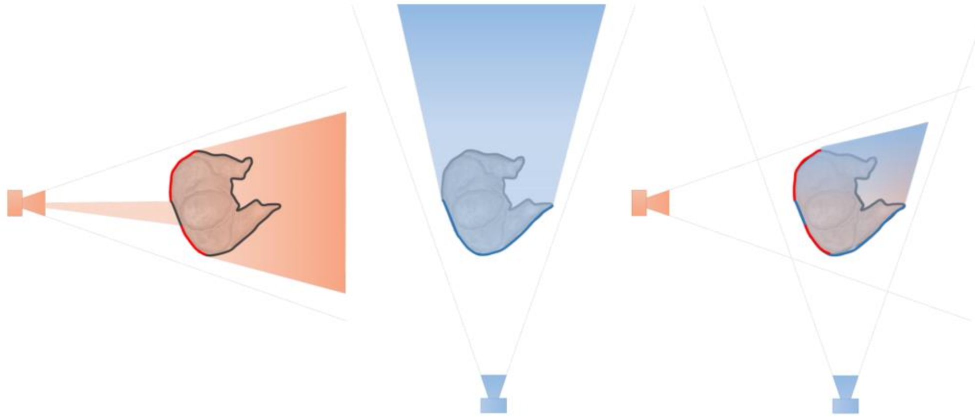
E_{hull}: Penalizes points that lie outside the visual hull

E_{corr}: Established correspondences between consecutive data frames (RGB)

E_{rot}: Encourages local deformation to be close to a rotation

E_{smooth}: Encourages neighboring ED nodes to be transformed similarly

Energy Function – Visual Hull



- The visual hull provides a bounding box in which all observed data points must lie
- It is the intersection of the projections of the object silhouettes from each camera

Energy Function – Optimization

- The energy function is a sum of square residuals and can be reformulated to

$$\mathbf{E(x) = f(x)^T f(x)}$$

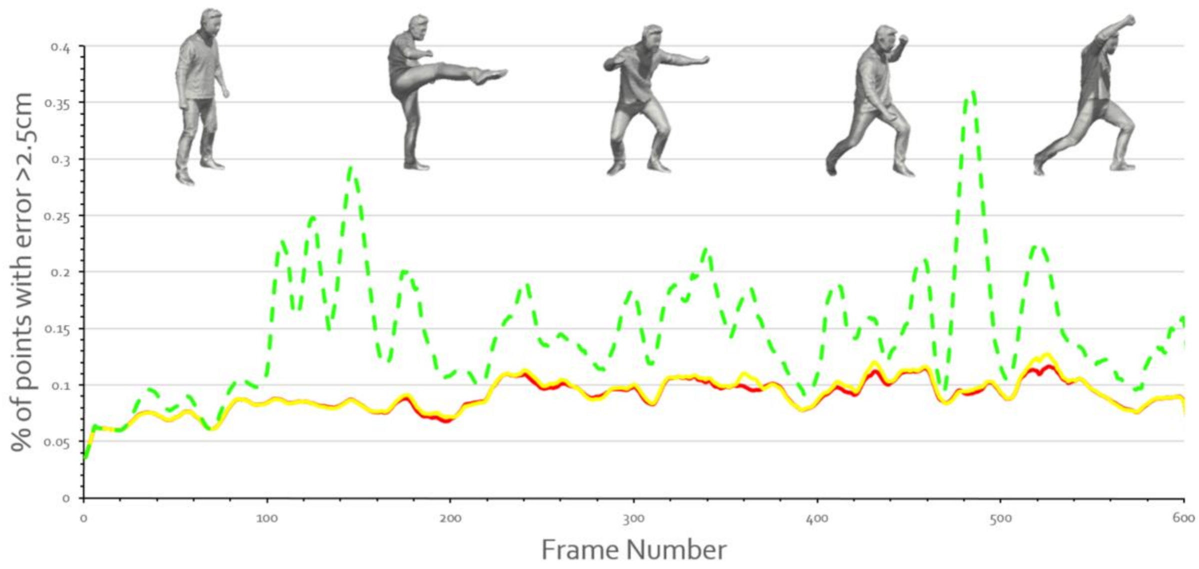
- Optimization using the Levenberg-Marquardt algorithm (dampened least squares)
- Step direction h is obtained by solving:

