Felix Scheidhammer
Supervised by Benjamin Busam
Recent Trends in 3D Computer Vision

Fusion4D: Real-time Performance Capture of Challenging Scenes

Recent trends in 3D computer vision

Technische Universität München
JOHNS HOPKINS
WHITING SCHOOL
of ENGINEERING

Fusion4D: Real-time Performance Capture of Challenging Scenes

By:
Mingsong Dou, Sameh Khamis, Yuri Degtyarev, Philip Davidson, Sean Ryan Fanello
Adarsh Kowdle, Sergio Orts Escolano, Christoph Rhemann, David Kim, Jonathan Taylor
Pushmeet Kohli, Vladimir Tankovich, Shahram Izadi
Microsoft Research

Conference: SIGGRAPH2016
July 2016

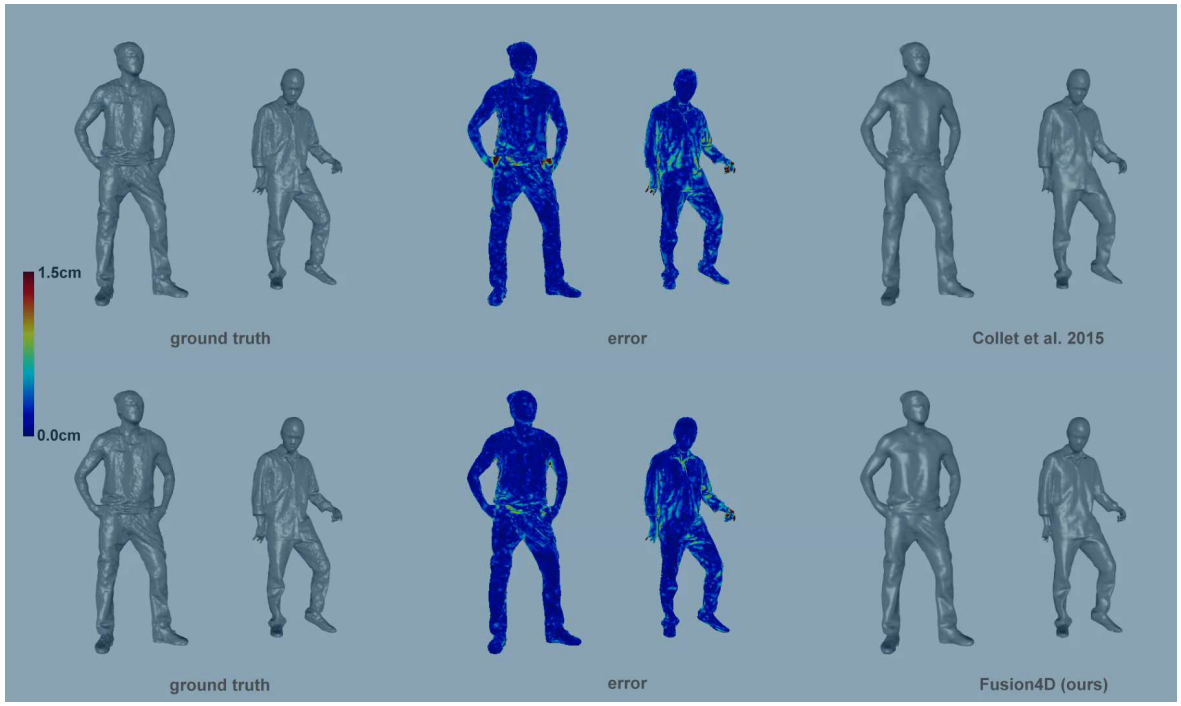
Technische Universität München + Johns Hopkins Whiting School of Engineering

Structure

- Related Work
- Overview
- Nonrigid Motion Field Estimation
- Fusion
- Results
- Conclusion
- Questions

Related Work

- Offline Approach Collet
  - 30s per frame
  - 106 cameras -> 24 depthmaps
  - controlled studio setting
- Dynamic Fusion 2015
  - incrementally updated Model
  - only slow motions
  - no topology-changes
- Zollhöfer 2014
  - Template-based
    - Fixed model



(2)

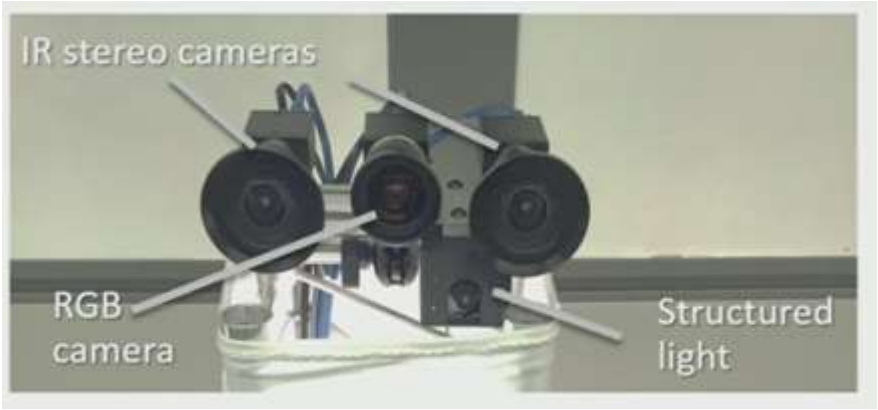
Key Contributions

- No prior
- Robust to large motion and topology changes
- Multi-view RGBD
- Real-time

Overview

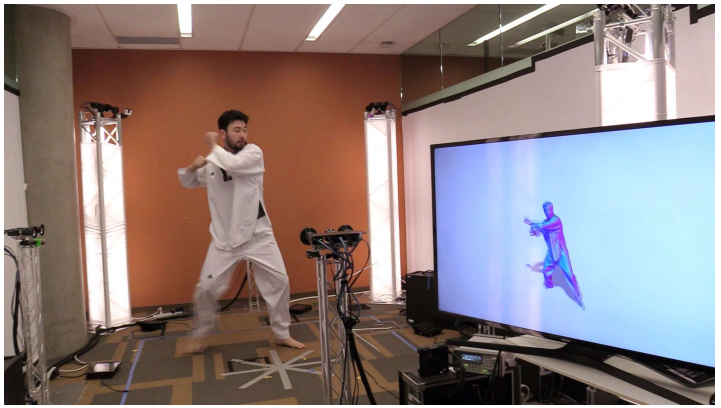
- Studio: (Bild aus Teaser)
  - Trinocular cameres (2xIR + 1xRGB) 1 megapixel
  - 24 cameras -> 8 depthmaps

(2)



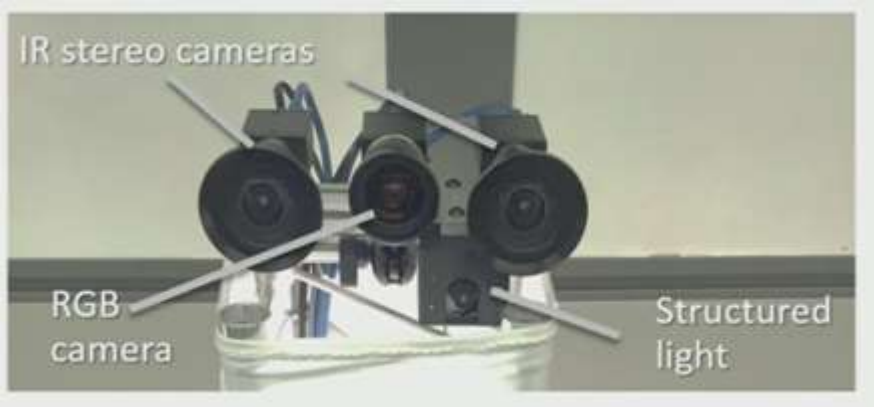
(1)

Overview – Setup

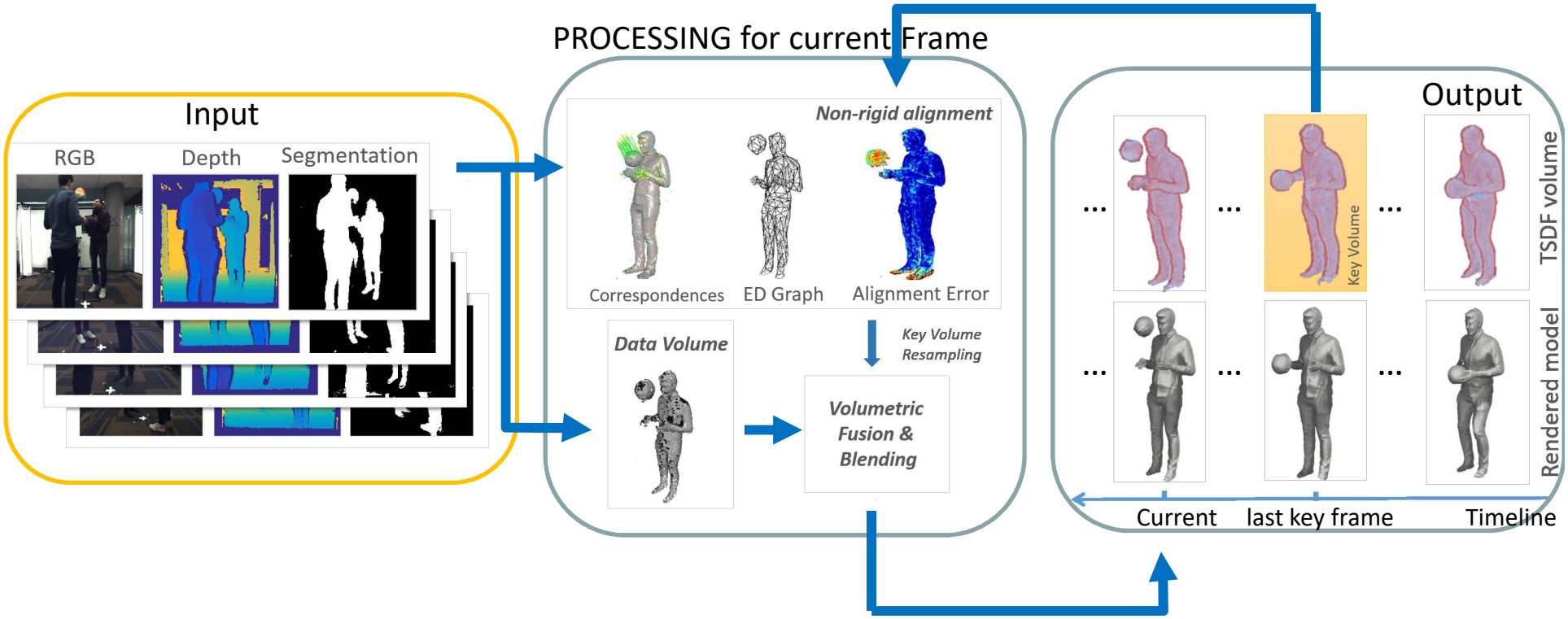


- Studio:
  - Trinocular cameres (2xIR + 1xRGB) 1 megapixel
  - 24 cameras -> 8 depthmaps

