Monocular Tracking and Reconstruction in Non-Rigid Environments

Kick-Off Presentation, M.Sc. Thesis

Supervisors: Federico Tombari, Ph.D; Benjamin Busam, M.Sc.

Patrick Ruhkamp

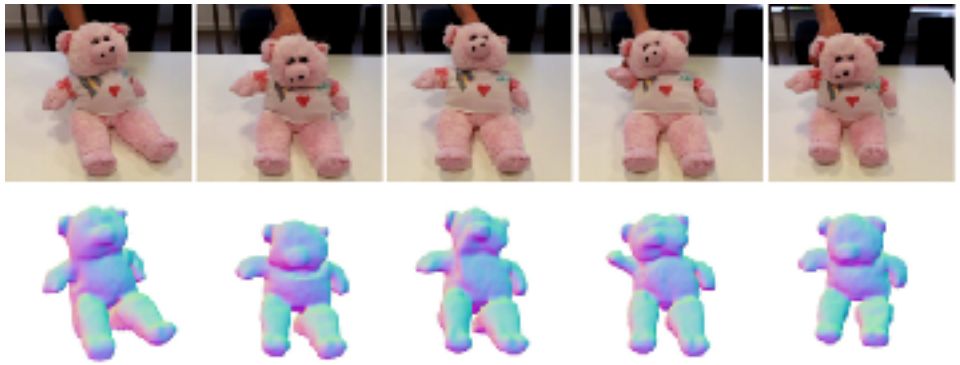
13.01.2017

Introduction

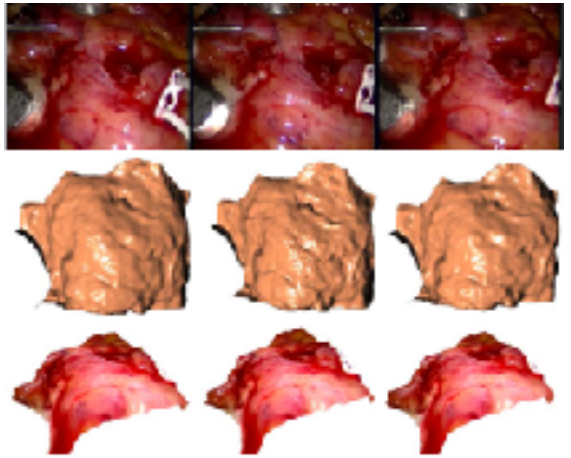
- **Motivation:** Development of a system capable of reconstructing non-rigid scenes with simple and affordable hardware
- **Goals:** Tracking of scenes and self-positioning of the system within a non-rigid scene with a monocular RGB-camera, thus providing depth information
- **Realization:** Extend proposed methods (VolumeDeform[1], DDD[3]) such that a moving monocular RGB camera can be used: ideally without any template and in real-time

Motivation

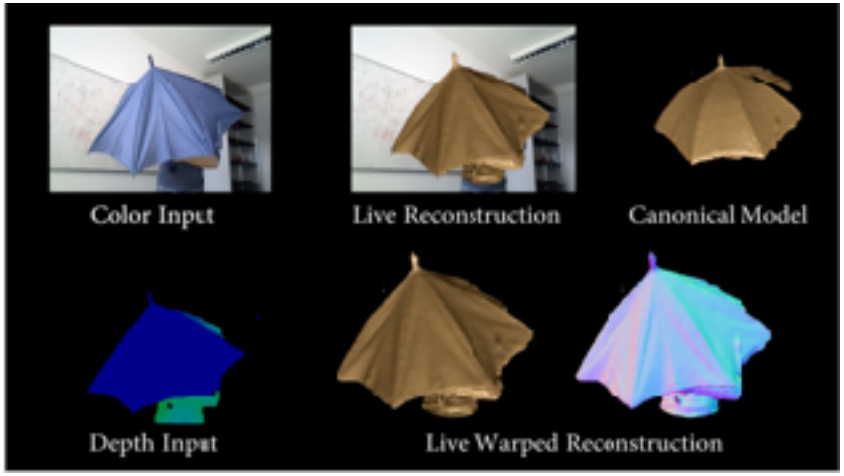
- Application Scenarios:
  - Medical Applications
  - 3D models
  - Motion Capturing
  - Augmented Reality
- Affordable Hardware
- Simple Setup