$$\mathbf{(J^T J + \mu I)\, h = -J^T f}$$

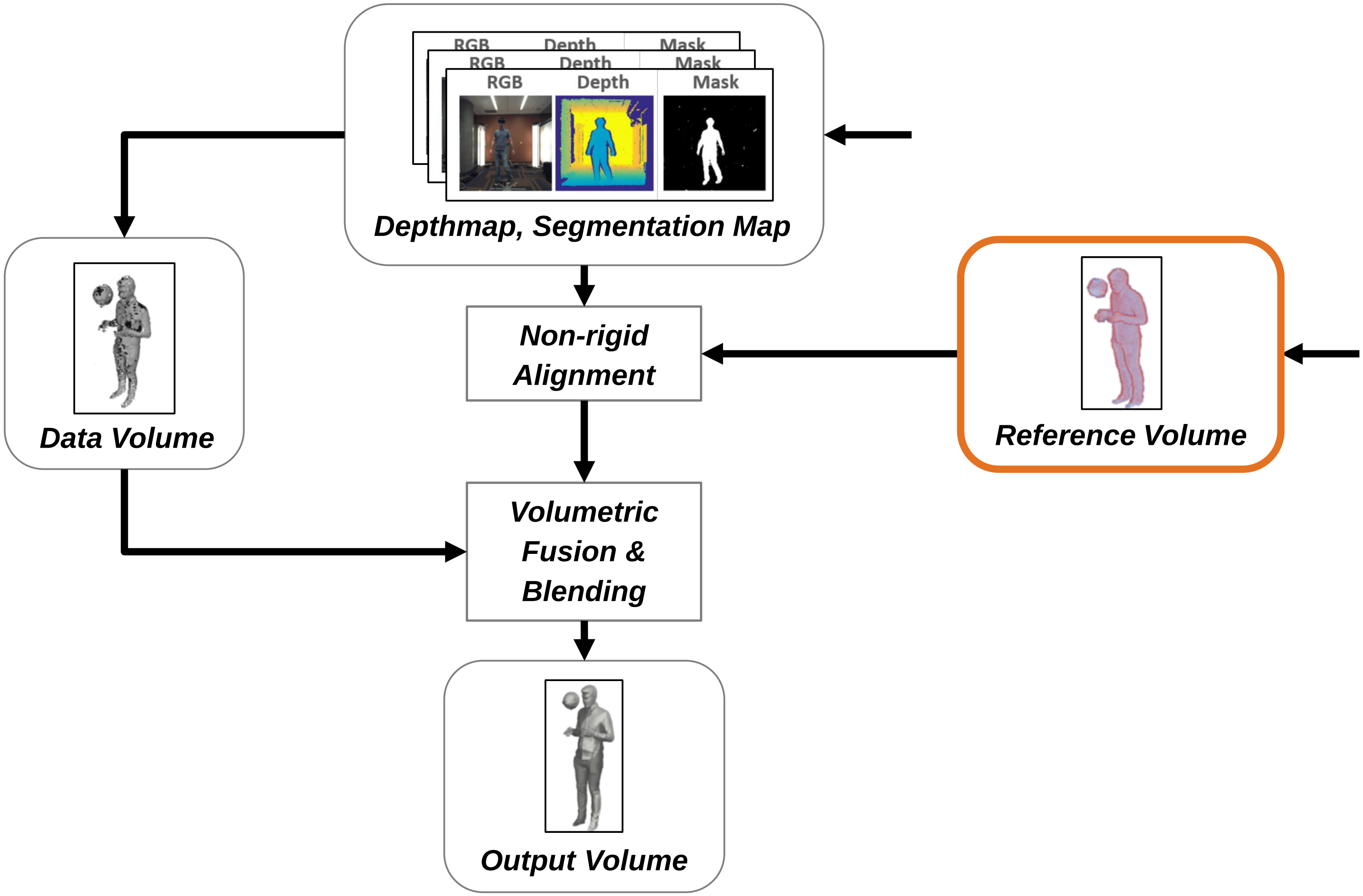
- Damping factor μ is adjusted to make the solver more or less aggressive