- Hardware:
  - 12 Computer (Intel Core i7, 3.4GHz CPU, 16GB of RAM and it uses 2 NVIDIA Titan X GPUs)
  - Master Computer (same, but with a single NVIDIA Titan X)

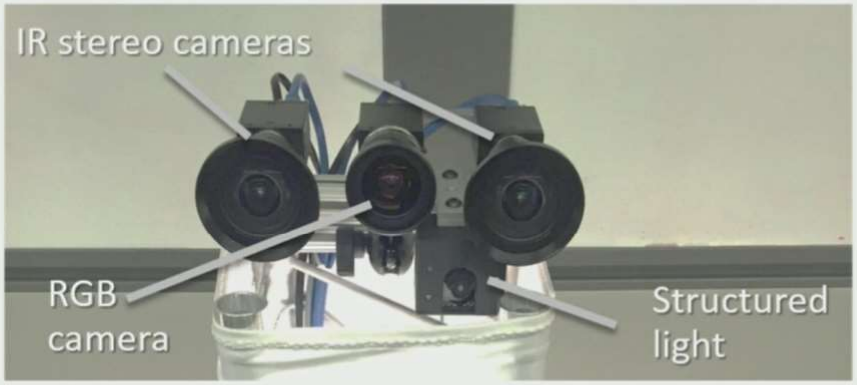


Overview - Pipeline



(2)

Depth Acquisition



Holoportation: Virtual 3D Teleportation in Real-time
Orts-Escolano et al. [UIST '16]

(1)

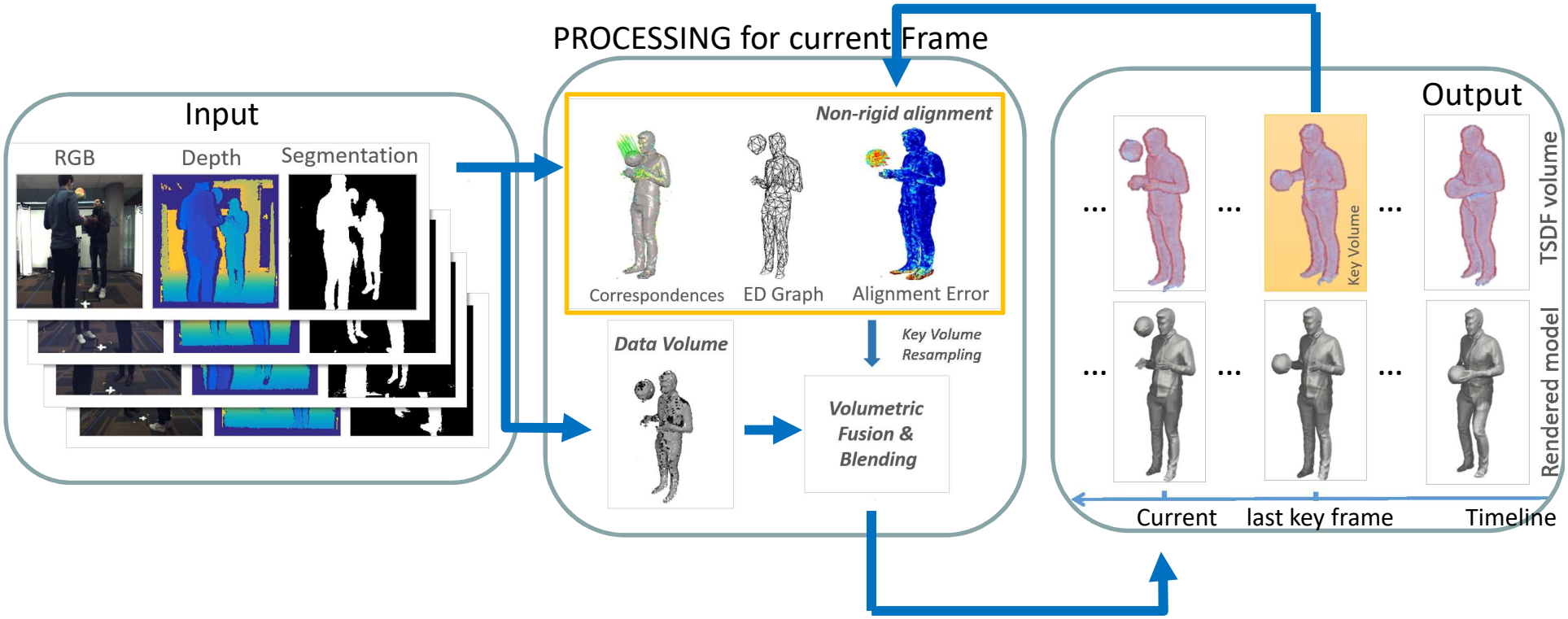
Depth Acquisition

Real-time foreground/background segmentation

Krähenbühl et al. [ICML ’13], Vineet et al. [CVPR ’08]

(1)

Overview - Pipeline



(1)

Nonrigid Motion Field Estimation

- TSDF of N Depthmaps
- Embedded Deformation(ED)-model to warp the model to align with the raw depth maps
- Energy function E(G) for optimization of the ED-Model

Deformation field

- (Embedded Deformation)ED-model
  - Set of K ED-nodes with sampling locations $g_k \in R^3$ from the mesh

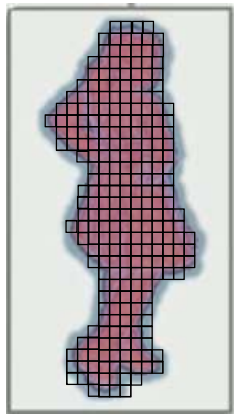
- Deformation (warp) of each point then is:

(1)

$$\mathcal{T}(vm;G) = R \sum_{k \in s_m} w_k^m [A_k(v - g_k) + g_k + t_k] + T$$

- and the normal $\mathcal{T}^{\perp}(n_m;G) = R \sum_{k \in s_m} w_k^m A_k^{-T} n_m$

Deformation field

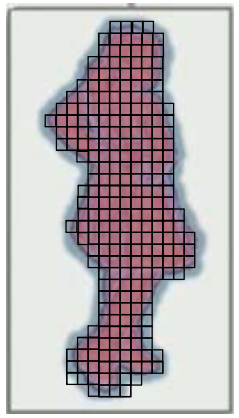


- (Embedded Deformation)ED-model
  - Set of K ED-nodes with sampling locations $g_k \in R^3$ from the mesh
- Deformation (warp) of each point then is:

$$\mathcal{T}(vm;G) = R \sum_{k \in s_m} w_k^m [A_k(v - g_k) + g_k + t_k] + T$$

- and the normal $\mathcal{T}^{\perp}(n_m;G) = R \sum_{k \in s_m} w_k^m A_k^{-T} n_m$

Deformation field



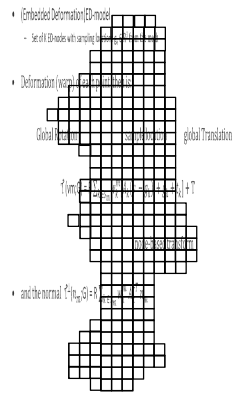
- (Embedded Deformation)ED-model
  - Set of K ED-nodes with sampling locations $g_k \in R^3$ from the mesh
- Deformation (warp) of each point then is:

Global Rotation — sample location — global Translation

$$\mathcal{T}(vm;G) = R \sum_{k \in s_m} w_k^m [A_k(v - g_k) + g_k + t_k] + T$$

- and the normal $\mathcal{T}^{\perp}(n_m;G) = R \sum_{k \in s_m} w_k^m A_k^{-T} n_m$

Deformation field



- (Embedded Deformation)ED-model
  - Set of K ED-nodes with sampling locations $g_k \in R^3$ from the mesh
- Deformation (warp) of each point then is:

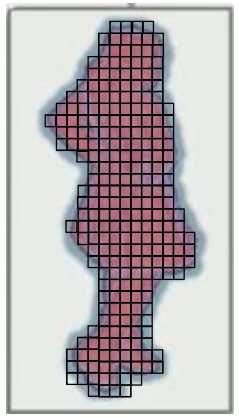
Global Rotation — sample location — global Translation

$$\mathcal{T}(vm;G) = R \sum_{k \in s_m} w_k^m [A_k(v - g_k) + g_k + t_k] + T$$

node-based transform

- and the normal $\mathcal{T}^{\perp}(n_m;G) = R \sum_{k \in s_m} w_k^m A_k^{-T} n_m$

Deformation field



- (Embedded Deformation)ED-model
  - Set of K ED-nodes with sampling locations $g_k \in R^3$ from the mesh
- Deformation (warp) of each point then is:

Global Rotation — global Translation

$$\mathcal{T}(v_m;G) = R \sum_{k \in s_m} w_k^m [A_k(v - g_k) + g_k + t_k] + T$$

weighted by distance — node-based transform

- and the normal $\mathcal{T}^{\perp}(n_m;G) = R \sum_{k \in s_m} w_k^m A_k^{-T} n_m$

Energy Functions – Data Term

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

Accumulated Misalignment :

$$E_{data}(G) = \sum_{n=1}^{N} \sum_{m=1}^{M} \min_{x \in P(D_n)} \| \mathcal{T}(v_m; G) - x \|^2$$

-> expensive

Energy Functions – Data Term

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

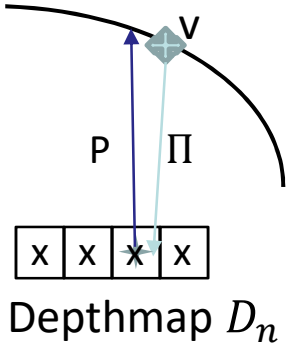
Projective point-to-plane approximation of every visible point in depthmap N

$$E_{data}(G) = \sum_{n=1}^{N} \sum_{m \in V_n(G)} \left(\mathcal{T}^{\perp}(n_m; G)^T \left(\mathcal{T}(v_m; G) - \Gamma_n(\mathcal{T}(v_m; G))\right)\right)^2$$

$\Gamma_n(v) = P_n(\Pi_n(v))$:

$\Pi_n(v)$: projection of v to a pixel in depthmap n

$P_n(p)$: projection of pixel in depthmap n into 3D



Energy Functions – Data Term

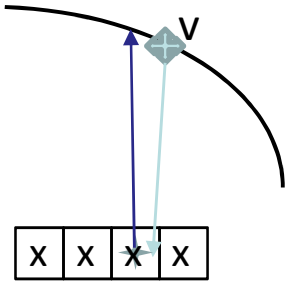
$$E(G) = \lambda_{data}\, E_{data}(G) + \lambda_{hull} E_{hull}(G) + \lambda_{corr} E_{corr}(G) + \lambda_{rot} E_{rot}(G) + \lambda_{smooth} E_{smooth}(G)$$

