Geometric Registration for Deformable Shapes

4.2 Animation Reconstruction

Basic Algorithm · Efficiency: Urshape Factorization

31st Annual Conference of the European Association for Computer Graphics
eurographics 2010

Overview & Problem Statement

Overview

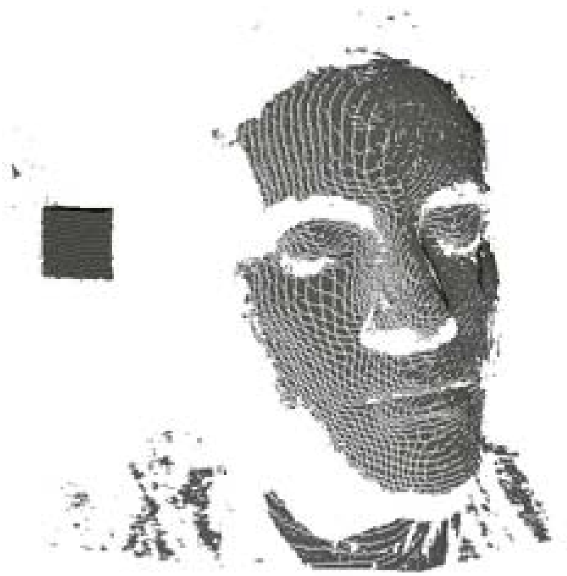
Two Parallel Topics

- Basic algorithms
- Two systems as a case study

Animation Reconstruction

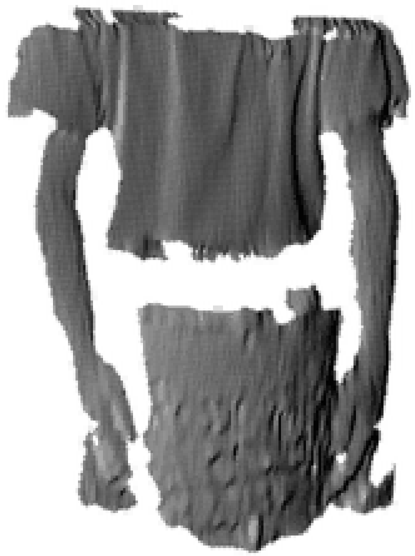
- Problem Statement
- Basic algorithm (original system)
  - Variational surface reconstruction
  - Adding dynamics
  - Iterative Assembly
  - Results
- Improved algorithm (revised system)

Real-time Scanners



space-time stereo

courtesy of James Davis, UC Santa Cruz



color-coded structured light

courtesy of Phil Fong, Stanford University

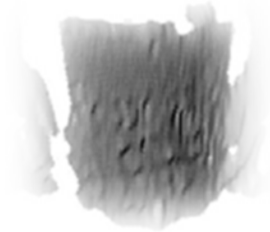
motion compensated structured light

courtesy of Sören König, TU Dresden

Animation Reconstruction

Problems

- Noisy data
- Incomplete data (acquisition holes)
- No correspondences



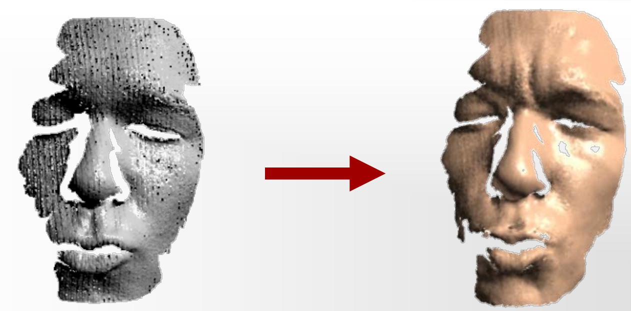
noise

holes

missing correspondences

Animation Reconstruction

Remove noise, outliers



Fill-in holes
(from all frames)

