Seminar: Recent Advances in 3D Computer Vision

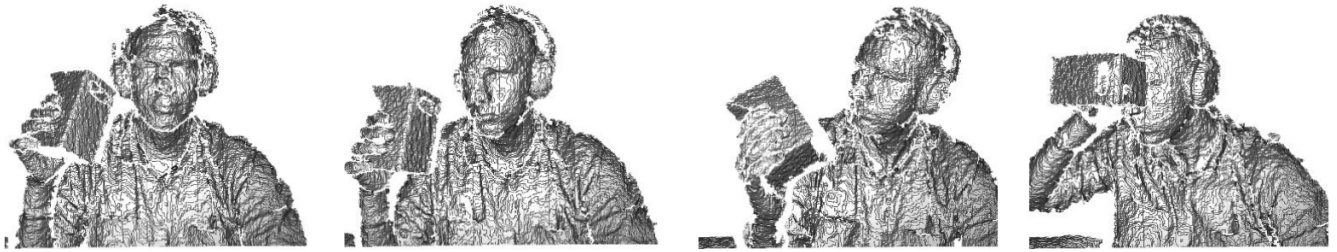
DynamicFusion: Reconstruction and Tracking of non-rigid Scenes in Real-Time

Newcombe et al. 2015, CVPR

Presented by Rebecca Richter

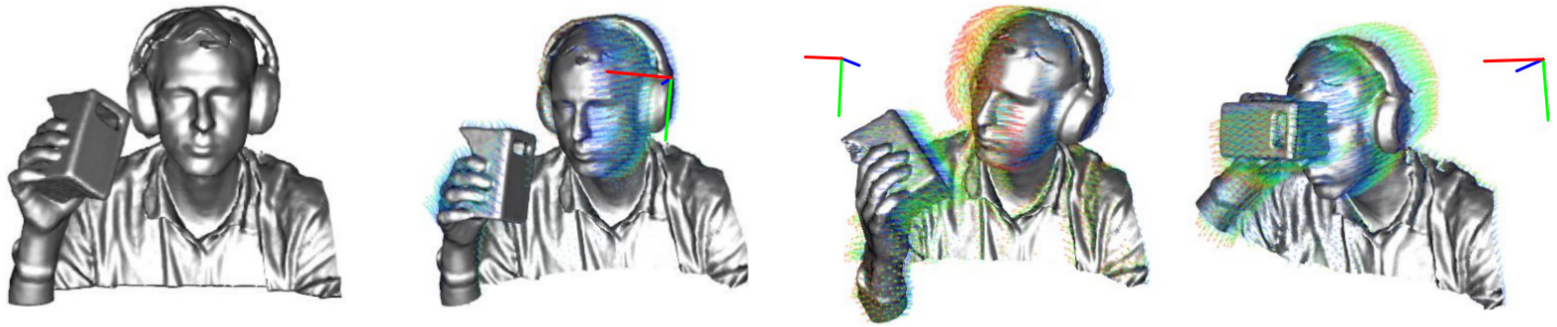
Basic Idea

Input: online stream of noisy depth maps from one RGB-D camera



t = 0s | t = 1s | t = 10s | t = 15s

Output: Real-time dense reconstruction of the moving scene



Canonical Model

→ Fuse all depth maps into one canonical frame to get a dense reconstruction
→ With every time step this reconstruction gets better