Projective point-to-plane approximation of every visible point in depthmap N

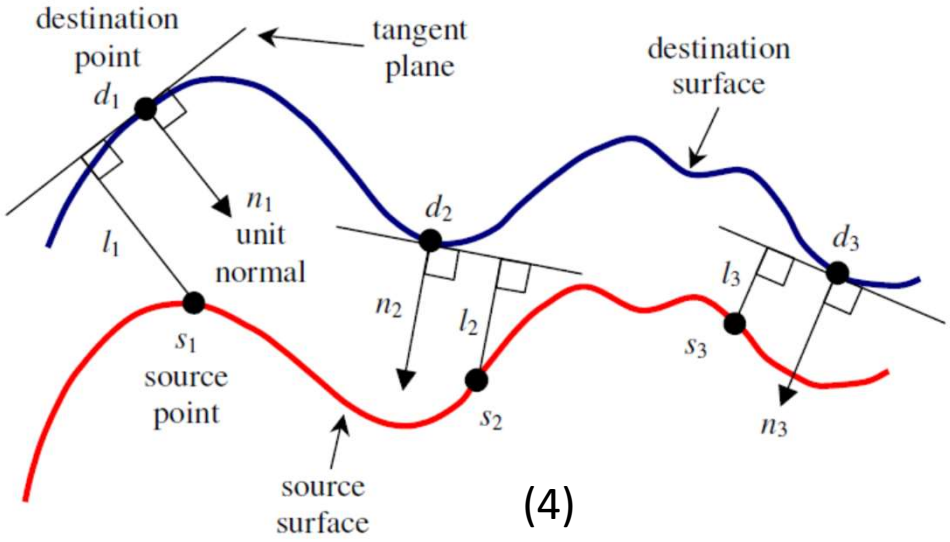
$$E_{data}(G) = \sum_{n=1}^{N} \sum_{m \in V_n(G)} \left(\mathcal{T}^{\perp}(n_m; G)^T \left(\mathcal{T}(v_m; G) - \Gamma_n(\mathcal{T}(v_m; G)) \right) \right)^2$$

Point-to-plane

$\Gamma_n(v) = P_n(\Pi_n(v))$:



Depthmap D_n



(4)

Energy Functions – Regularization Terms

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

Restriction of the class of allowed deformations:

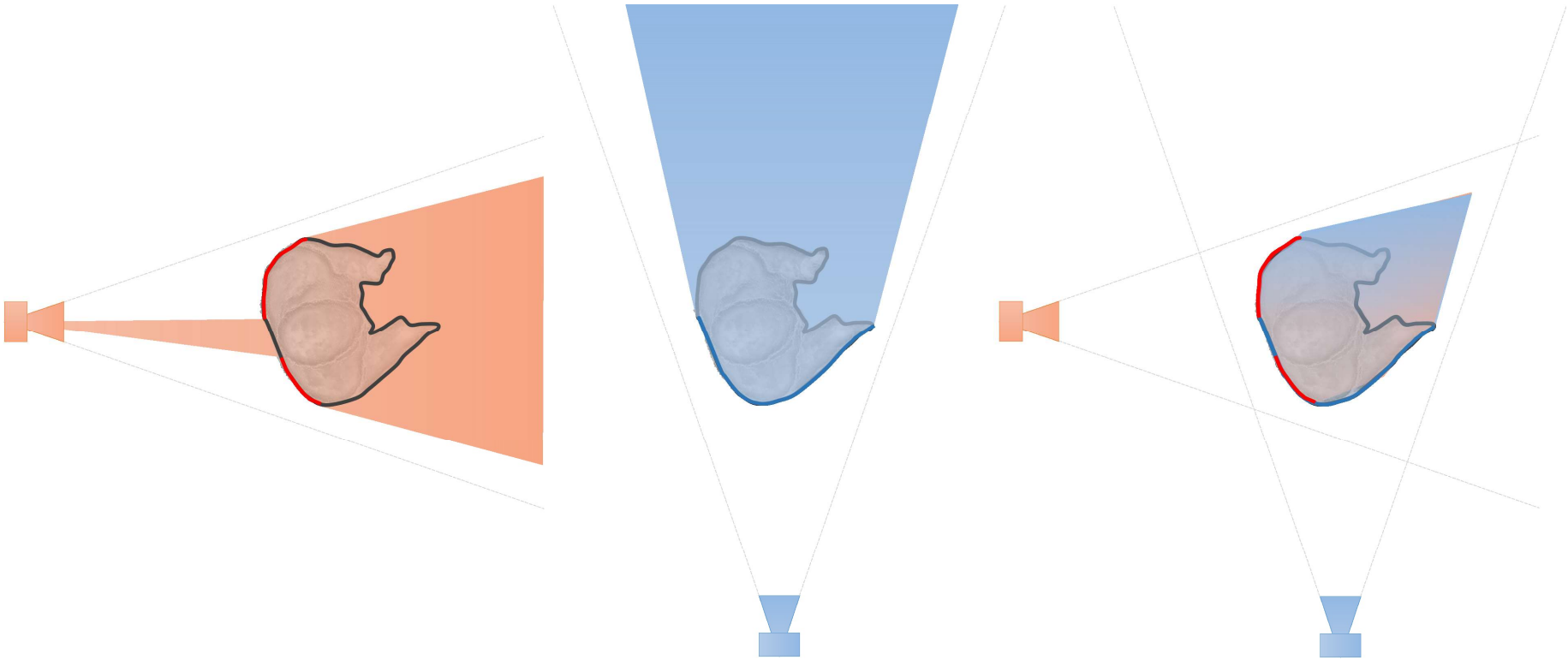
$$E_{rot}(G) = \sum_{k=1}^{K} \left\| A_k^T A_k - \mathrm{I} \right\|_F + \sum_{k=1}^{K} (\det(A_k) - 1)^2$$

Enforce similiar deformation for nearby positions:

$$E_{smooth}(G) = \sum_{k=1}^{K} \sum_{j \in N_k} w_{jk}\, \rho\left(\left\| A_j (g_k - g_j) + g_j + t_j - (g_k + t_k) \right\|^2 \right)$$

Energy Functions – Visual Hull Term

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



(3)

Energy Functions – Visual Hull Term

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

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

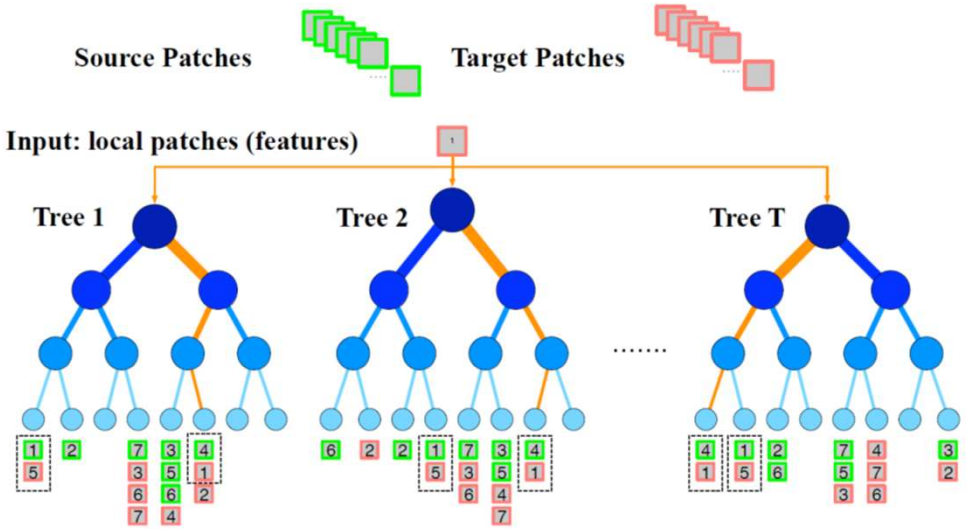
occupancy Volume: $\mathbb{H}(voxel) = \begin{cases} 1 & voxel\ inside\ of\ hull \\ 0 \end{cases}$

$\mathcal{H}(v)$: distance v to $\mathbb{H}$

-> expensive: approximation of $\mathcal{H}$ via Gaussian blur to $\mathbb{H}$

Correspondence Term – Global Patch Collider

- Decision Tree to find matches
  - 5 trees with 15 levels
- Voting Scheme:
  - Vote of all trees with a UNIQUE collision



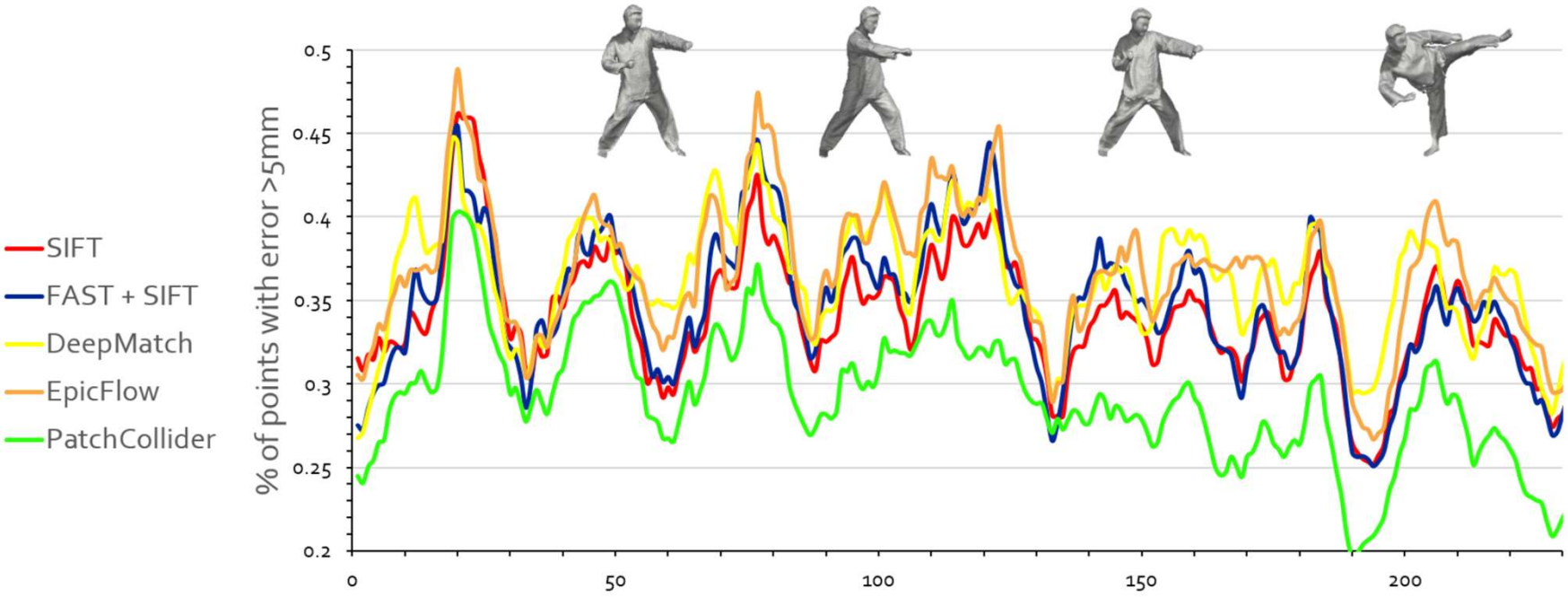
- Splitfunction (5)

$$f(p;\Theta) = \begin{cases} L, & if\ I_s(p + u/D_s(p)) - I_s(p + v/\mathrm{D}_t(p)) < \Theta \\ R, & otherwise \end{cases}$$