Dense correspondences

Animation Reconstruction

Surface Reconstruction

Variational Approach

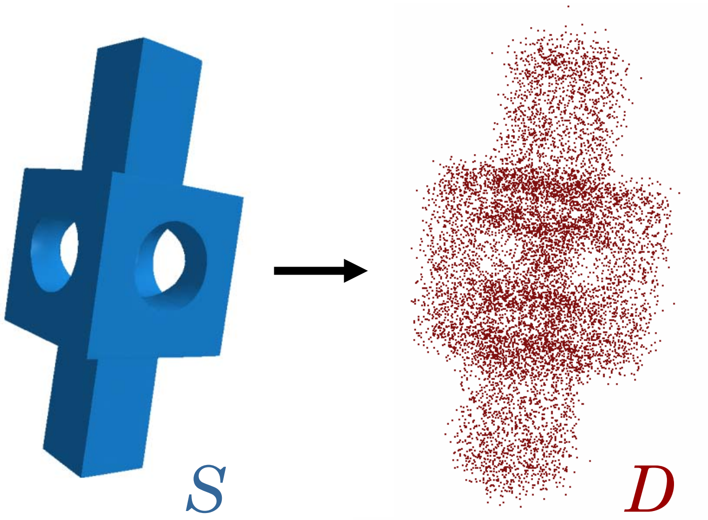
Variational Approach:

- S – original model
  D – measurement data

- Variational approach:

$$E(S \mid D) \sim E(D \mid S) + E(S)$$

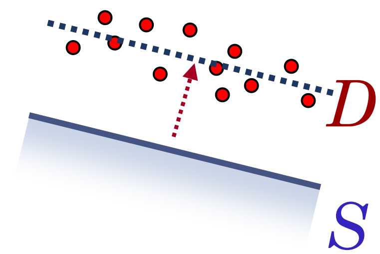
measurement — prior



3D Reconstruction

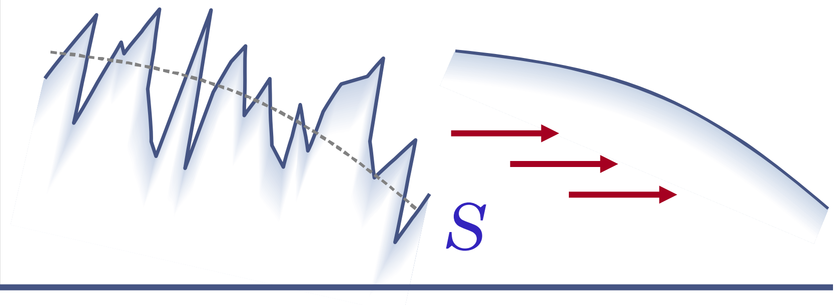
Data fitting

$$E(D \mid S) \sim \Sigma_i \, \mathrm{dist}(S, d_i)^2$$



Prior: Smoothness

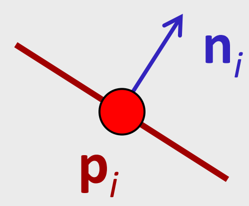
$$E_s(S) \sim \int_S \mathrm{curv}(S)^2$$



Implementation...

Implementation: Point-based model

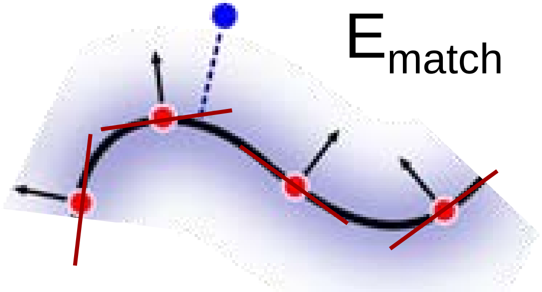
- Our model is a set of points
- "Surfels": Every point has a latent surface normal
- We want to estimate *position* and *normals*



"Surfel"

Data Term – E(D|S)

Data fitting term:

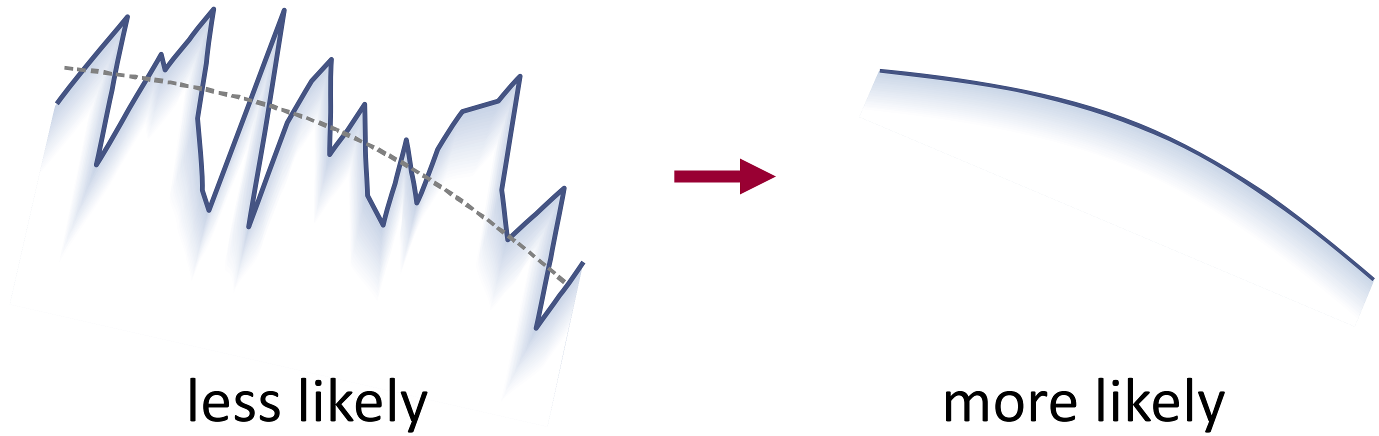


- Surface should be close to data
- Truncated squared distance function

$$E_{match}(D,S) = \sum_{data\,pts} trunc_{\delta}(dist(S,\mathbf{d}_i)^2)$$

- Sum of distances² of data points to surfel planes
- Point-to-plane: No exact 1:1 match necessary
- Truncation (M-estimator): Robustness to outliers

Priors – P(S)



Canonical assumption: smooth surfaces

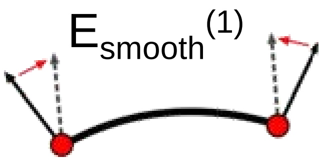
- Correlations between neighboring points

Point-based Model

Simple Smoothness Priors:

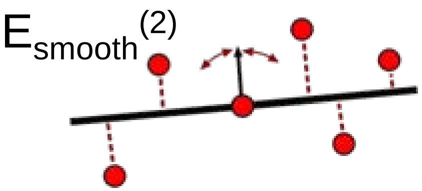
- Similar surfel normals:

$$E^{(1)}_{smooth}(S) = \sum_{surfels} \sum_{neighbors} \left(n_i - n_{i_j}\right)^2, \ \|n_i\| = 1$$



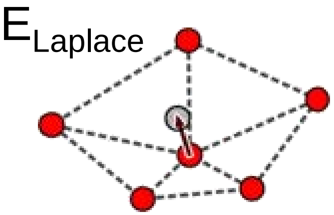
- Surfel positions – flat surface:

$$E^{(2)}_{smooth}(S) = \sum_{surfels} \sum_{neighbors} \left\langle \mathbf{s}_i - \mathbf{s}_{i_j}, \mathbf{n}(\mathbf{s}_i) \right\rangle^2$$



- Uniform density:

$$E_{Laplace}(S) = \sum_{surfels} \sum_{neighbors} \left(\mathbf{s}_i - average\right)^2$$



[c.f. Szeliski et al. 93]

Nasty Normals

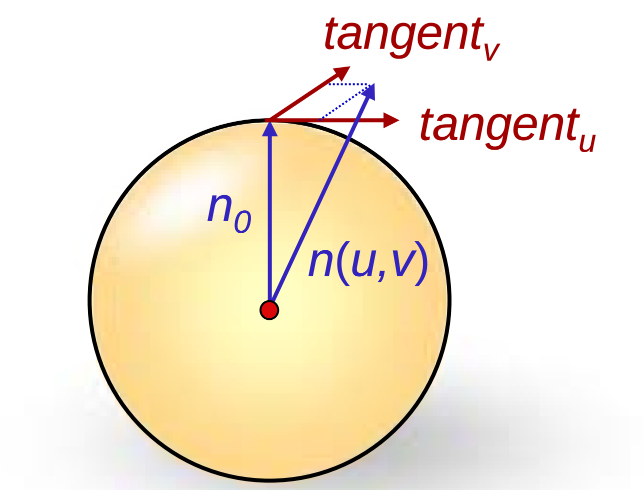
Optimizing Normals

- Problem: $E^{(1)}_{smooth}(S) = \sum_{surfels} \sum_{neighbors} \left(n_i - n_{i_j}\right)^2, \ \ s.t. \left\|n_i\right\| = 1$
- Need unit normals: constraint optimization
- Unconstraint: trivial solution (all zeros)