Outline

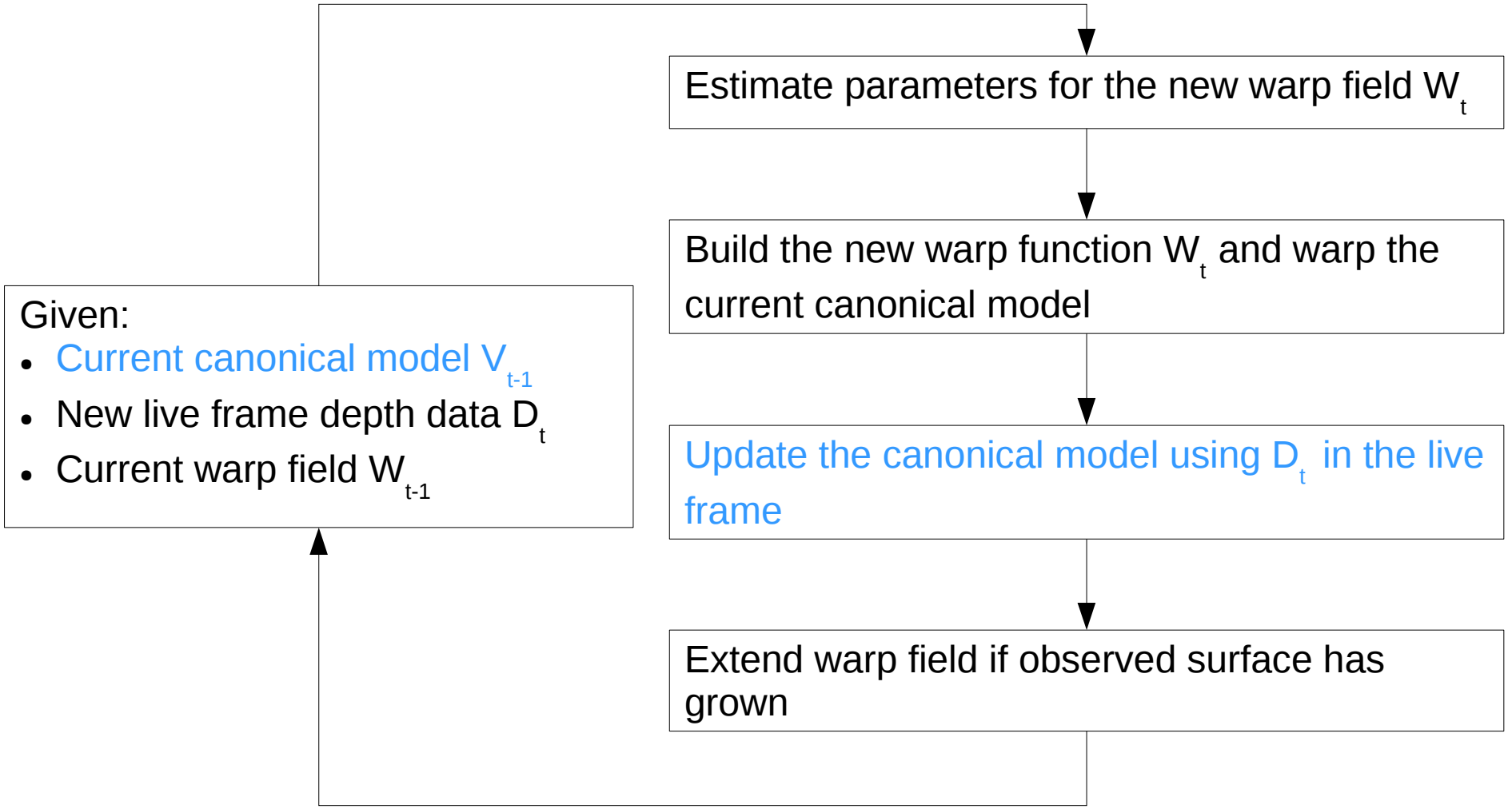
1) Method description

2) Introduction
   - Algorithm Overview
   - Canonical Model
   - Warp field
   - Estimation of warp field parameters