- learned: threshold Θ, 2D Pixel offset (u, v)

- Trained OFFLINE to maximize weighted harmonic mean between precision and recall
  - Ground truth: accurate Method of [Dou et al. 2015]

Correspondence Term - Global Patch Collider



(3)

Energy Functions – Correspondence Term

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

Set of F_n Matches $\left\{u_{nf}^{prev}, u_{nf}\right\}$

Corresponding point $q_{nf} \in \mathbb{R}^3$ per match:

$$q_{nf} = \underset{v \in V}{\operatorname{argmin}} \left\| \Pi_n(\mathcal{T}(v; G^{prev})) - u_{nf}^{prev} \right\|$$

Correspondence Term:

Deformed point 3D correspondence

$$E_{corr}(G) = \sum_{n=1}^{N} \sum_{f=1}^{F_n} \rho\left(\left\| \mathcal{T}(q_{nf}; G) - P_n\left(u_{nf}\right)\right\|^2\right)$$

robustifier

Optimization of the ED-Model

Every single Energy function was squared, $E(G) = f(X)^T f(X)$ possible

-> standard sparse linear least squares minimization problem

- Initialization:
  - ED-nodebased transformation of the old ED-Model fixed $\{A_k,\ t_k\}_{k=1}^{K}$
  - Iterative closest Points (4 Iterations) for the global translation and rotation $\{T, R\}$

Optimization of the ED-Model

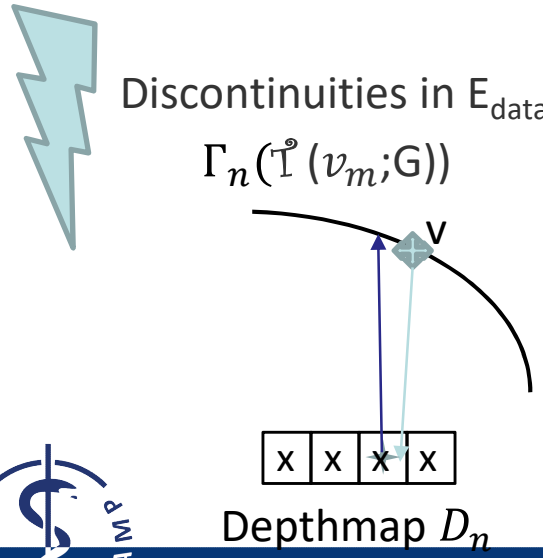
- Levenberg Marquardt Algorithm: damping factor

$$(J^T J + \mu I)h = -J^T f$$

- Stepwise update :
  - Accepted if $\mathrm{E}\big(f(X+h)\big) < E(X)$, μ lowered -> more aggressive
  - Otherwise μ increased and solved again

Discontinuities in $E_{data}(G)$:

$\Gamma_n(\mathbb{T}(v_m;G))$



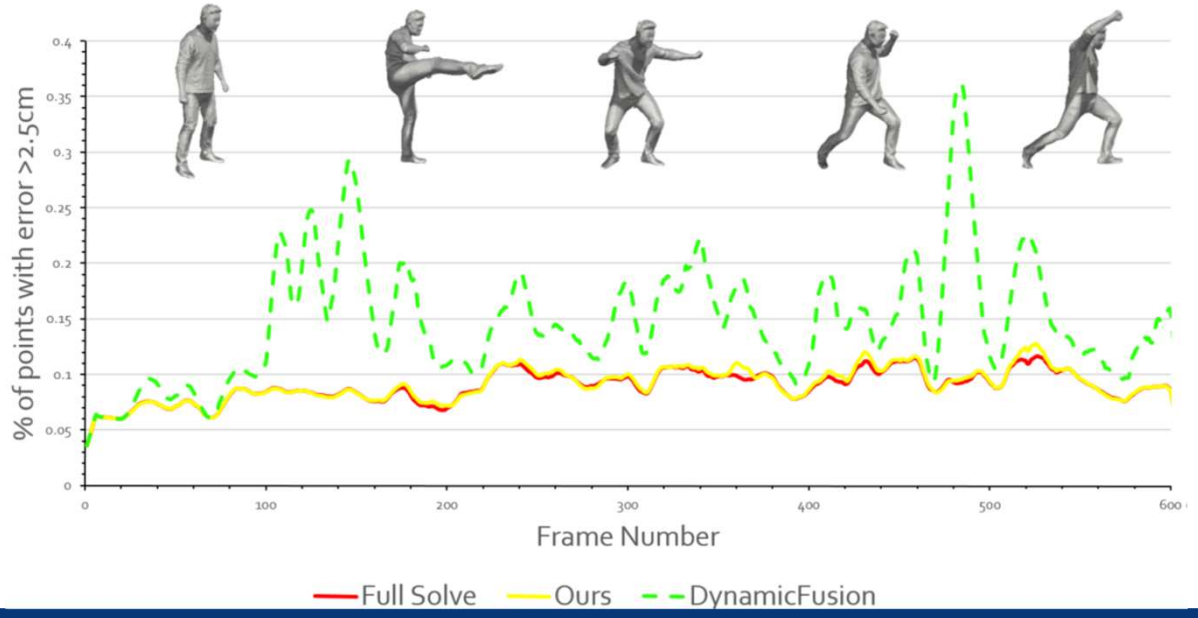
differentiable Approximation:

$$\sum_{n=1}^{N} \sum_{m \in V_n(G_0)} (\widetilde{n_m}((G_0)^T (\widetilde{v_m}((G) - \Gamma_n(\widetilde{v_m}((G_0)))^2$$

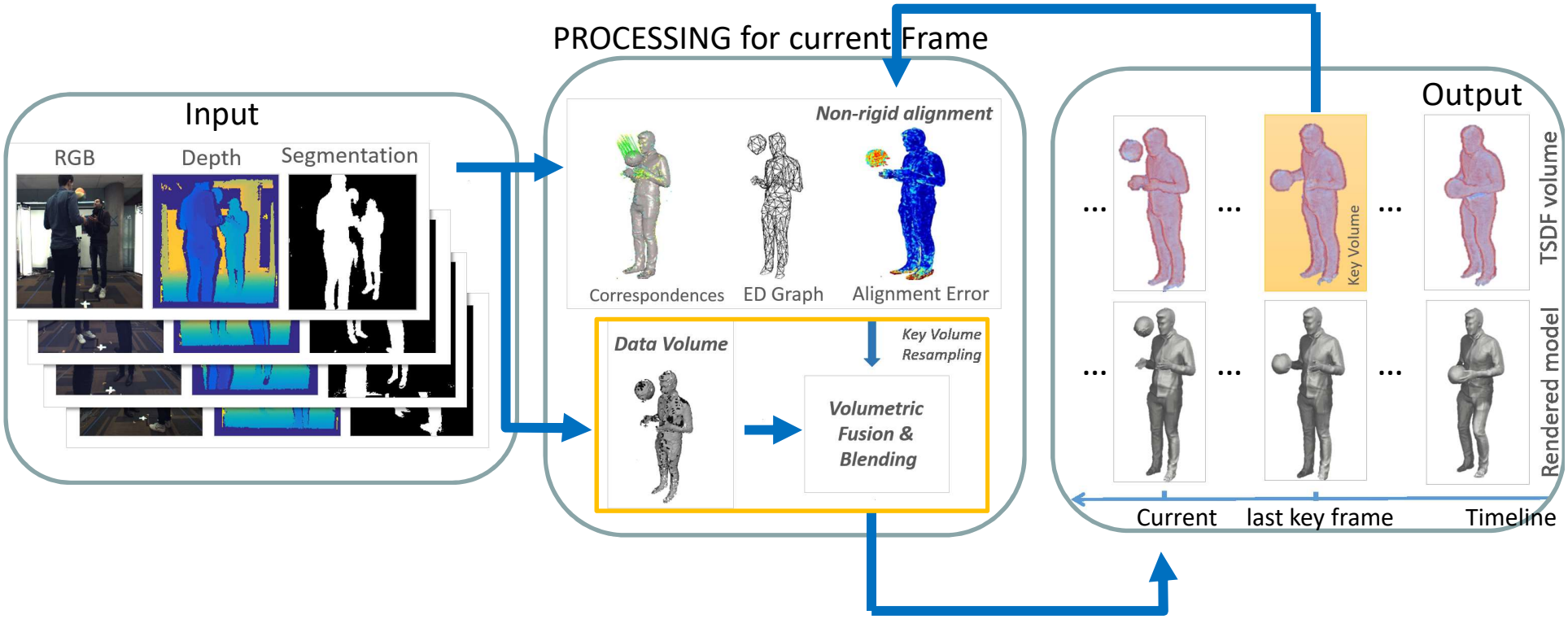
current Parameters

Linear Solving

- Store $J^T J$ and $J^T f$ instead of J, since it is magnitudes larger
  - Lower memory footprint -> time efficience
  - $E_{data}(G)$ Approximation -> independent of the number of cameras
  - $J^T J$ is a sparse matrix -> consists of nonzero blocks
    - » Blockwise calculation via CUDA blocks
- iterative solve of $J^T J$ via preconditioned conjugat gradient (PCG)
  - Preconditioner: diagonal Blocks of $J^T J$
- DynamicFusion 2015 uses direct sparse Cholesky decomposition:
  - Approximation of $J^T J$



Overview - Pipeline



(1)

Fusion

- Fusion at Data Frame
  - Warping
  - Selective Fusion
  - Blending
- Fusion at Reference Frame
- Key Volume

Volume Warping

- Warp key Volume to align the data via ED-model
  - Voxel $\in \mathbb{R}^3$ containing $\langle d, w \rangle$ in the key Volume

- Fusion into fused TSDF: x^d weighted average of warped neighboring reference-voxels $\langle \bar{d}^r, w^r \rangle$

warped Gradient of key Vol.

Correction of the depthvalue: $\bar{d}^r = \mathrm{d}^r + (\tilde{x}^r - x^d)^T \tilde{\Delta}$

Fusion at Data frame – Selective Fusion

- Voxel Collision
  - $|x - \dot{x}^R| > \eta$
  - $\dot{x}^R$ closest reference voxel

(3) reference volume, without and with voxelcolldetect.