Nasty Normals

Solution: Local Parameterization

- Current normal estimate
- Tangent parameterization
- New variables *u, v*
- Renormalize
- Non-linear optimization
- No degeneracies



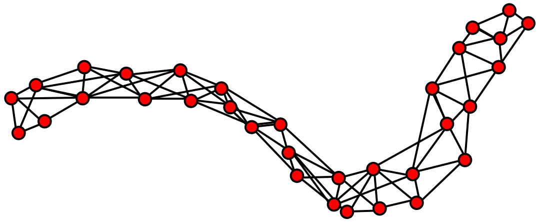
$$n(u,v) = n_0 + u \cdot tangent_u + v \cdot tangent_v$$

[Hoffer et al. 04]

Neighborhoods?

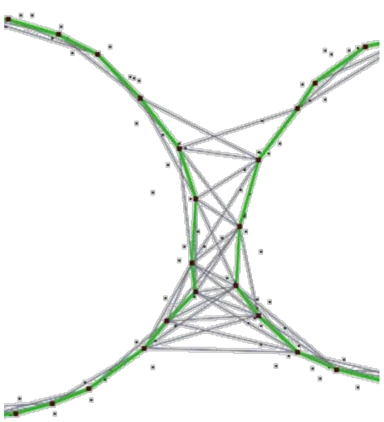
Topology estimation

- Domain of *S*, base shape (topology)
- Here, we assume this is easy to get
- In the following
  - *k*-nearest neighborhood graph
  - Typically: k = 6..20



Limitations

- This requires dense enough sampling
- Does not work for undersampled data



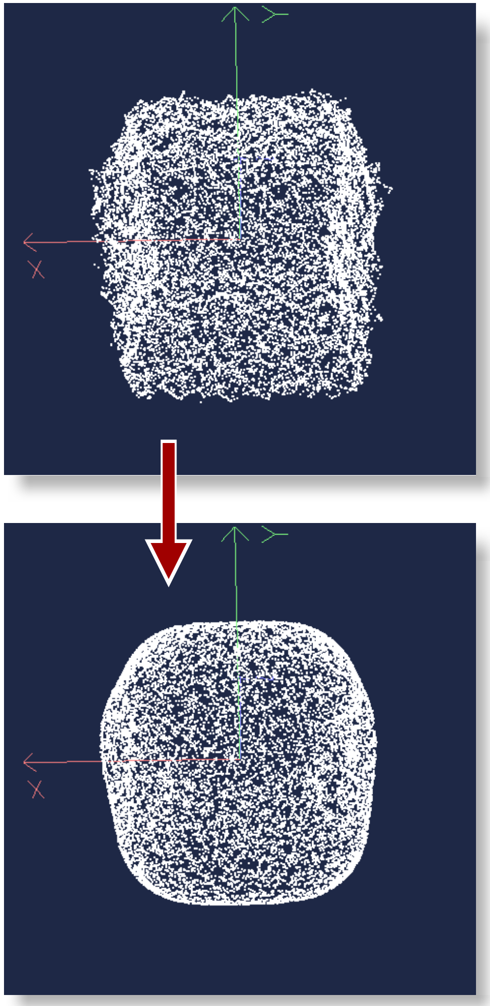
Numerical Optimization

Task:

- Compute most likely "original scene" S
- Nonlinear optimization problem

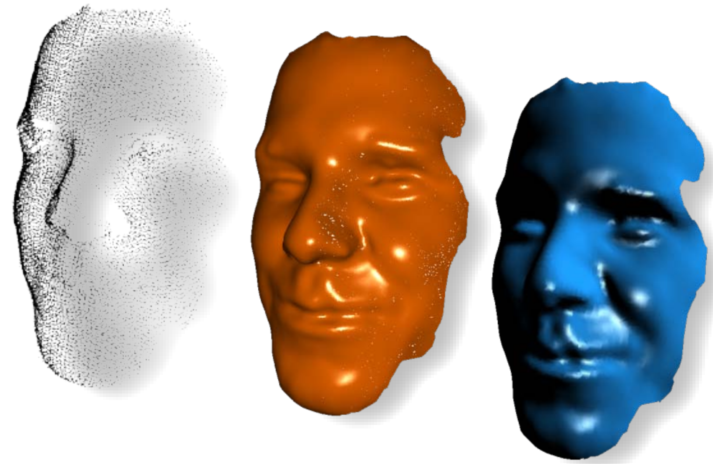
Solution:

- Create initial guess for S
  - Close to measured data
  - Use original data
- Find local optimum
  - (Conjugate) gradient descent
  - (Gauss-) Newton descent

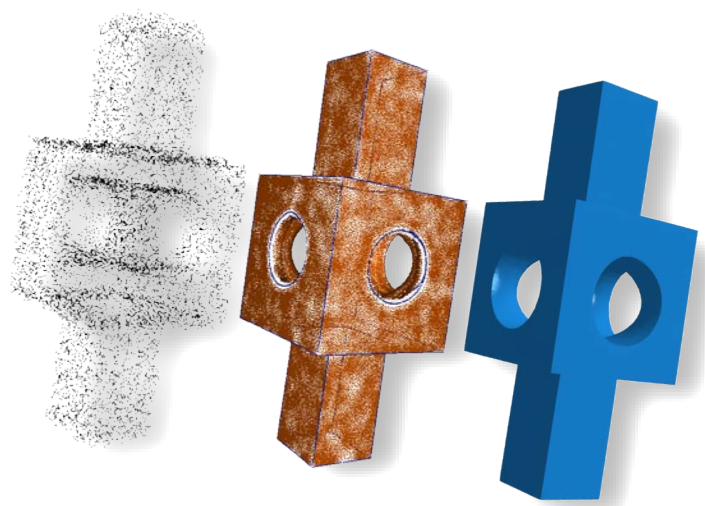


3D Examples

3D reconstruction results:



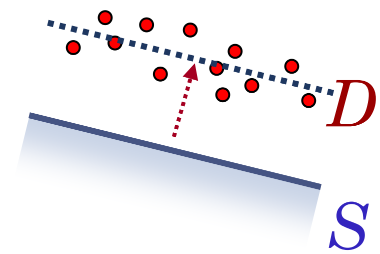
(With discontinuity lines, not used here):



3D Reconstruction Summary

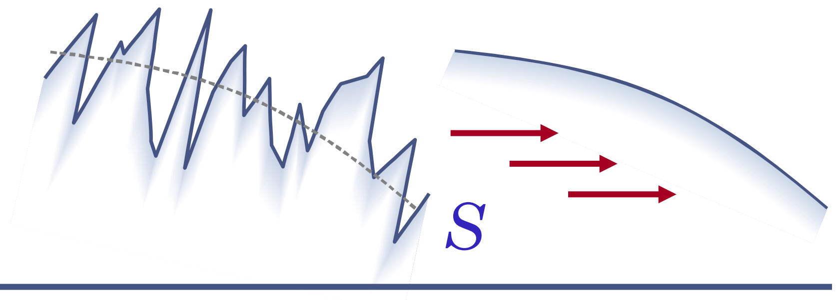
Data fitting:

$E(D \mid S) \sim \Sigma_i \, \mathrm{dist}(S, d_i)^2$



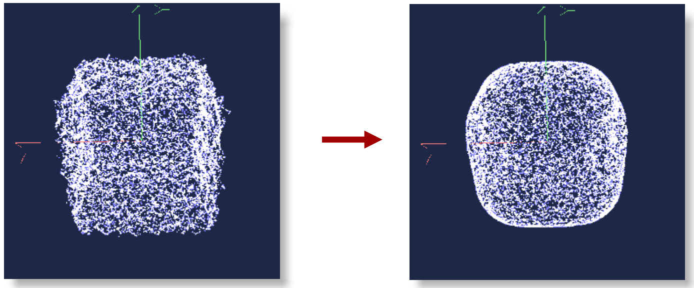
Prior: Smoothness

$E_s(S) \sim \int_S \mathrm{curv}(S)^2$



Optimization:

Yields 3D Reconstruction



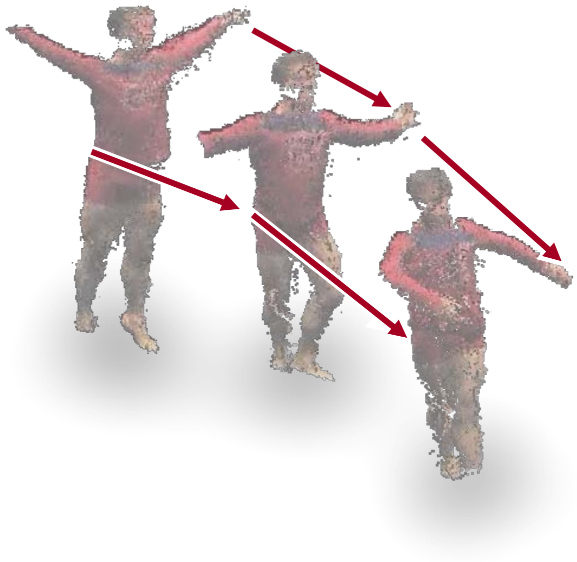
Animation Reconstruction

Adding the Dynamics

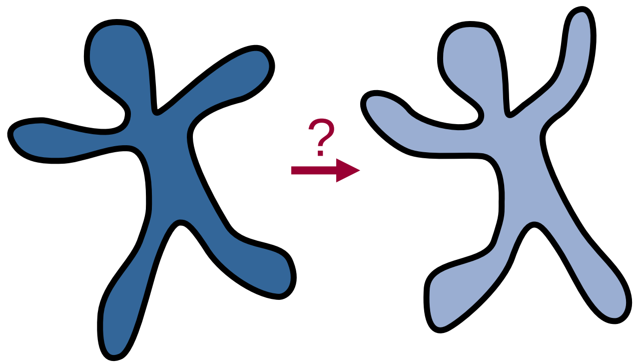
Extension to Animations

Animation Reconstruction

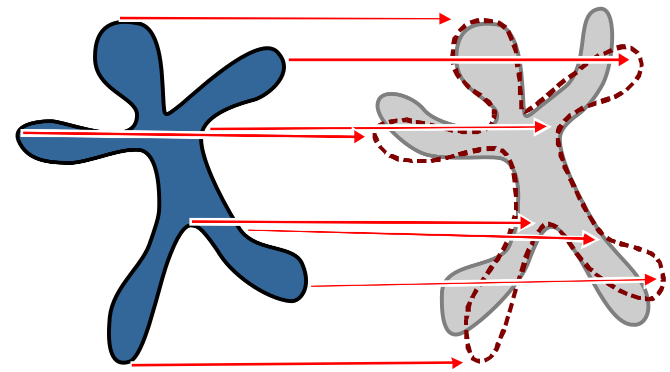
- Not just a 4D version
  - Moving geometry,
    not just a smooth hypersurface
- Key component: correspondences
- Intuition for “good correspondences”:
  - Match target shape
  - Little deformation



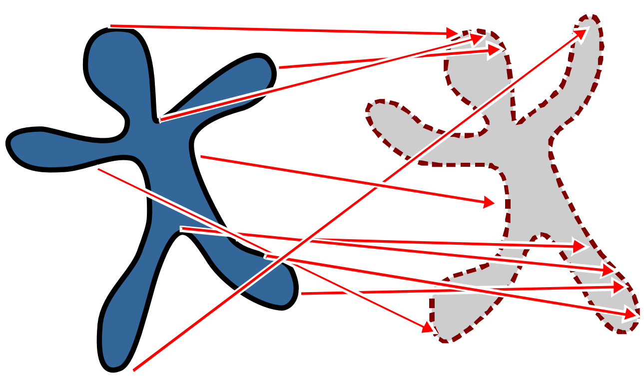
Recap: Correspondences



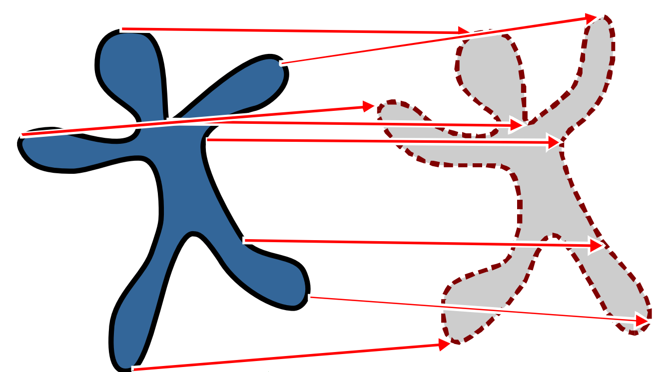
Correspondences?