Energy Function – Evaluation of J^TJ

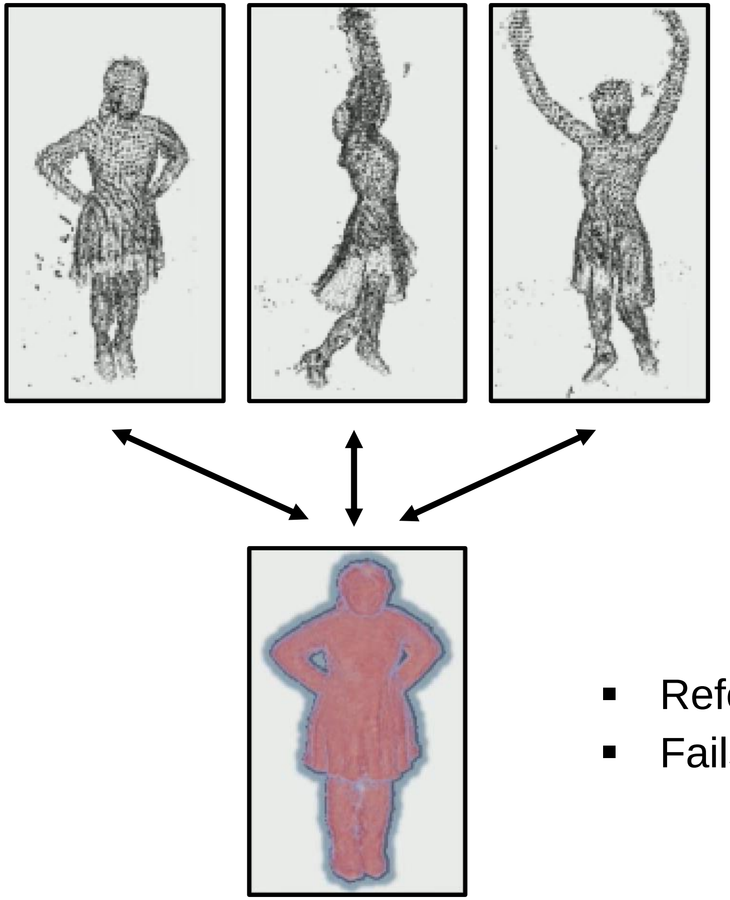
- DynamicFusion approximates J^TJ as a block diagonal matrix
- Idea: Use this approximation to J^TJ as a starting point for an iterative solve using the *preconditioned conjugate gradient* method