3) Experiments and results

4) Summary

Algorithm Overview



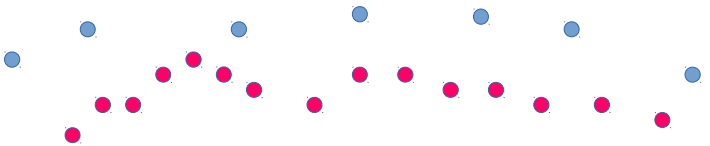
Canonical Model

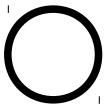
- Model safed as truncated signed distance function V: S → $\mathbb{R}^2$

$$V(x) \rightarrow [v(x) \in \mathbb{R}, w(x) \in \mathbb{R}]$$

- Updating the canonical model

Warped canonical points
Values from new depth map D_t





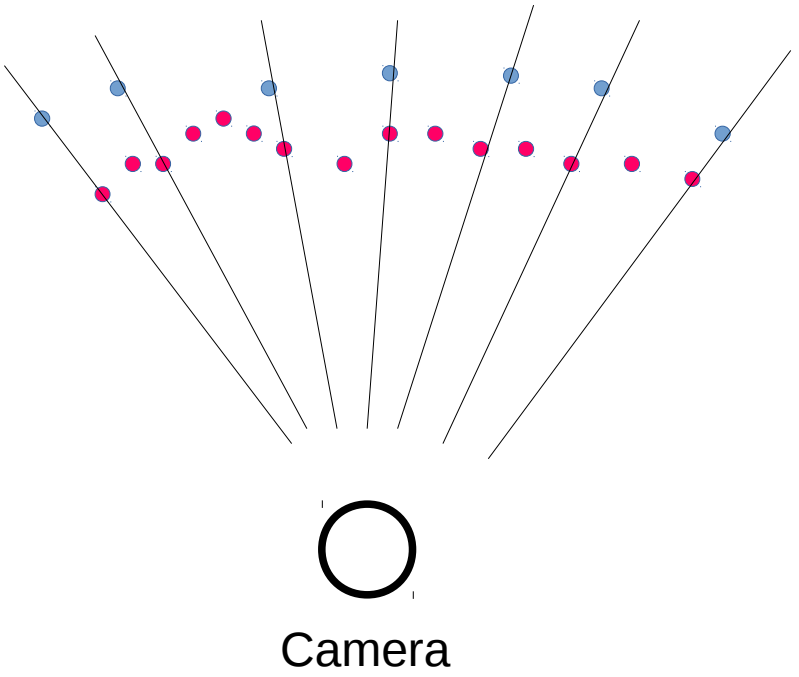
Camera

Canonical Model

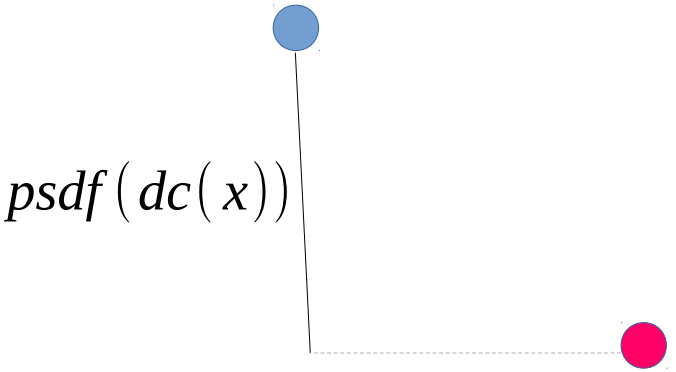
- Model saved as truncated signed distance function V: S $\rightarrow \mathbb{R}^2$

$$V(x) \rightarrow [v(x) \in \mathbb{R}, w(x) \in \mathbb{R}]$$

- Updating the canonical model



Warped canonical points
Values from new depth map D_t



Canonical Model

- Model saved as truncated signed distance function V: S $\rightarrow \mathbb{R}^2$