Short non-rigid reconstruction; taken from DDD[3]

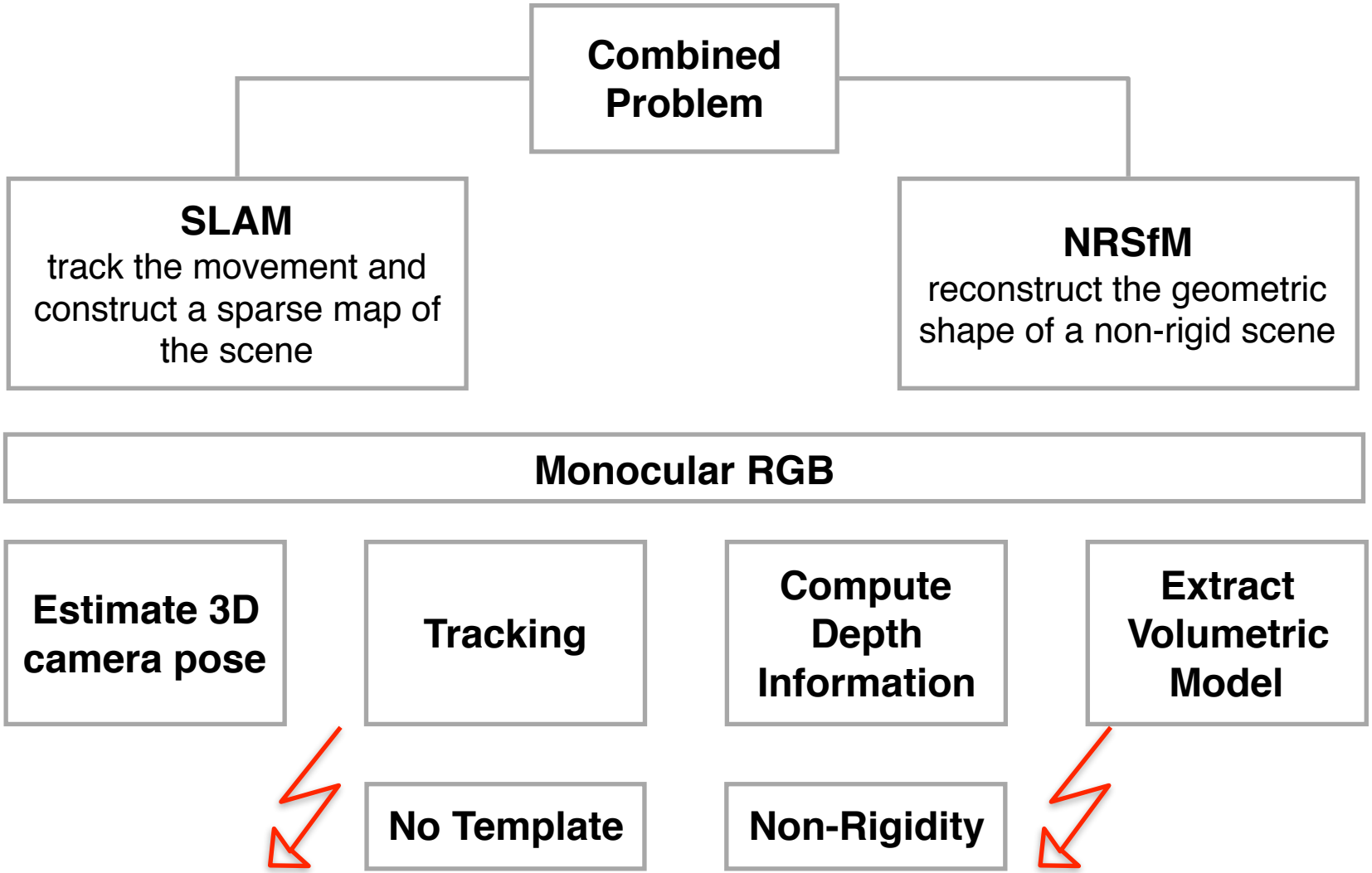


Reconstruction of a beating heart sequence; taken from Garg et al., 2013



Screenshot from supplementary Video of RGB-D method VolumeDeform[1]