The Fusion4D Pipeline

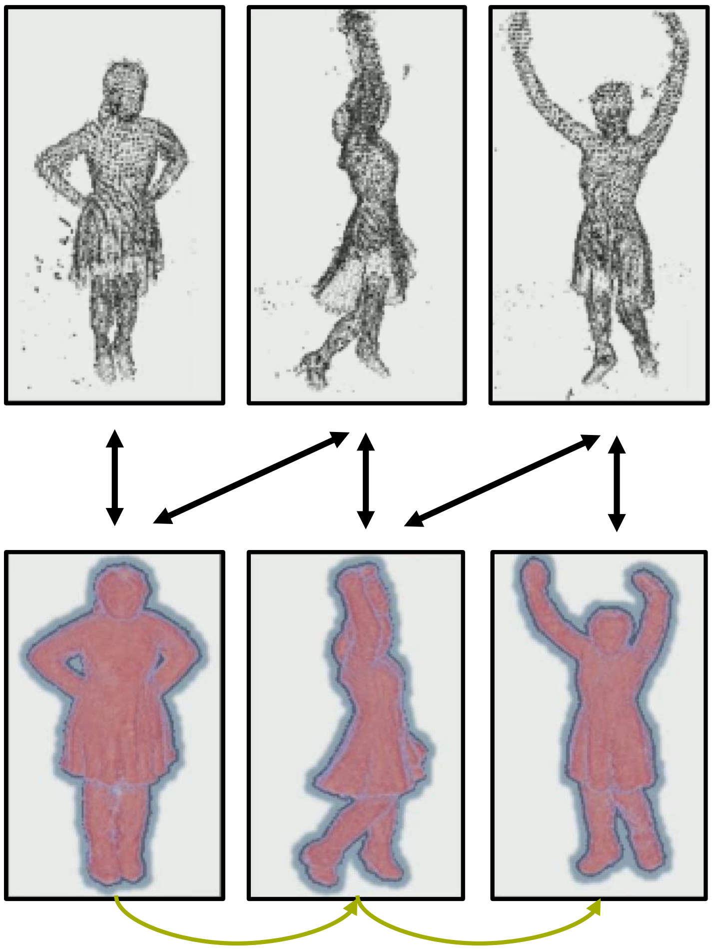


Data Fusion – Reference to Frame