- Voxel Misalignment
  - Discard voxel if alignment error > threshold

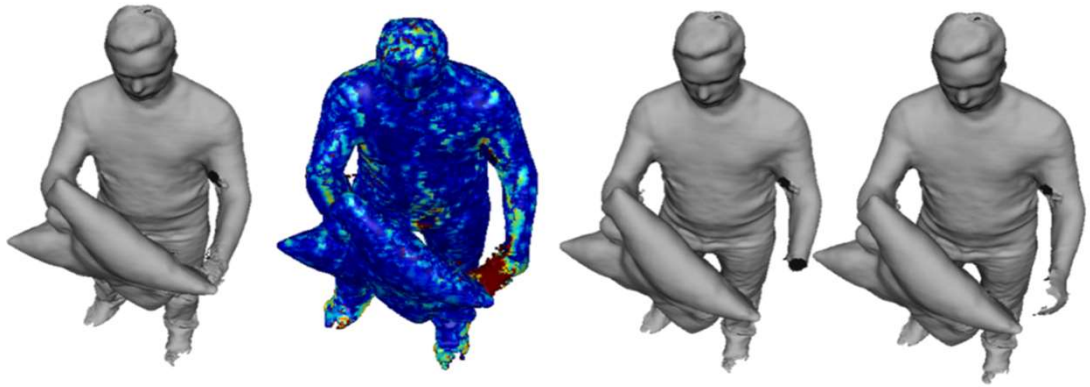
fused TSDF Hull

$$e_{\tilde{x}^r} = \begin{cases} \left|D^d(\tilde{x}^r)\right|, & if\ \mathcal{H}^d(\tilde{x}^r) = 0 \\ \min\left(\left|D^d(\tilde{x}^r)\right|, \mathcal{H}^d(\tilde{x}^r)\right), & otherwise \end{cases}$$

Fusion at Data frame - Blending

- Fuse dataVolume and warped reference Volume
  - No naive blending (boundary-voxels)

$$\bar{d}^d = \frac{\tilde{d}^r \tilde{w}^r (1.0 - e_{voxel}) + d^d w^d}{\tilde{w}^r (1.0 - e_{voxel}) + w^d}$$



e_{voxel}: aggregated average of e_{pixel} of every depthmap

$$e_{pixel} = \begin{cases} \min\left(1.0, \frac{|d - d_{proj}|}{d_{max}}\right), & if\ d_{proj}\ is\ valid \\ 1.0, & otherwise \end{cases}$$

Fusion at the Reference Frame

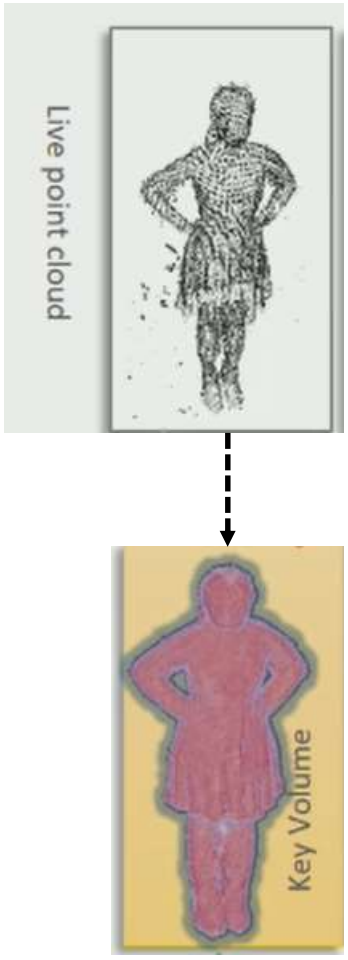
Just like DynamicFusion:

- Discarding Voxels not aligned well to the Data, and refreshing them
  - $alignmenterror\ e_{\tilde{x}^r}$
- Update depth and weight by projecting it to the depthmaps

BUT: Key Volumes

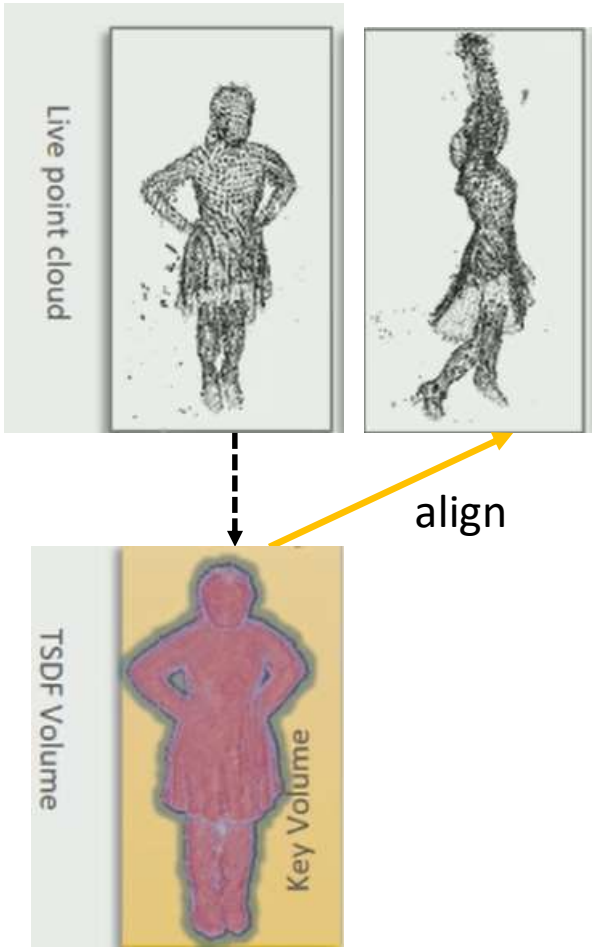
- Complete Reset periodically (here 10 frames)

Key Volumes



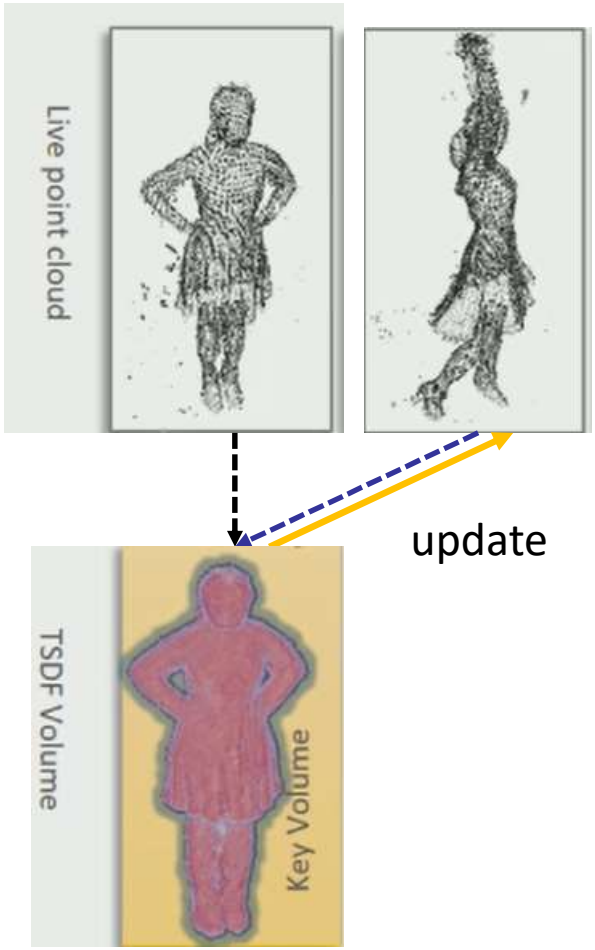
(1)

Key Volumes



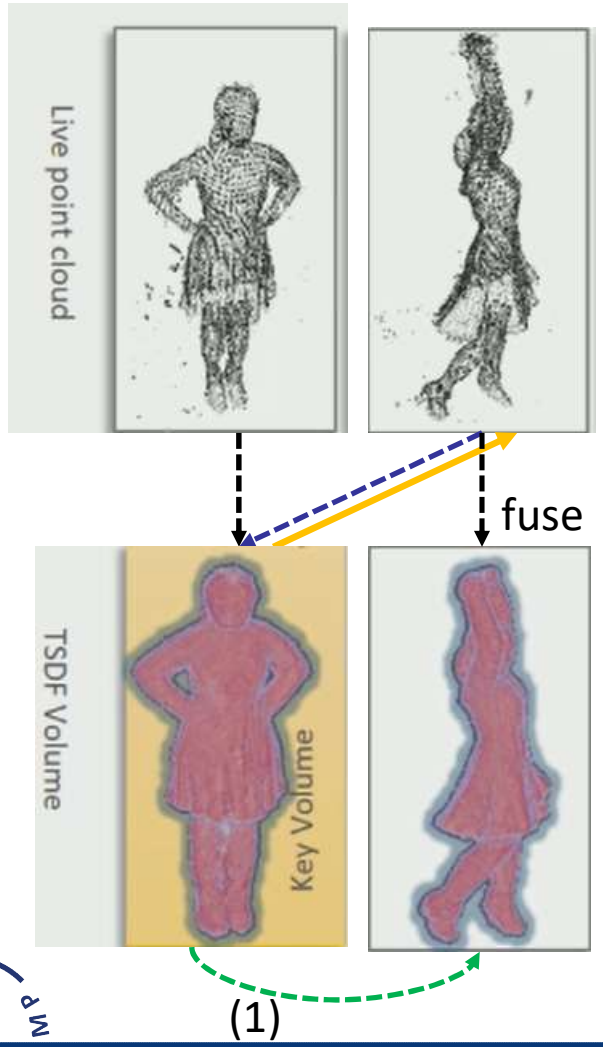
(1)