Problem Formulation

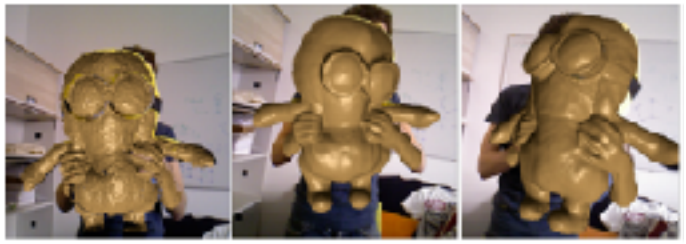


Related Work

- Existing non-rigid methods with RGB-D cameras such as **VolumeDeform**[1]
  - Dense ICP
  - Coarse Features (SIFT)
  - As Rigid As Possible (ARAP)

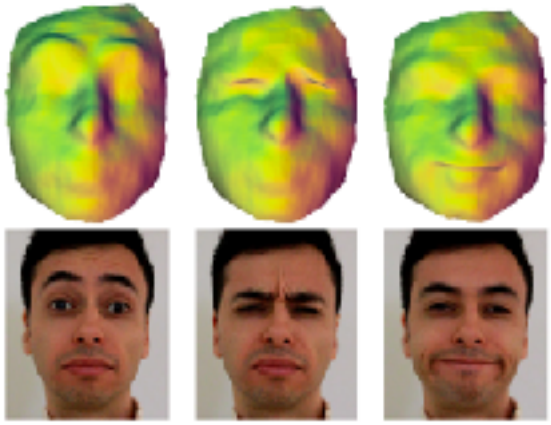
- **MonoFusion**[2]
  - Initial coarse Tracking
  - Depth information from single RGB

- **Direct, Dense and Deformable (DDD**[3]**)**
  - Template required (generated online with RGB)
  - Non-rigid scenes
  - Energy functional incorporating dense Intensity and Regularization



Real-time Reconstruction of non-rigid scene; taken from **VolumeDeform**[1]

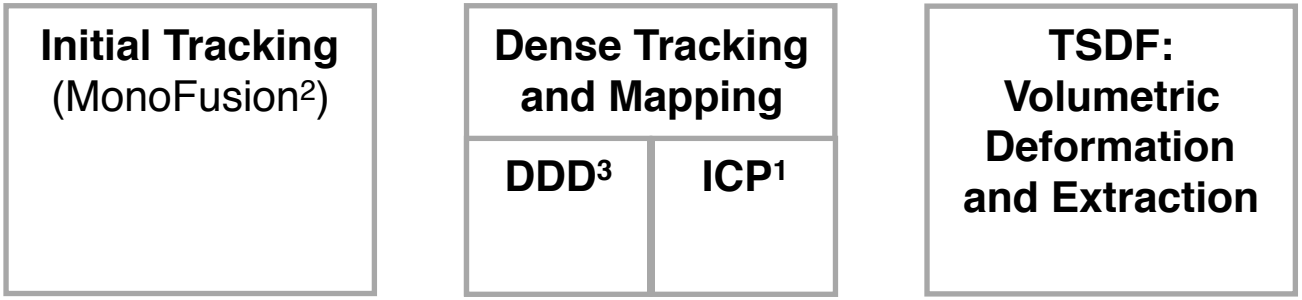
Initial Template generation; taken from **DDD**[3]