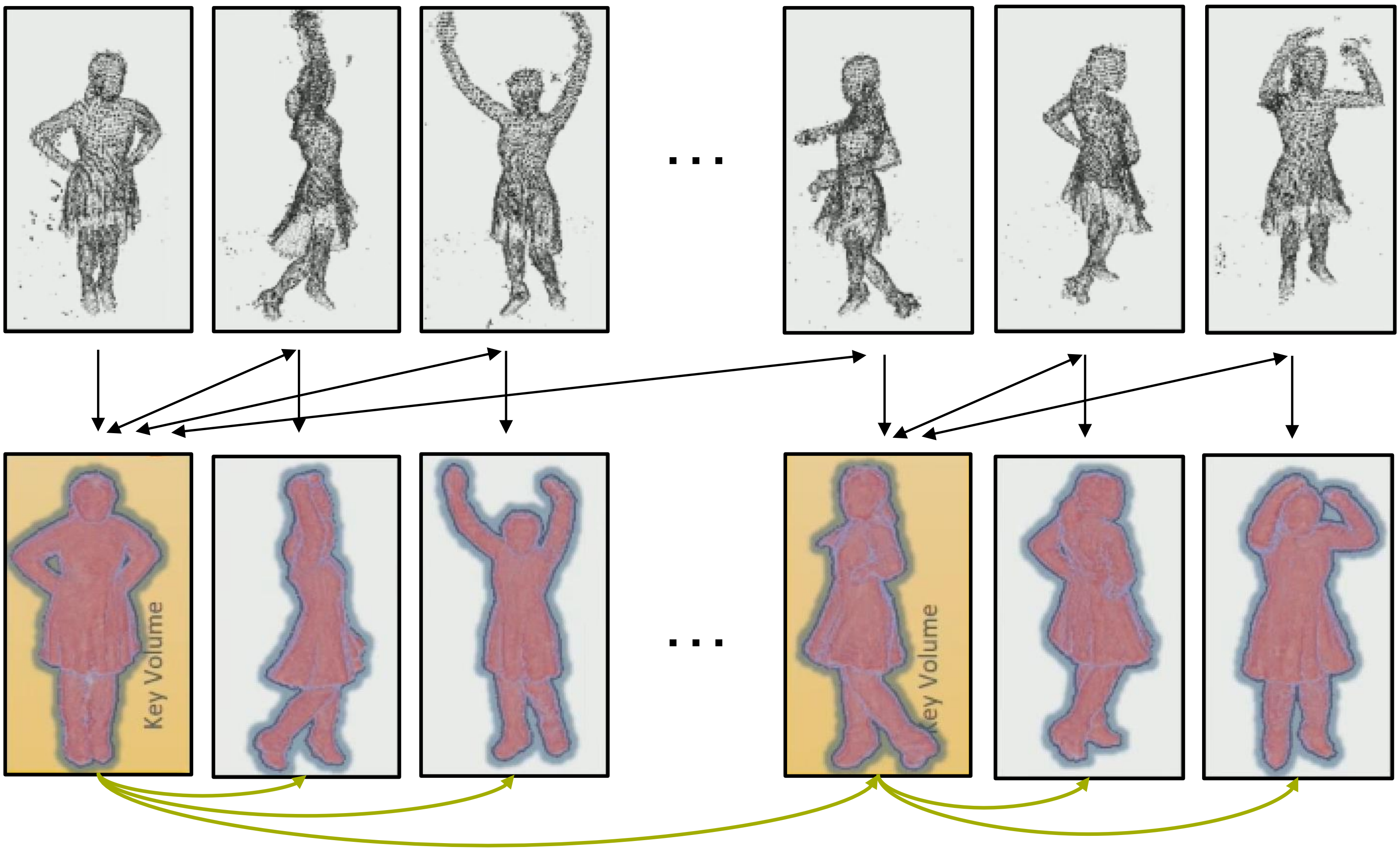
- Reference volume is refined over time
- Fails on a topology change, tracking failure