Key Volumes

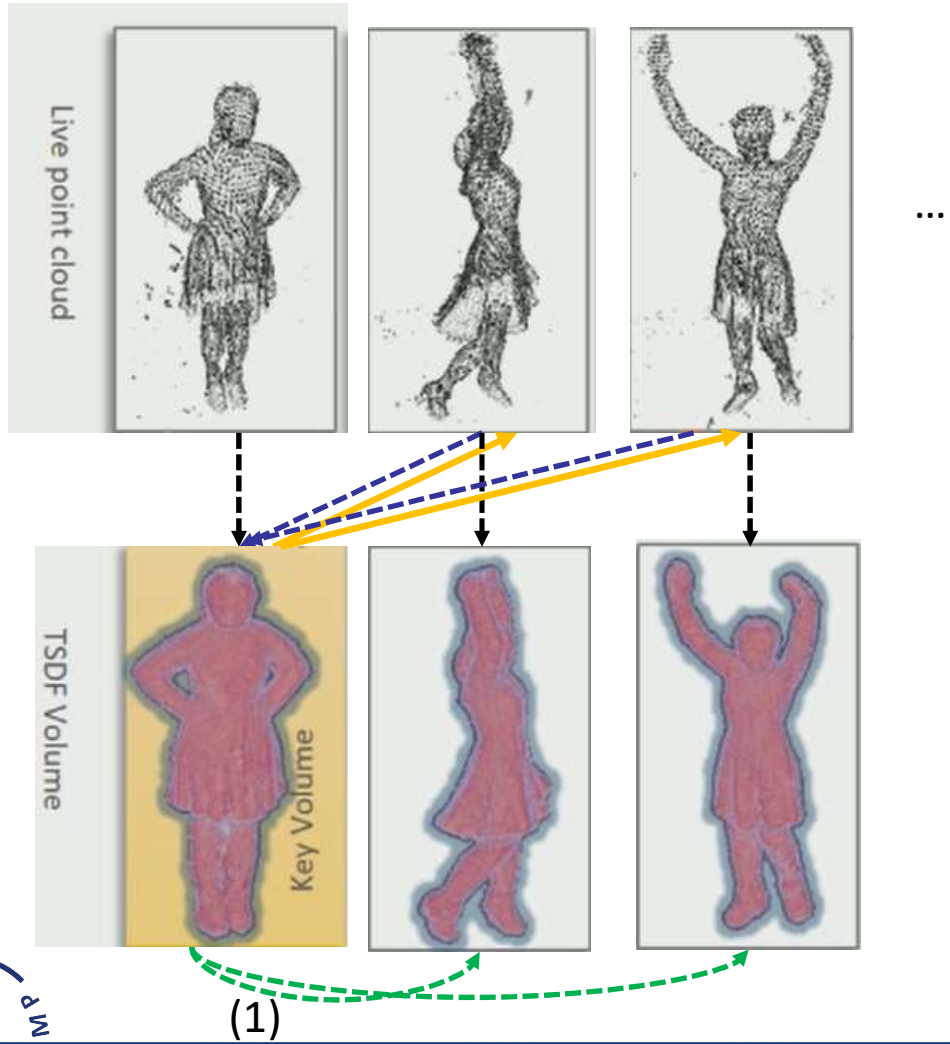


(1)