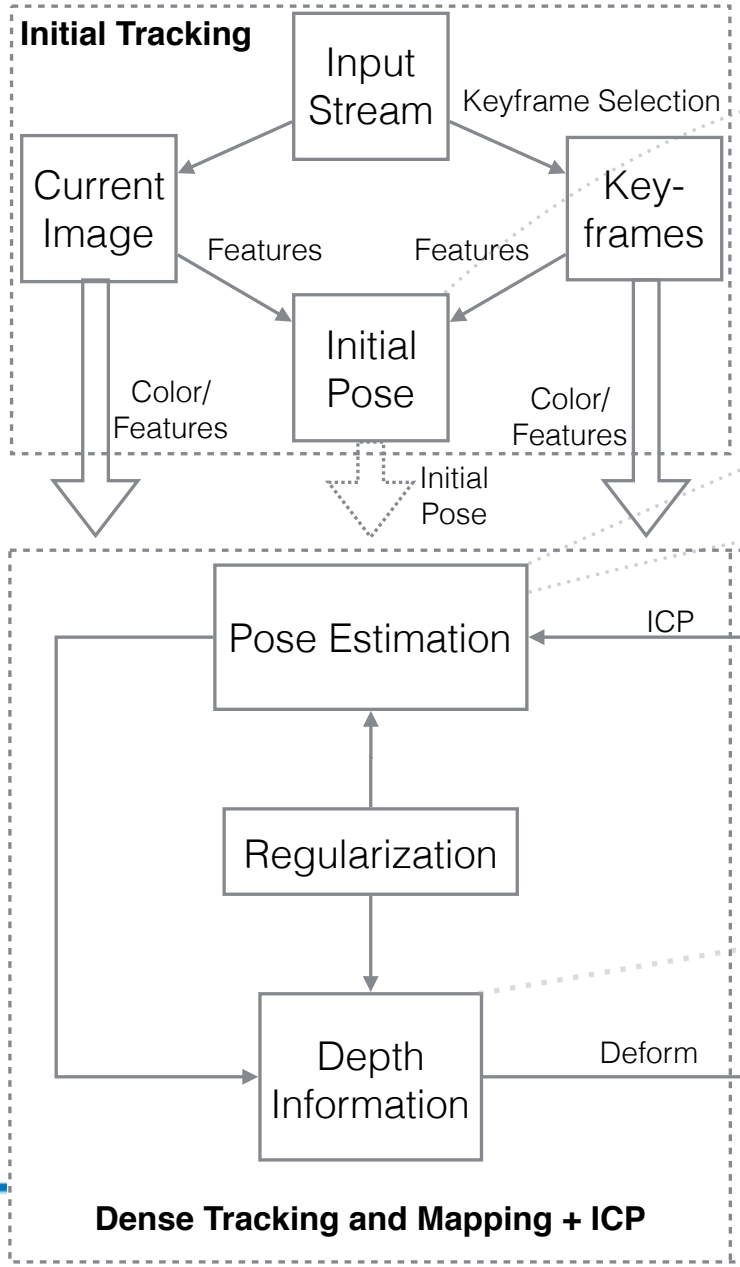


Short non rigid scene; taken from **DDD**[3]

Approaches

- Problem: Source Code for Kinect approaches not available
- Start with source code of DDD[3]
- Compute Depth Information and incorporate TSDF
- Extend Energy functional (Regularization, dense ICP)
- Ideally: no template, real time





MonoFusion:
ZNCC -> 5-point-Algo + RANSAC -> Essential Matrix -> SVD -> Initial Pose