$$V(x) \rightarrow [v(x) \in \mathbb{R}, w(x) \in \mathbb{R}]$$

- Updating the canonical model

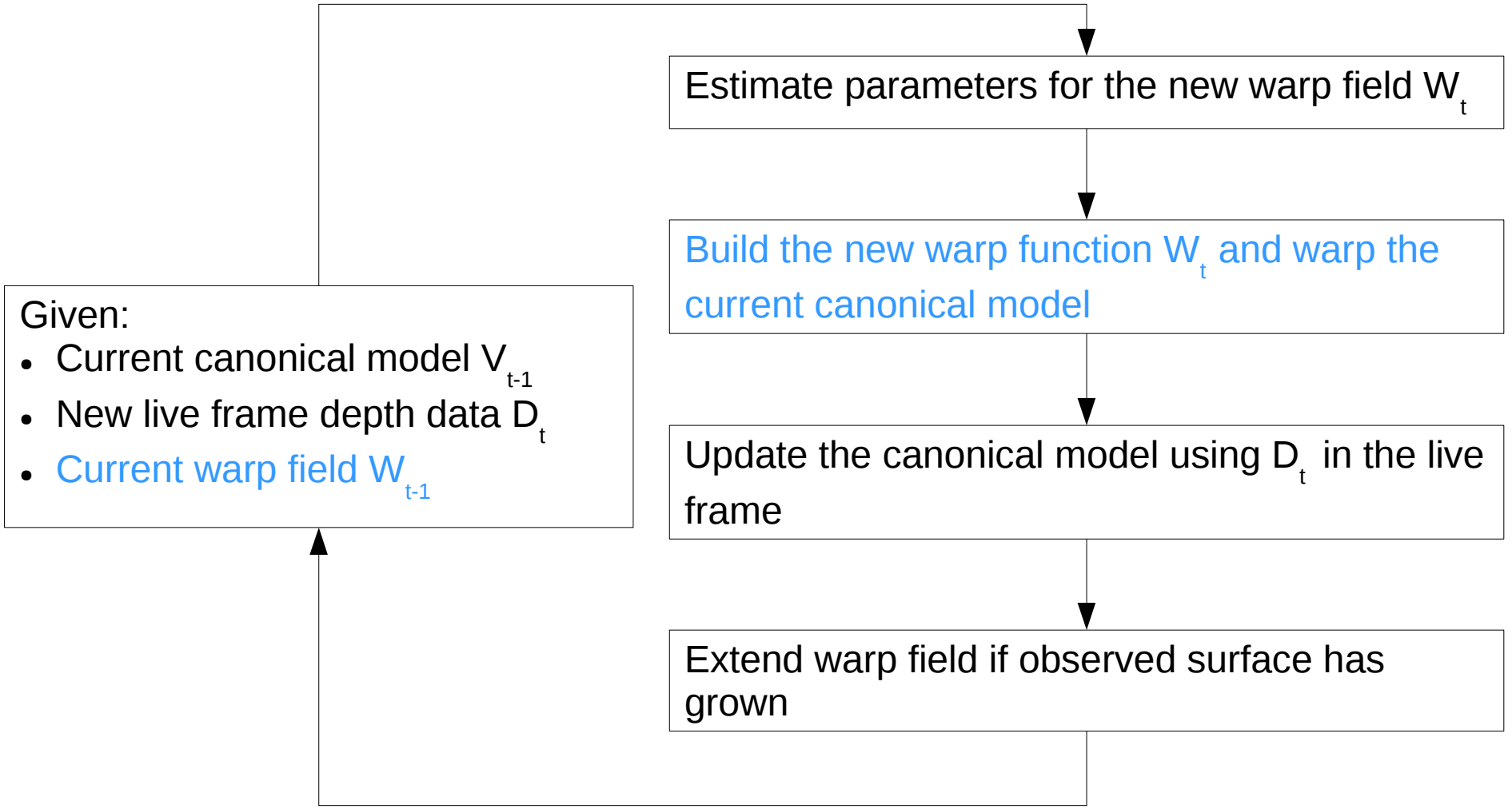
$$\mathcal{V}(\mathbf{x})_t = \begin{cases} [\mathrm{v}'(\mathbf{x}), \mathrm{w}'(\mathbf{x})]^\top, & \text{if } \mathbf{psdf}(\mathbf{dc}(\mathbf{x})) > -\tau \\ \mathcal{V}(\mathbf{x})_{t-1}, & \text{otherwise} \end{cases}$$

$$v'(x) = \frac{v(x)_{(t-1)} w(x)_{(t-1)} + min(psdf(dc(x)), \tau)}{w(x)_{(t-1)} + w(x)}$$