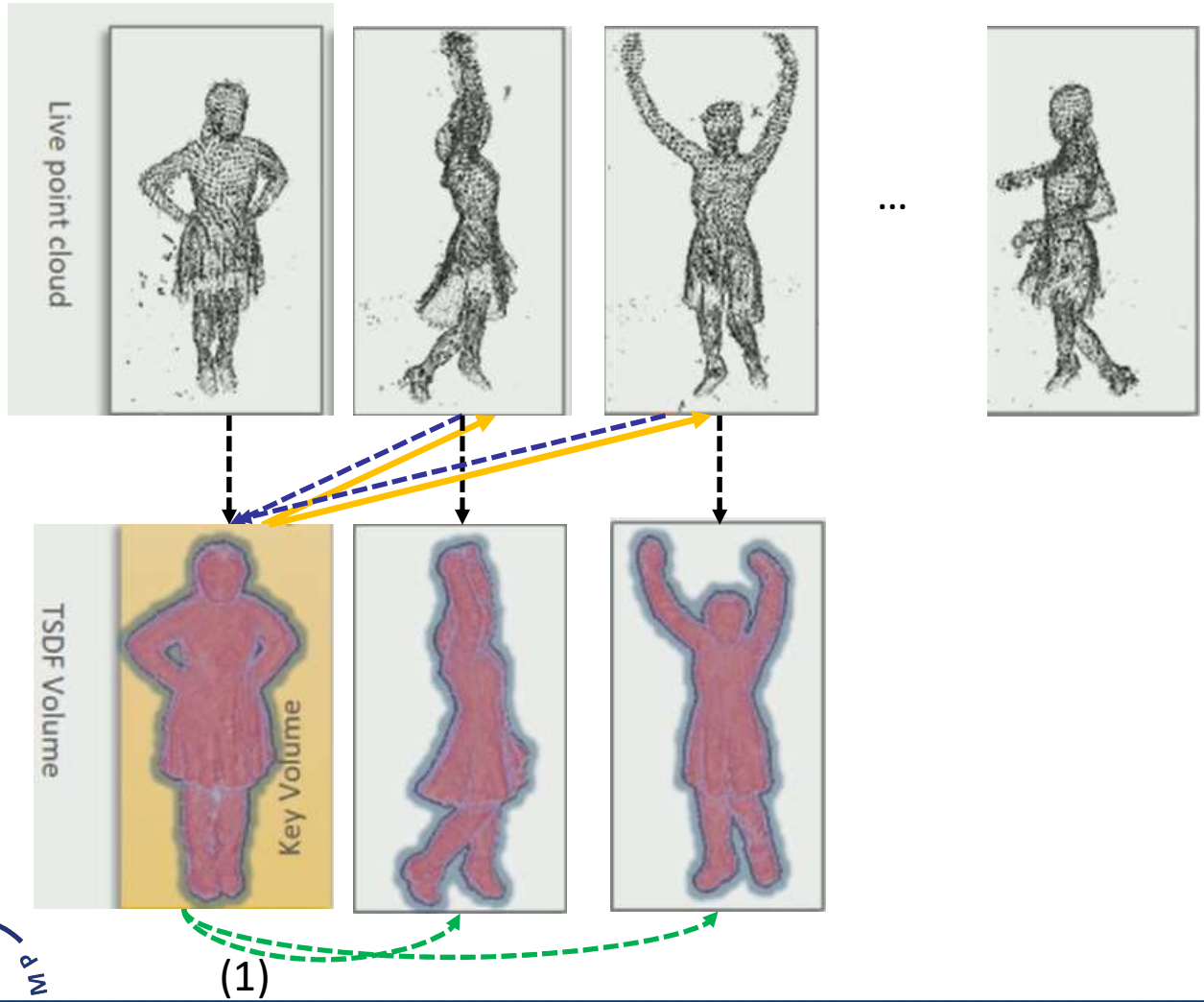
Key Volumes



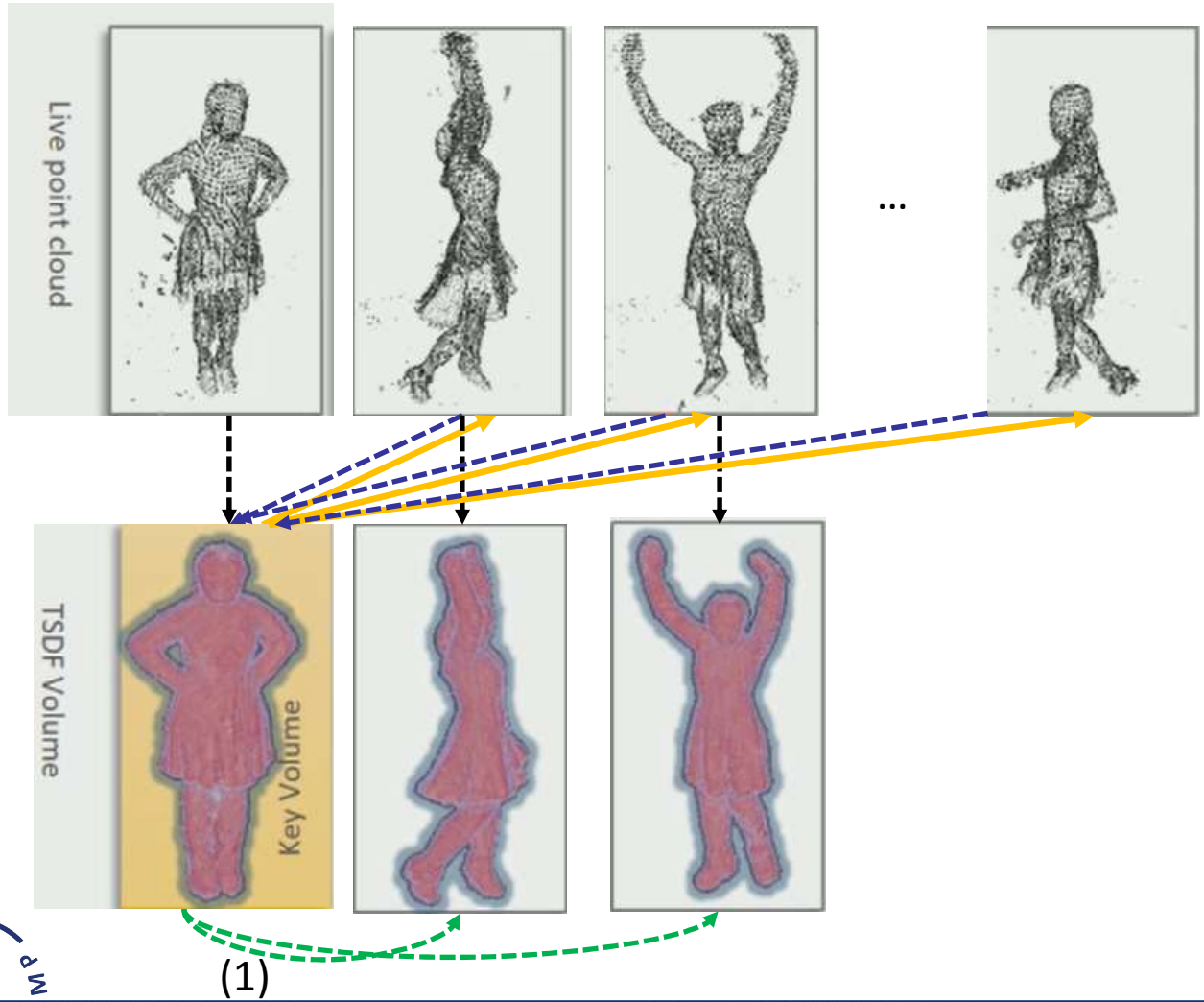
Key Volumes



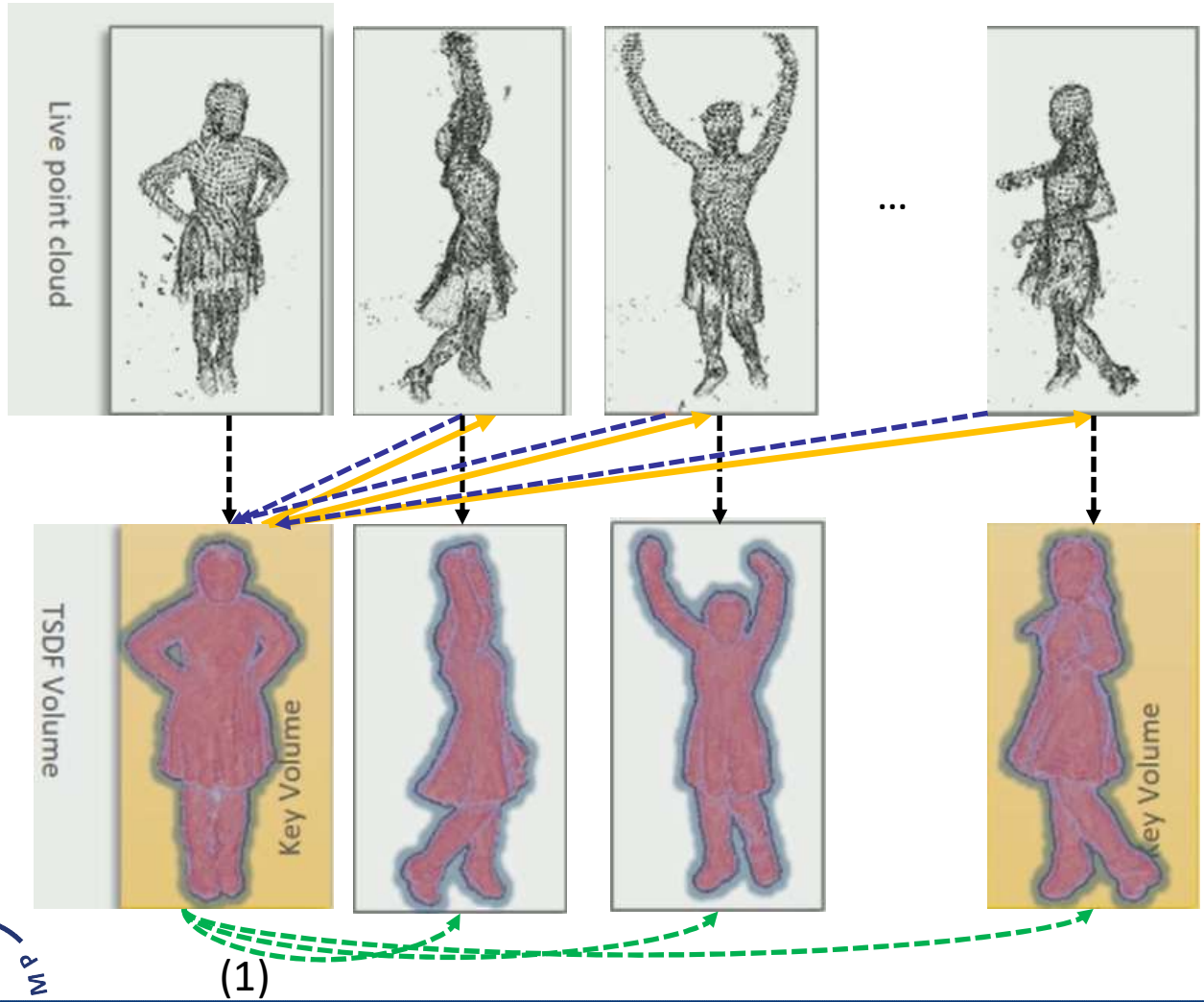
Key Volumes



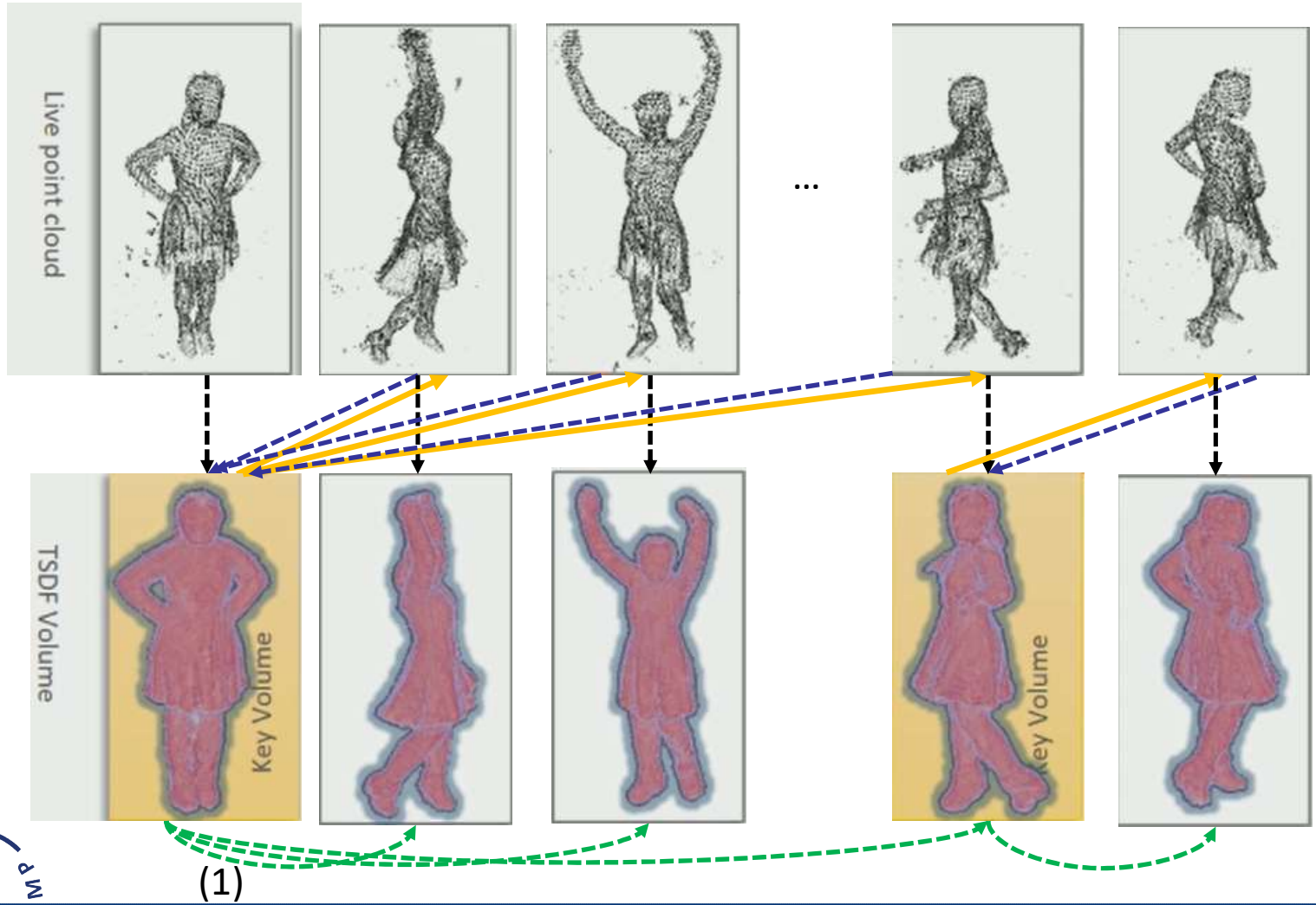
Key Volumes



Key Volumes

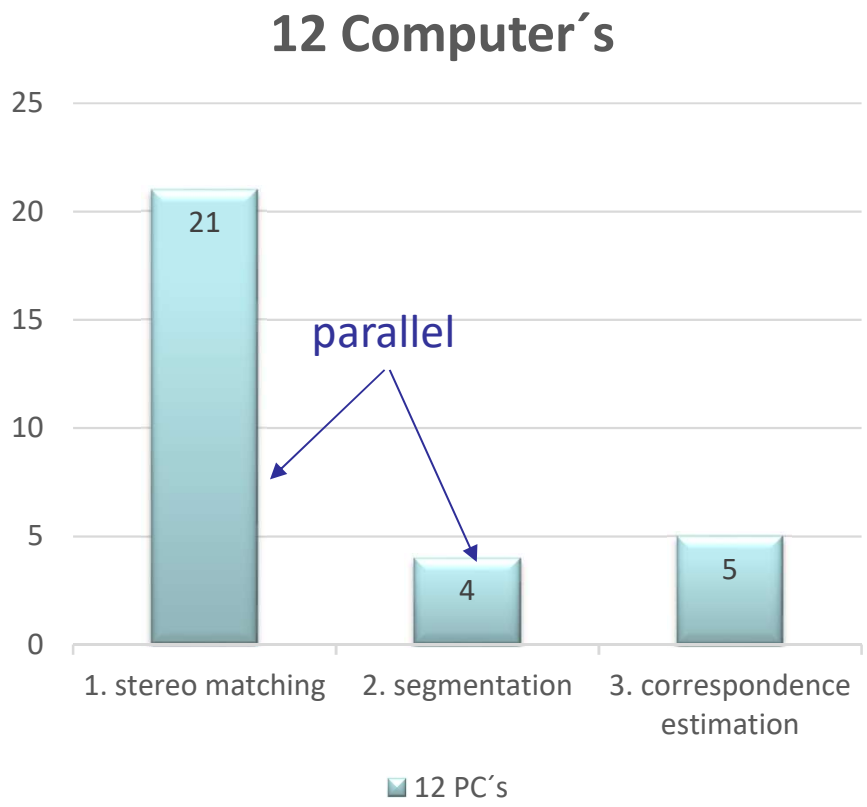


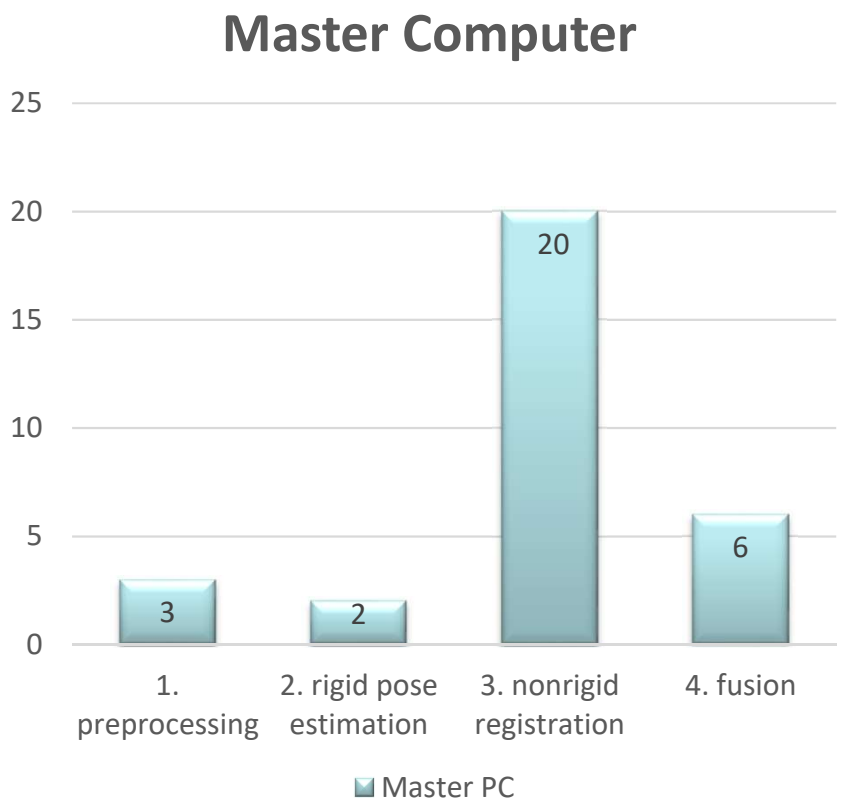
Key Volumes



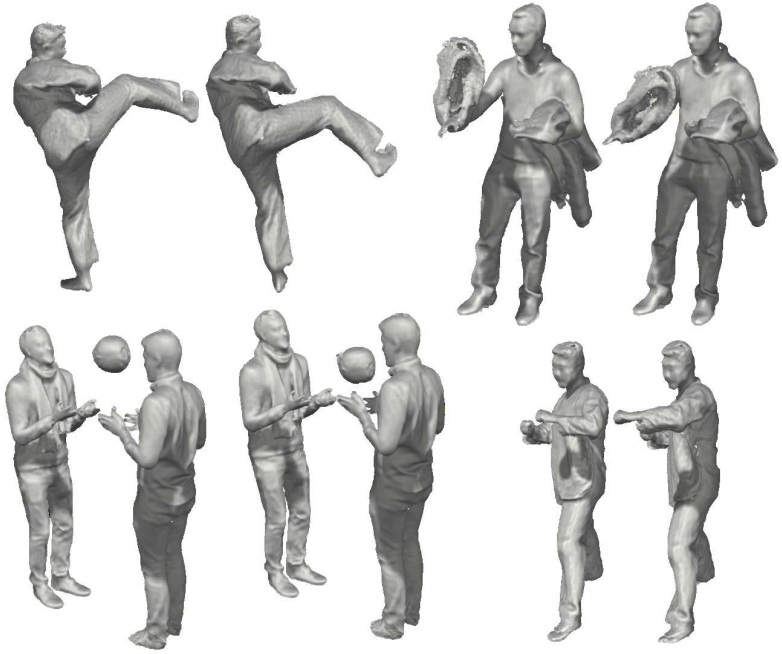
Conclusion

- Time Consumption:





Results

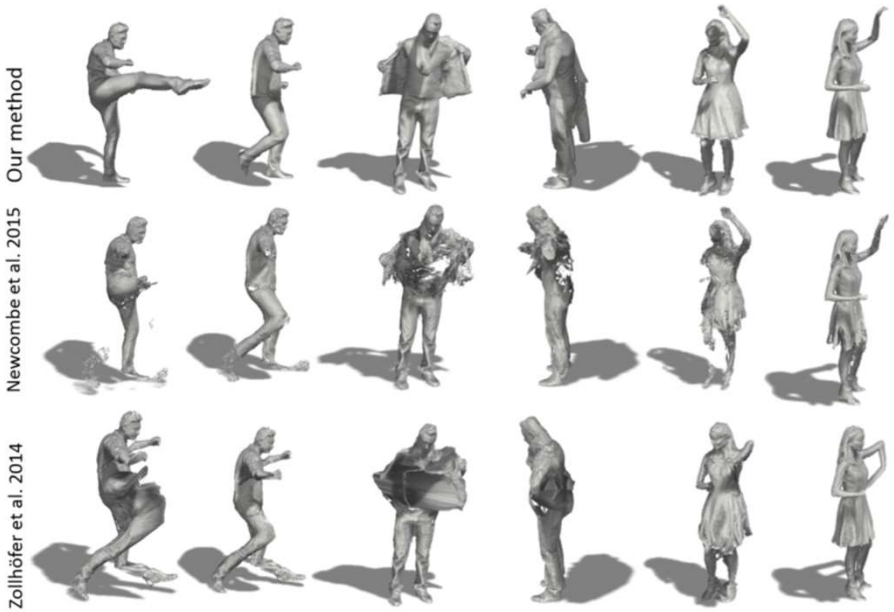


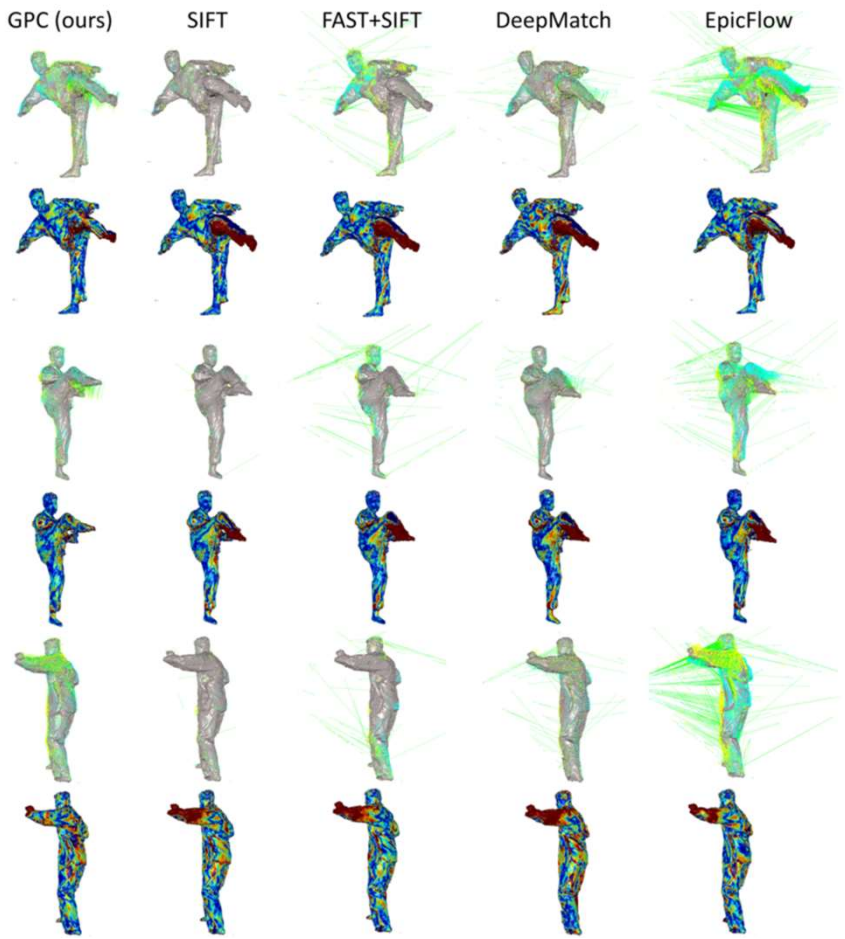
- robustness to fast motion

- complex topology changes

(3)

Comparison


Our method
Newcombe et al. 2015
Zollhöfer et al. 2014

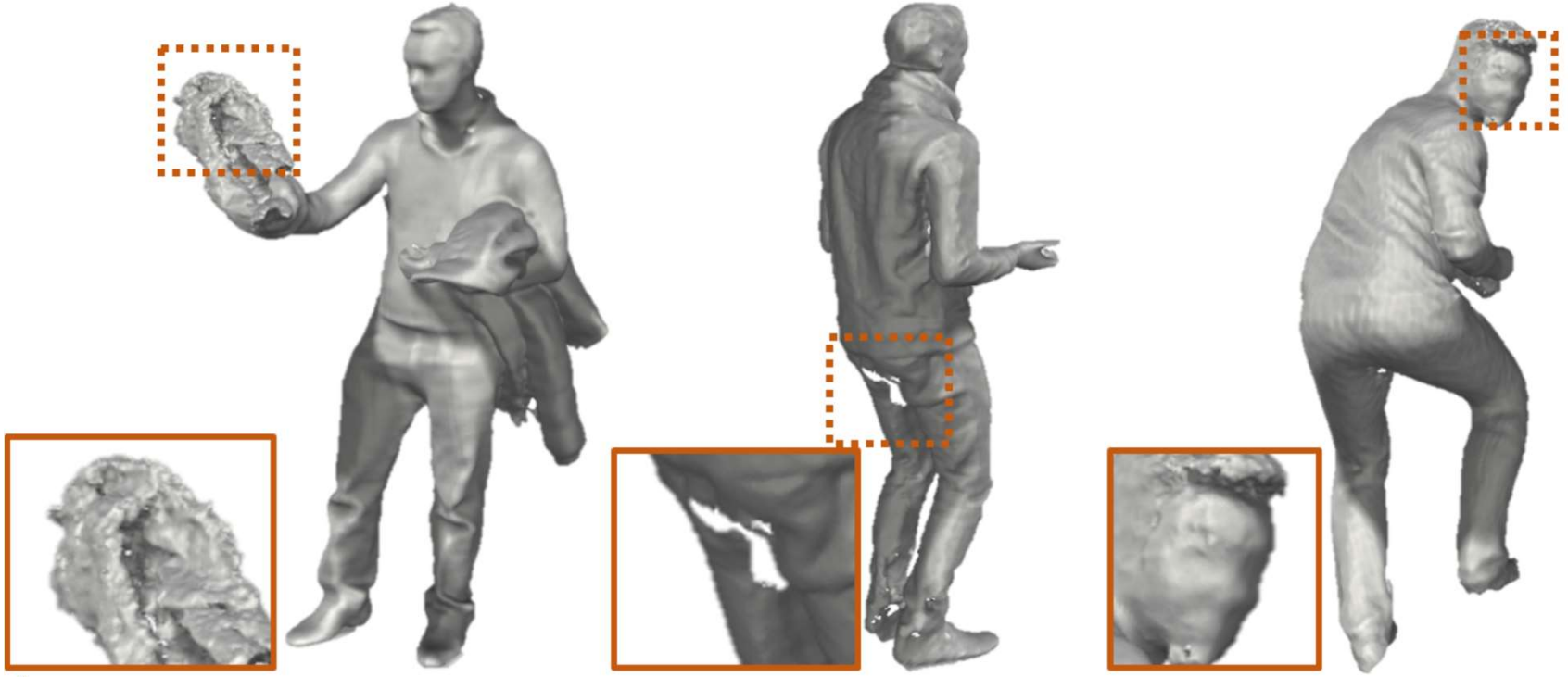

GPC (ours)
SIFT
FAST+SIFT
DeepMatch
EpicFlow

Limitations

Lost tracking

segmentation error

alignment error



(3)

Sources

- (1): http://dl.acm.org/citation.cfm?id=2925969 05.12.2016 Source Materials „Mp4“
- (2): http://dl.acm.org/citation.cfm?id=2925969 05.12.2016
  Source Materials „Supplemental files“
- (3): **Fusion4D: Real-time Performance Capture of Challenging Scenes**
  in SIGGRAPH2016
- (4): **Linear Least-Squares Optimization for Point-to-Plane ICP Surface Registration**
  - https://www-new.comp.nus.edu.sg/~lowkl/publications/lowk_point-to-plane_icp_techrep.pdf
    07.12.2016

- (5): **The Global Patch Collider** in CVPR

Questions?