VolumeDeform:
- Sparse Features
- Dense ICP
- ARAP

DDD:
- Dense Color
- Local Reg.
- Temp. Reg.
- ARAP

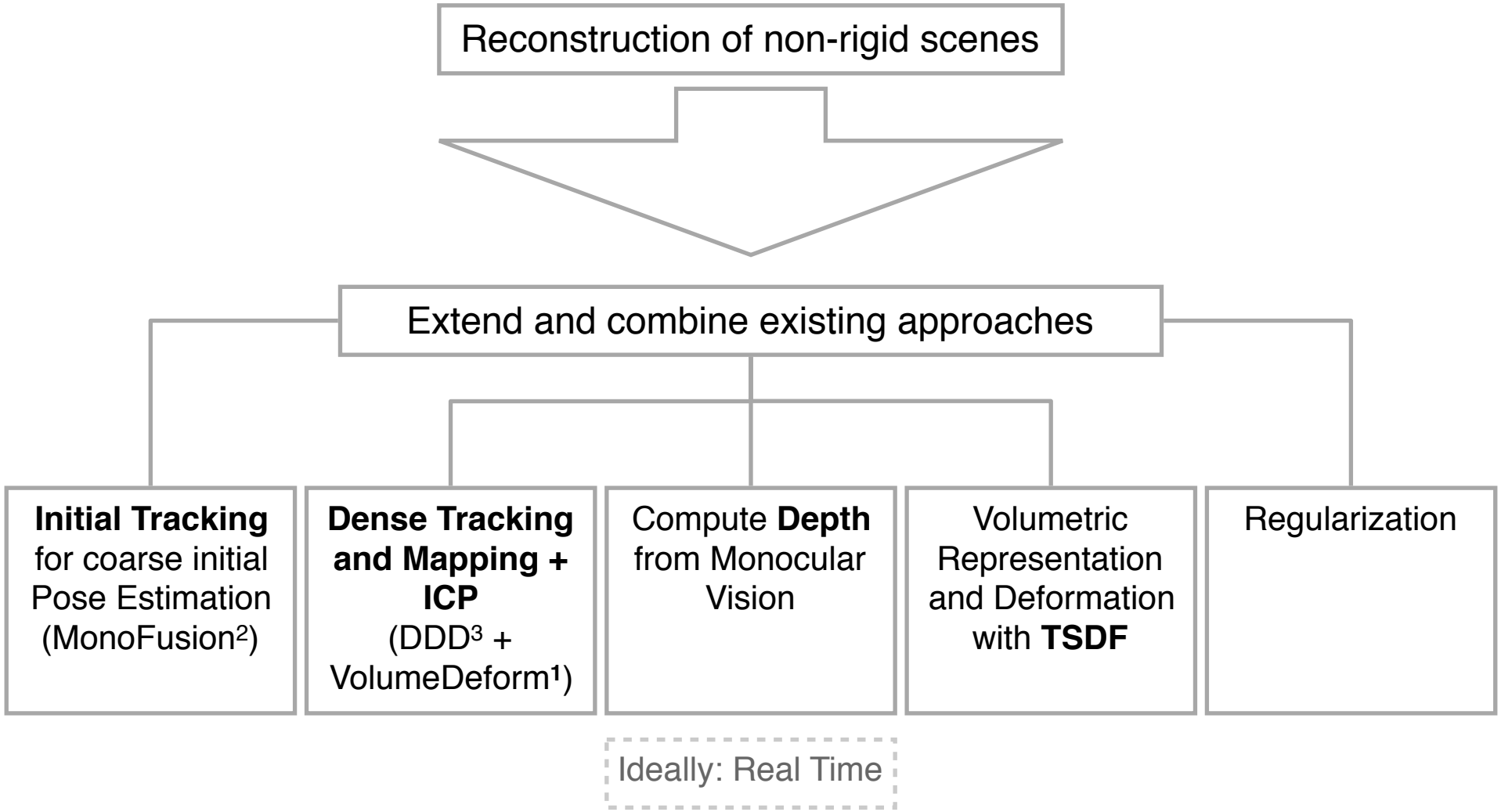
Energy Minimization

VolumeDeform:
- Volumetric Deformation

Stereo Comparison

Legend:
- Functional Entity
- Continuous Input to other Entity
- Input only for Initialzation
- Workflow
- Description

Summary


Reconstruction of non-rigid scenes
Extend and combine existing approaches
Initial Tracking for coarse initial Pose Estimation (MonoFusion[2])
Dense Tracking and Mapping + ICP (DDD[3] + VolumeDeform[1])
Compute Depth from Monocular Vision
Volumetric Representation and Deformation with TSDF
Regularization
Ideally: Real Time