$$w'(x) = min(w(x)_{(t-1)} + w(x), w_{(max)})$$

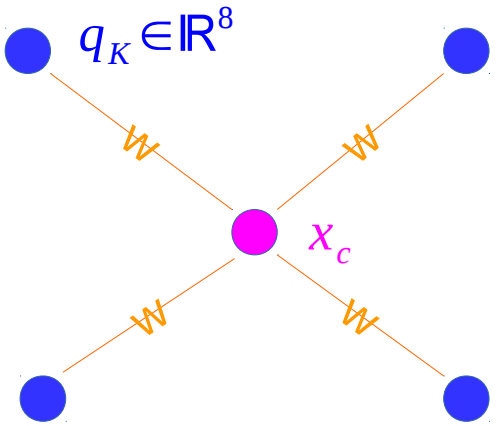
→ use weighted average of all measured depths for this point
→ w(x) represents probability that the associated depth value is really observed at that point

Algorithm Overview



Warp Field

- Estimated parameters for rigid transformations of all deformation nodes
- Define dense non-rigid volumetric warp field for all other voxels through interpolation



Sparse set of k-nearest deformation nodes (N(x)) with already computed rigid local transformations

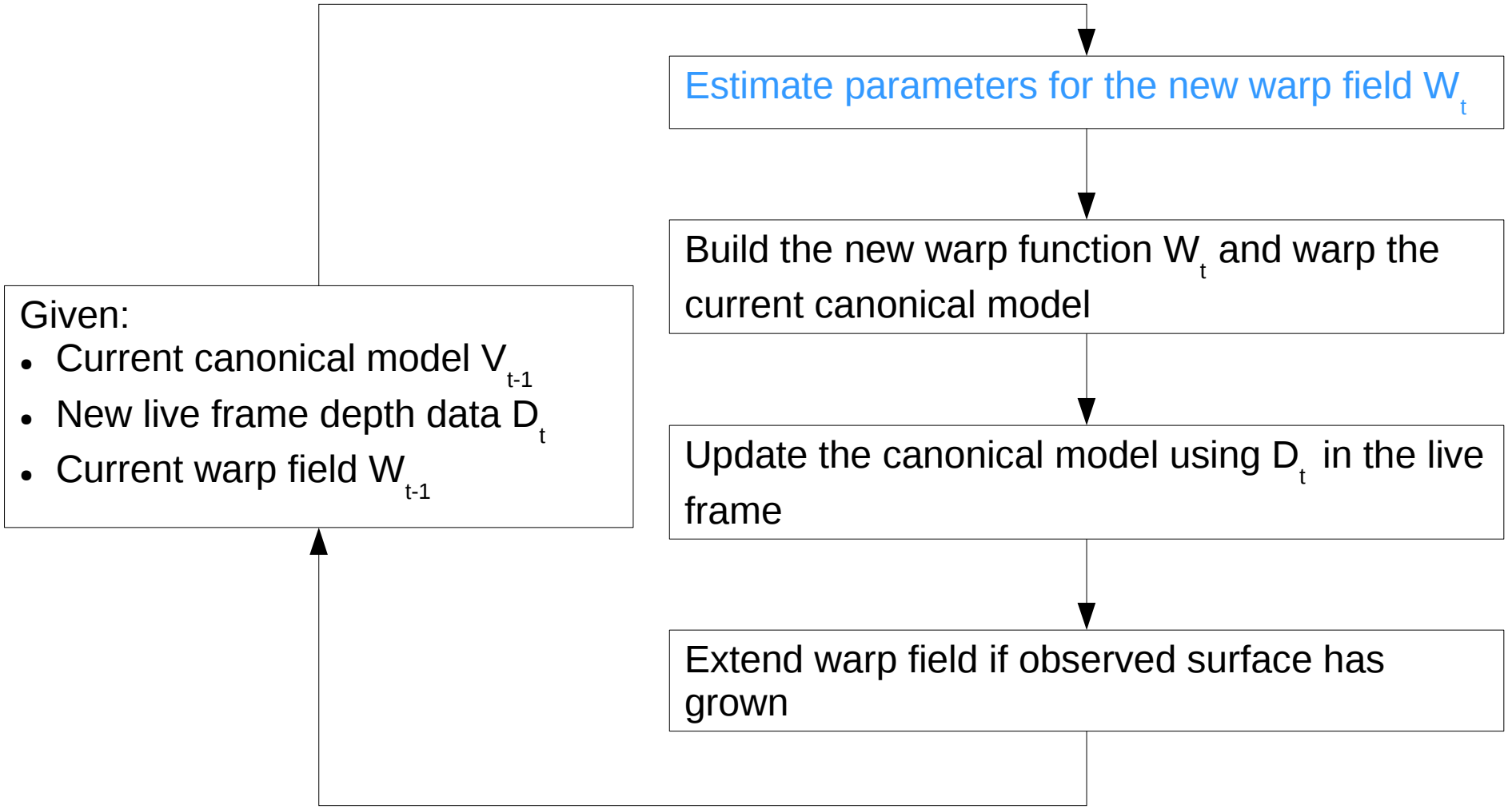
Voxel center from canonical frame where we want to estimate the transformation