✗ no shape match



✗ too much deformation



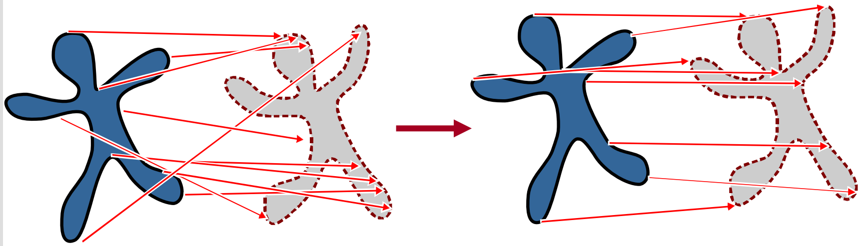
✓ optimum

Animation Reconstruction

Two additional priors:

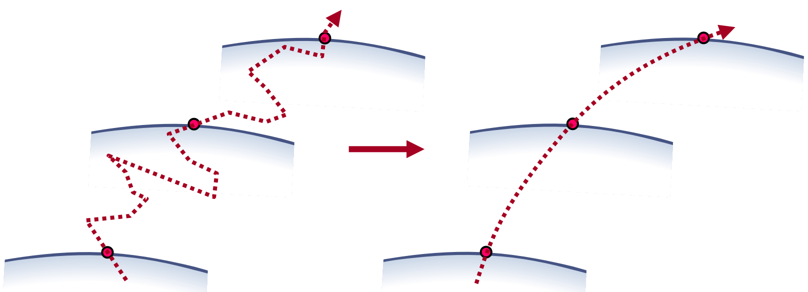
Deformation

$$E_d(S) \sim \int_S \mathrm{deform}(S_t, S_{t+1})^2$$

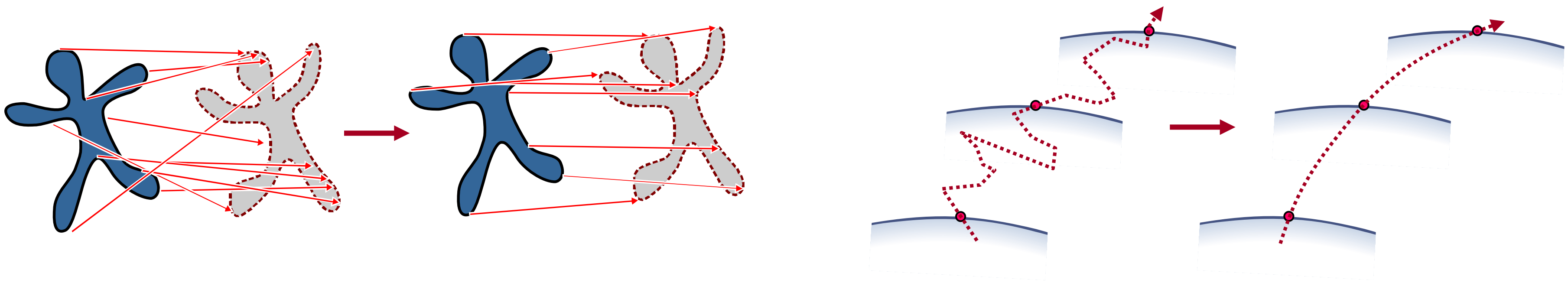


Acceleration

$$E_a(S) \sim \int_{S,t} \ddot{s}(x, t)^2$$



Animation Reconstruction



Not just smooth 4D reconstruction!

- Minimize
  - Deformation
  - Acceleration
- This is quite different from smoothness of a 4D hypersurface.

Animations

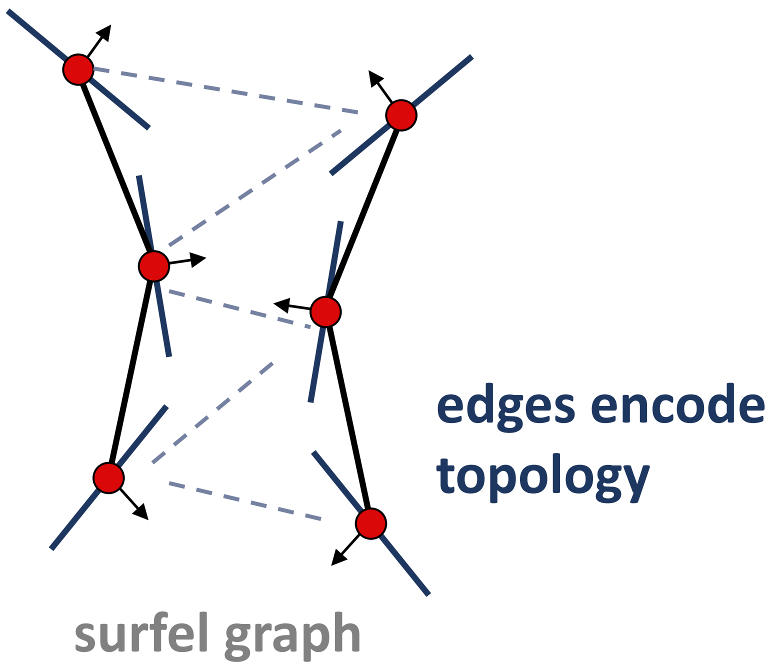
Refined parametrization of reconstruction *S*