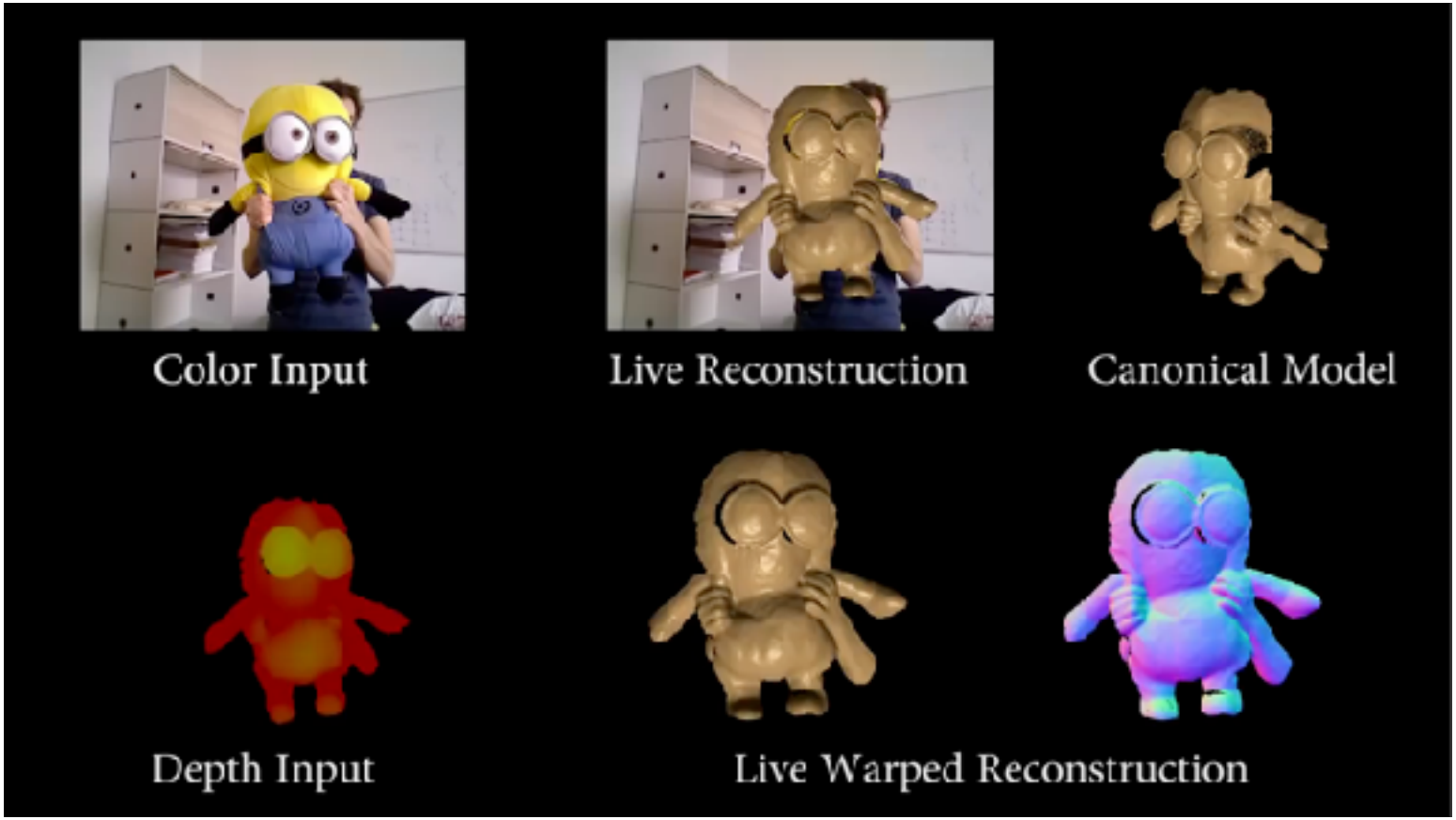
References

1. Matthias Innmann, Michael Zollhöfer, Matthias Nießner, Christian Theobalt, and Marc Stamminger. **VolumeDeform**: Real-time Volumetric Non-rigid Reconstruction. CoRR, 2016