Weight value computed from position distance and radial basis weight of deformation node

- Dense non-rigid Warp Field:

$$W_t(x_c) = T_{global} * SE3\left(\frac{\sum_{k \in N(x_c)} w_k(x_c) q_k}{\| \sum_{k \in N(x_c)} w_k(x_c) q_k \|}\right)$$

Algorithm Overview



Estimate Warp Field parameters

- Given: new live depth values, current canonical model, edge set ϵ
- Looking for: transformation of the deformation nodes

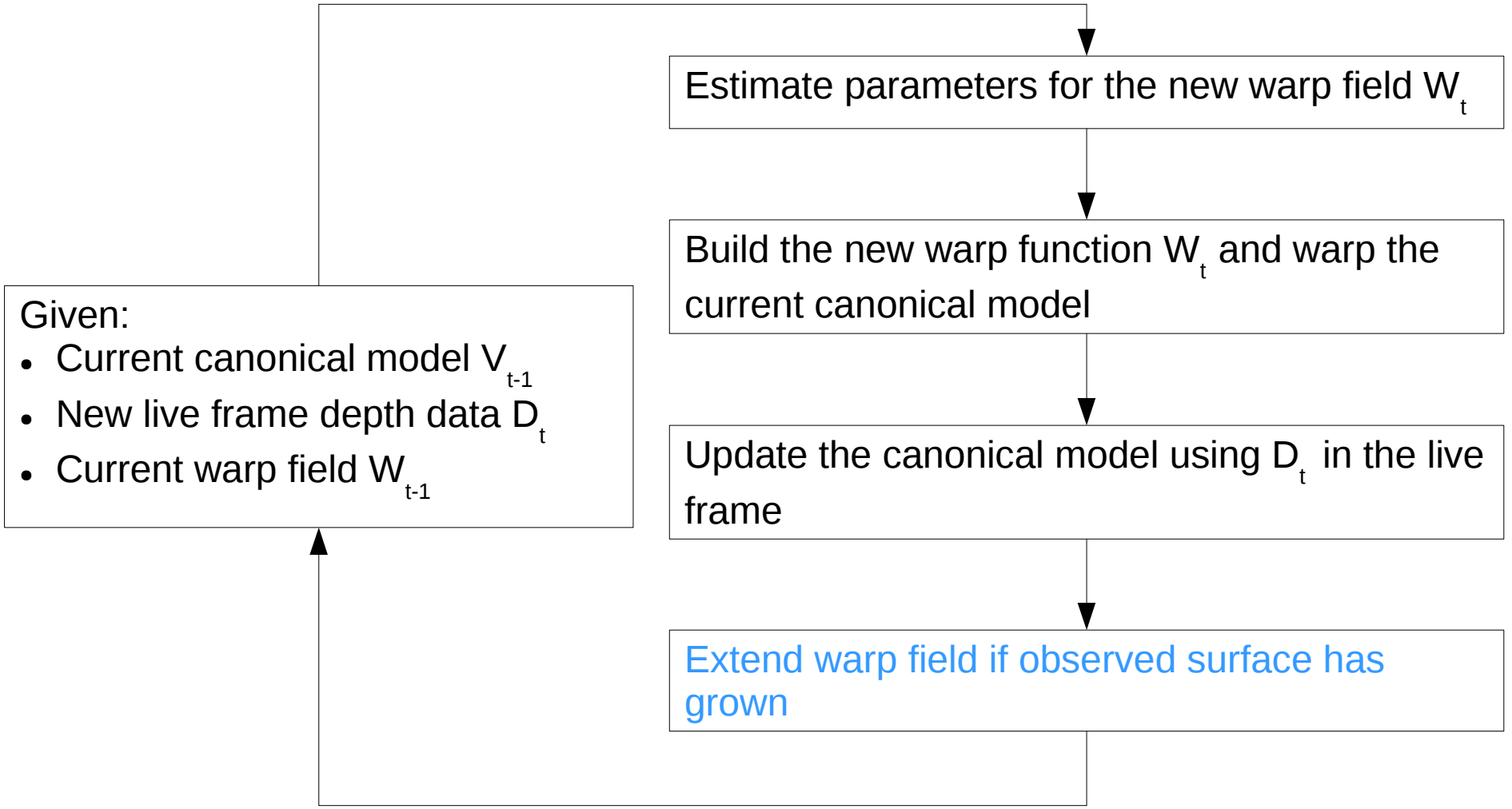
How good does the current warp match the life frame?