Data Fusion – Frame to Frame

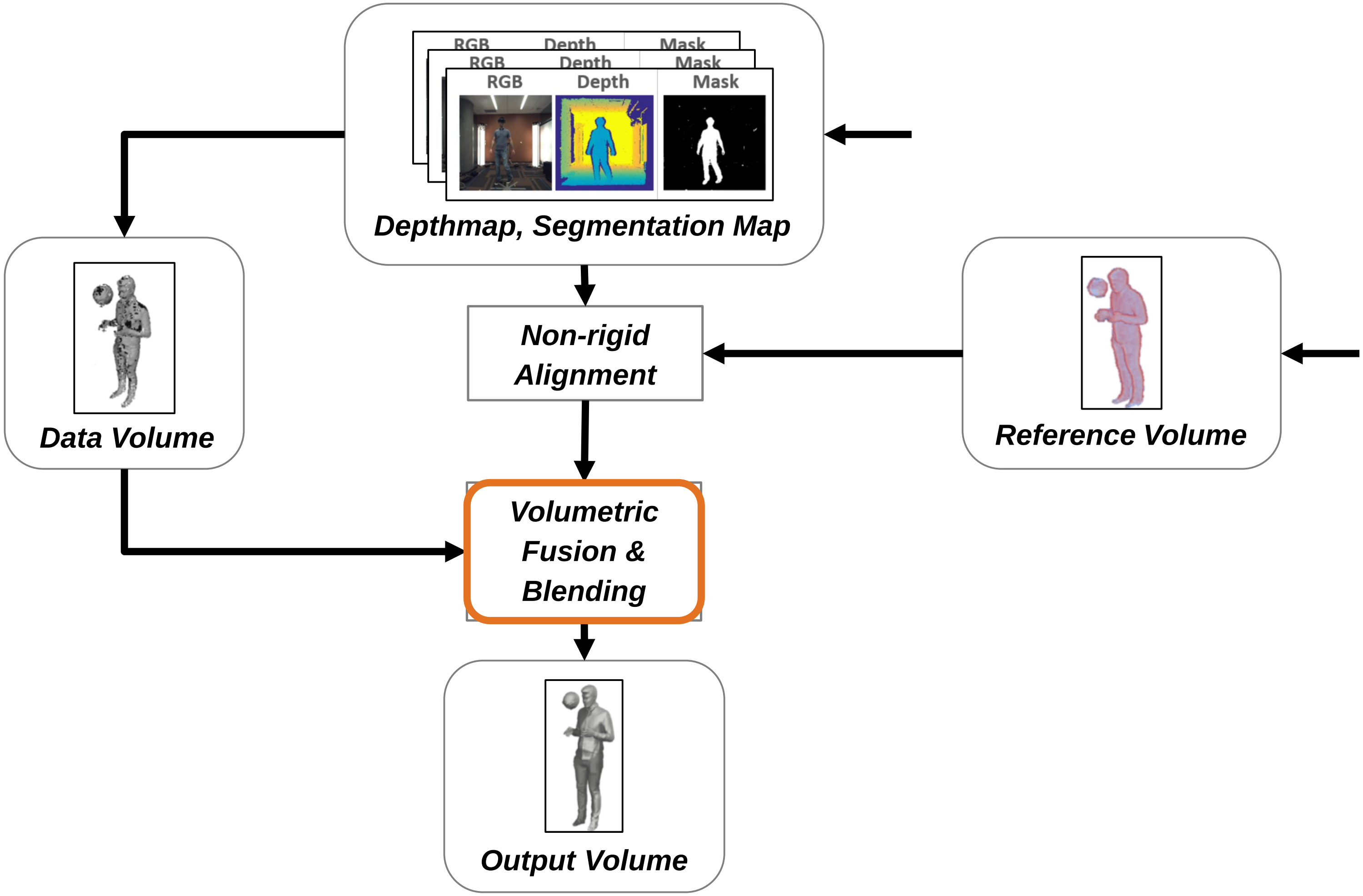


- Geometry blur due to resampling
- Robust to topology change, tracking failure

Data Fusion – Key Volumes



The Fusion4D Pipeline

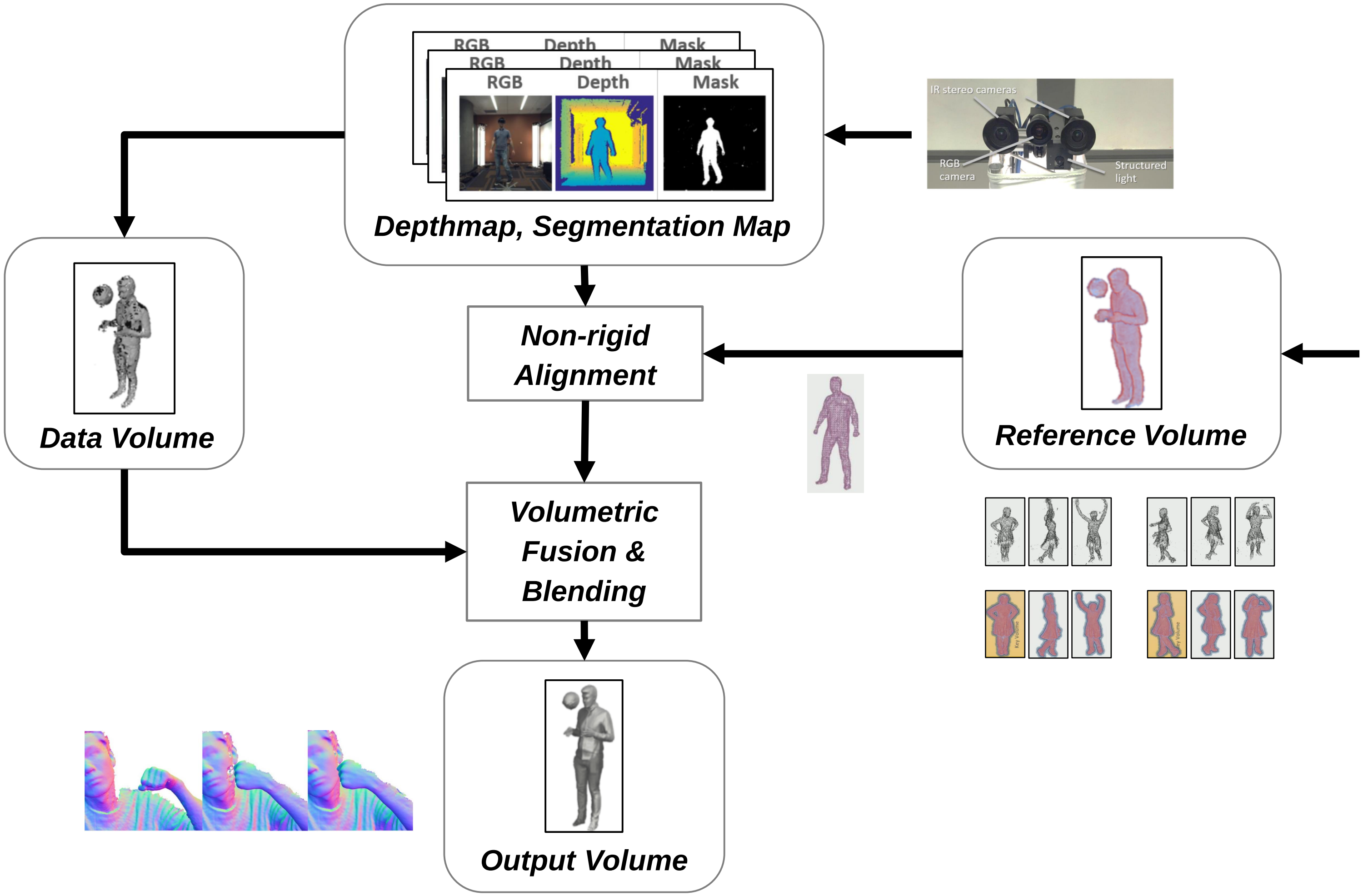


Fusion at the Data Frame

- Before fusing a warped voxel, two tests are performed and the voxel is rejected if it fails either:
  1. **Voxel Collision**: Reference voxels contributing to different surface areas might collide after warping (e.g. clapping hands)