2. Pradeep, Vivek, Rhemann, Christoph, Izadi, Shahram, Zach, Christopher, Bleyer, Michael, Bathiche, Steven. **MonoFusion**: Real-time 3D reconstruction of small scenes with a single web camera. IEEE, 2013

3. Rui Yu, Chris Russell, Neill D F Campbell, and Lourdes Agapito. **Direct, Dense, and Deformable**: Template-Based Non-rigid 3D Reconstruction from RGB Video. ICCV, 2015

Questions?


Color Input
Live Reconstruction
Canonical Model
Depth Input
Live Warped Reconstruction

Backup Slides

DDD Direct, Dense, and Deformable

- Initialization procedure required -> “template” generation
- Energy term:
  - (1) photometric error
  - (2) Total Variation term
  - (3) ARAP
  - (4) temporal smoothness

$$E(\boldsymbol{S}, \boldsymbol{R}, \boldsymbol{t}) = E_{\text{data}}(\boldsymbol{S}, \boldsymbol{R}, \boldsymbol{t}) + \lambda_r E_{\text{reg}}(\boldsymbol{S}) + \lambda_a E_{\text{arap}}(\boldsymbol{S}) + \lambda_t E_{\text{temp}}(\boldsymbol{S})$$

- Frame-to-Frame non-rigid alignment -> drift might occur
- constant brightness assumption

DDD Direct, Dense, and Deformable

$$E(\boldsymbol{S}, \boldsymbol{R}, \boldsymbol{t}) = E_{\text{data}}(\boldsymbol{S}, \boldsymbol{R}, \boldsymbol{t}) + \lambda_r E_{\text{reg}}(\boldsymbol{S}) + \lambda_a E_{\text{arap}}(\boldsymbol{S}) + \lambda_t E_{\text{temp}}(\boldsymbol{S})$$

$$E_{\text{data}}(\boldsymbol{S}, \boldsymbol{R}, \boldsymbol{t}) = \sum_{i \in \mathcal{V}} \left| \widehat{\mathbf{I}}_i - \boldsymbol{I}(\pi(\boldsymbol{R}(\boldsymbol{s}_i) + \boldsymbol{t})) \right|_c$$

$$E_{\text{reg}}(\boldsymbol{S}) = \sum_{i=1}^{N} \sum_{j \in \mathcal{N}_i} \left\| (\boldsymbol{s}_i - \boldsymbol{s}_j) - (\widehat{\mathbf{s}}_i - \widehat{\mathbf{s}}_j) \right\|_\epsilon$$

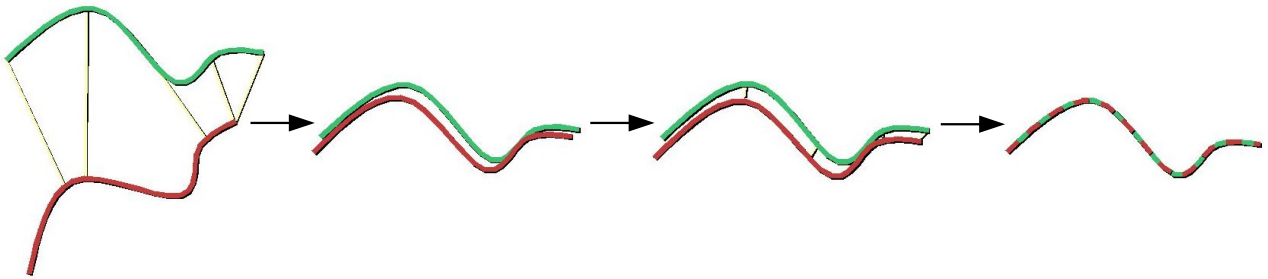
$$E_{\text{arap}}(\boldsymbol{S}, \{\boldsymbol{A}_i\}) = \sum_{i=1}^{N} \sum_{j \in \mathcal{N}_i} \left\| (\boldsymbol{s}_i - \boldsymbol{s}_j) - \boldsymbol{A}_i (\widehat{\mathbf{s}}_i - \widehat{\mathbf{s}}_j) \right\|_2^2$$