$$min\, E(W_t, V, D_t, \epsilon) = Data(W_t, V, D_t) + \lambda\, Reg(W_t, \epsilon)$$

How big is the transformation difference between neighbored nodes?
→ model should deform in a smooth way

→ Use non-linear Gauss-Newton optimization to minimize energy function

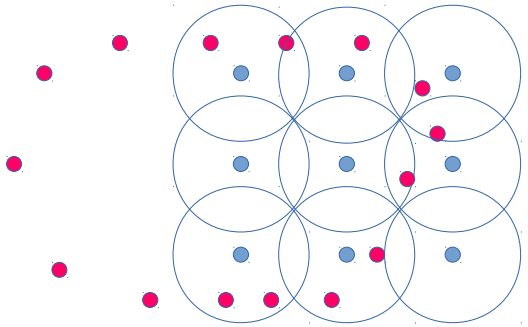
Algorithm Overview



Extending the Warp Field

Surface Estimate of the updated canonical frame

Current deformation nodes with their radius of influence

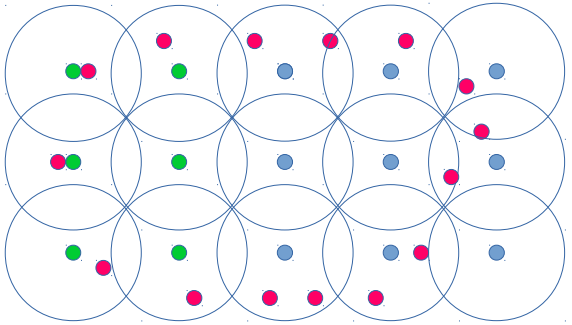


1) Check for each surface point if its covered by a deformation node

Extending the Warp Field

Surface Estimate of the updated canonical frame

Current deformation nodes with their radius of influence



1) Check for each surface point if its covered by a deformation node

2) Subsample the unsupported area with new deformation nodes

3) Initialize the transformation of the new nodes using the interpolation scheme used for warp field

4) Update the edge set

Experiments and Results

Abilities:

- Tracking across motion during reconstruction

*Dynamic***Fusion** takes as input a live depth stream:

Surface Normal Shading

Live Input Depth Map

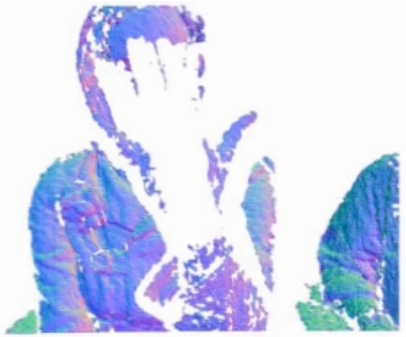
Kinect/xtion/primesense
commodity depth camera

Live RGB Image (unused)

Experiments and Results

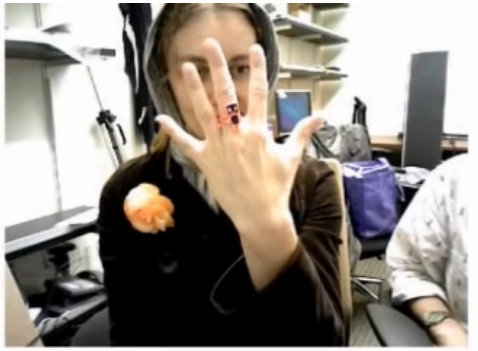
Abilities:

- Tracking across motion during reconstruction
- Fill in initially occluded parts of the scene



Live Input Depth Map

Live Model Output



Live RGB Image (unused)

Canonical Model Reconstruction

Warped Model

Experiments and Results

Abilities:

- Tracking across motion during reconstruction
- Fill in initially occluded parts of the scene
- Generate consistent geometry despite loop closures

“Dieter”

Experiments and Results

Abilities:

- Tracking across motion during reconstruction
- Fill in initially occluded parts of the scene
- Generate consistent geometry despite loop closures
- Movements from open to closed topology

Live Input Depth Map

Live Model Output

Live RGB Image (unused)

Canonical Model Reconstruction

Warped Model

Experiments and Results

Abilities:

- Tracking across motion during reconstruction
- Fill in initially occluded parts of the scene
- Generate consistent geometry despite loop closures
- Movements from open to closed topology

Limitations:

- Large inter-frame movements
- Large deformations of occluded regions
- Movements from closed to open topology
- Growing warp field makes it hard to predict motion of occluded parts

Summary

Dynamic Fusion:

- Uses an online stream of depth images from one RGB-D Camera to reconstruct a 3D model in real time
- Uses a warp field to handle moving scenes and fuse all maps into a canonical frame
- Saves reconstruction as a truncated signed distance function for each point
- Model can be extended, by adding new nodes to the model
- Results in dense reconstruction in canonical frame and increasing quality in life frame reconstruction

Thank you for listening !

Dense Non-Rigid ICP Data-term

1) Extract current surface of canonical frame and save it as point normal pairs
2) Transform these pairs into the new life frame using the current warp field estimate
3) Render parts of warped canonical surface that are currently visible in life frame using perspective projection
4) Compute for each point of the rendered canonical surface the model-to-frame point plane error

Robust Turkey penalty function
Warped point normal predictions from canonical frame
Life frame point normal

$$Data(W, V, D_t) = \sum_{u \in \Omega} \psi_{data}\left(n^T\left(v - vl\right)\right)$$

Warp Field Regularisation

Sum transformation differences of all in the edge set pairwise connected nodes

$$Reg(W, \varepsilon) = \sum_{i=0}^{n} \sum_{j \in \varepsilon(i)} \alpha_{ij} \psi_{reg} \left(T_{locali}\, pos_i - T_{localj}\, pos_i \right)$$

Edge set
Weight associeted with the edge
Huber penalty function

Regularisation Graph

- Hierarchical Deformation Tree
- Construct hierarchy of regularization nodes from current deformation nodes
  - Level l = 0 all nodes from N_{warp}
  - Level l = 1 all nodes that are in rage of $\epsilon\beta^{l}$
  - ….
- Build graph topology by adding edges from each node of the hierarchy to the k-nearest nodes in the next coarser level