  2. **Voxel Misalignment**: Only accept reference voxels whose ED node has a small alignment error

real-time multi-view reconstruction

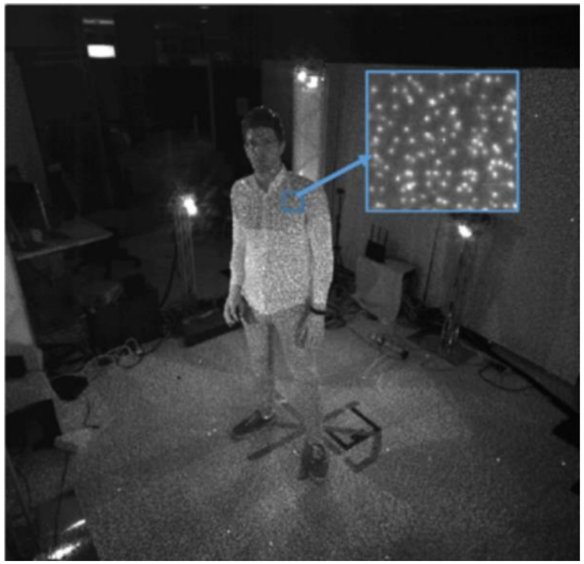
The Fusion4D Pipeline



Backup

Raw Depth Acquisition and Preprocessing

Depth Estimation



- Active Stereo Setup - two NIR cameras plus random IR dot pattern projector on each camera rig
- Real-time Depth estimation using PatchMatch Stereo Algorithm

PatchMatch Stereo - Stereo Matching with Slanted Support Windows | Bleyer et al. 2011