$$E_{\text{temp}}(\boldsymbol{S}, \boldsymbol{t}) = \left\| \boldsymbol{S} - \boldsymbol{S}^{t-1} \right\|_{\mathcal{F}}^2 + \left\| \boldsymbol{t} - \boldsymbol{t}^{t-1} \right\|_2^2$$

Scan alignment: Iterative Closest Points algorithm

- Align overlapping point clouds (roughly aligned)
- Compute 6DOF pose between these scans
- Use of dense depth maps: more accurate localization

- ICP algorithm: Minimize distances between two point clouds
  - Find closest pairs of points in the two point clouds
  - Minimize distances between all closest points (= align scans)
  - Compute closest points again and minimize distances
  - Repeat steps iteratively until convergence



VolumeDeform

- Scene Representation
  - shared Volumetric grid for representing TSDF of undeformed shape and space deformation field
  - Each grid point stores 6 attributes (3 for undeformed shape (signed distance, color, confidence weight))
  - Finer representation as in DynamicFusion[2] by interpolating
  - P is the current deformed surface (P=S(P'))
  - Data-parallel implementation of marching cubes
  - Space deformation S:

$$\mathcal{S}(\mathbf{x}) = \mathbf{R} \cdot \left[\sum_{i-1}^{|\mathcal{G}|} \alpha_i(\mathbf{x}) \cdot \mathbf{t}_i \right] + \mathbf{t}$$

VolumeDeform

- Space Deformation S has to be updated in real-time
- Therefore X needs to be solved

$$\mathbf{X} = (\underbrace{\cdots, \mathbf{t}_i^T, \cdots}_{3|\mathcal{G}| \text{ coordinates}} \mid \underbrace{\cdots, \mathbf{R}_i^T, \cdots}_{3|\mathcal{G}| \text{ angles}})^T$$

- Energy functional

$$E_{total}(\mathbf{X}) = \underbrace{w_s E_{sparse}(\mathbf{X}) + w_d E_{dense}(\mathbf{X})}_{data\ term} + \underbrace{w_r E_{reg}(\mathbf{X})}_{prior\ term}$$

- Solve for X

$$\mathbf{X}^* = \underset{\mathbf{X}}{\operatorname{argmin}}\, E_{total}(\mathbf{X})$$

- high-dimensional non-linear least squares problem (solved for one-ring neighbourhood M around isosurface

VolumeDeform

$$E_{total}(\mathbf{X}) = \underbrace{w_s E_{sparse}(\mathbf{X}) + w_d E_{dense}(\mathbf{X})}_{data\ \ term} + \underbrace{w_r E_{reg}(\mathbf{X})}_{prior\ \ term}$$

$$E_{dense}(\mathbf{X}) = \sum_{c=1}^{C} w_c \cdot \left[\left(\mathcal{S}(\hat{\mathbf{p}}_c) - \mathbf{p}_c^a \right)^T \cdot \mathbf{n}_c^a \right]^2$$

$$E_{sparse}(\mathbf{X}) = \sum_{s=1}^{S} \left\| \mathcal{S}(\hat{\mathbf{f}}_s) - \mathbf{f}_s \right\|_2^2$$

$$E_{reg}(\mathbf{X}) = \sum_{i \in \mathcal{M}} \sum_{j \in \mathcal{N}_i} \left\| (\mathbf{t}_i - \mathbf{t}_j) - \mathbf{R}_i (\hat{\mathbf{t}}_i - \hat{\mathbf{t}}_j) \right\|_2^2$$

Regularization term is non-linear due to rotations; measures the residual non-rigid component of the deformation which needs to be minimised