Energy Function – Data Term

$$E_{\text{data}}(G) = \sum_{n=1}^{N} \sum_{m \in \mathcal{V}_n(G)} \left(\tilde{\mathbf{n}}_m(G)^\top \left(\tilde{\mathbf{v}}_m(G) - \Gamma_n(\tilde{\mathbf{v}}_m(G)) \right) \right)^2$$

- Projective point-to-plane term as an approximation
- Penalizes misalignments between the deformed model and the data

Energy Function – Rotational Term

$$E_{\text{rot}}(G) = \sum_{k=1}^{K} \|A_k^T A_k - \mathbf{I}\|_F + \sum_{k=1}^{K} (\det(A_k) - 1)^2$$

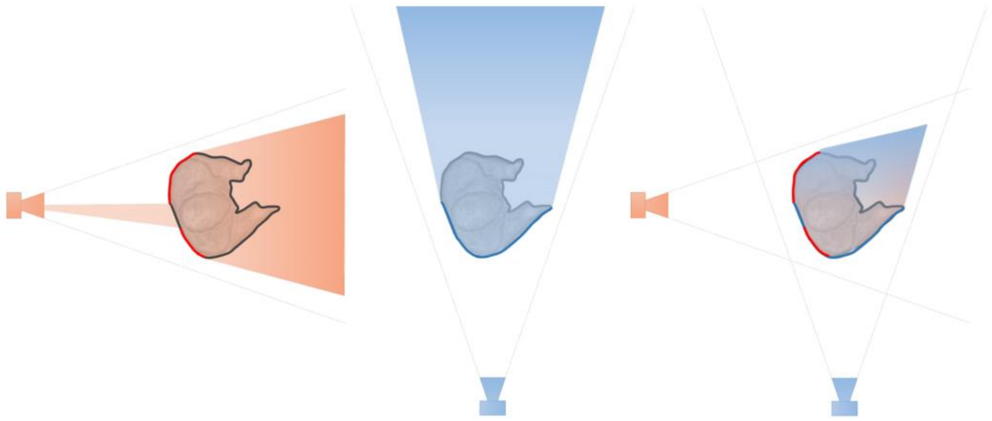
- Encourages local deformation to be close to a rotation

Energy Function – Smoothness Term

$$E_{\text{smooth}}(G) = \sum_{k=1}^{K} \sum_{j \in \mathcal{N}_k} w_{jk} \rho(\| A_j(\mathbf{g}_k - \mathbf{g}_j) + \mathbf{g}_j + \mathbf{t}_j - (\mathbf{g}_k + \mathbf{t}_k) \|^2)$$

- Encourages neighboring affine transformations to be similar

Energy Function – Visual Hull



- The visual hull provides a bounding box in which all observed data points must lie
- It is the intersection of the projections of the object silhouettes from each camera

$$E_{\text{hull}}(G) = \sum_{m=1}^{M} \mathcal{H}(\mathcal{T}(\mathbf{v}_m; G))^2$$

Energy Function – Correspondence Term

$$E_{\text{corr}}(G) = \sum_{n=1}^{N} \sum_{f=1}^{F_n} \rho(\|\mathcal{T}(\mathbf{q}_{nf}; G) - P_n(u_{nf})\|^2)$$

- Establish correspondences between two consecutive data frames

The Global Patch Collider | Wang et al., 2016

Optimization parameters

$$E(G) = \lambda_{\text{data}} E_{\text{data}}(G) + \lambda_{\text{hull}} E_{\text{hull}}(G) + \lambda_{\text{corr}} E_{\text{corr}}(G) + \lambda_{\text{rot}} E_{\text{rot}}(G) + \lambda_{\text{smooth}} E_{\text{smooth}}(G)$$

$$G = \{R, T\} \cup \{A_k, \mathbf{t}_k\}_{k=1}^{K}$$

G:	Deformation parameters
R:	Global rotation
T:	Global translation
A_k:	Affine transformation of ED node *k*
t_k:	Translation of ED node *k*