- Surfel graph (3D)
- Trajectory graph (4D)

Discretization

Refined parametrization of reconstruction *S*

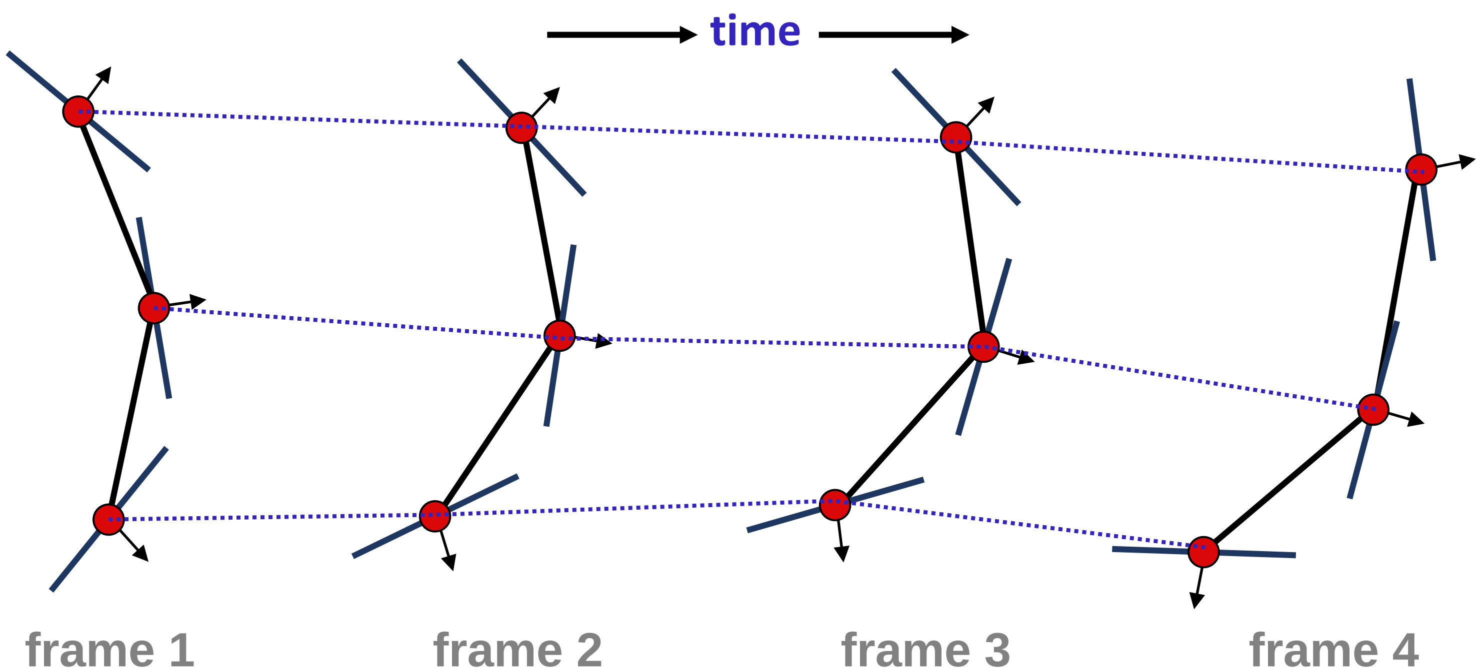
- Surfel graph (3D)
- Trajectory graph (4D)



Discretization

Refined parametrization of reconstruction *S*

- Surfel graph (3D)
- Trajectory graph (4D)



Missing Details...

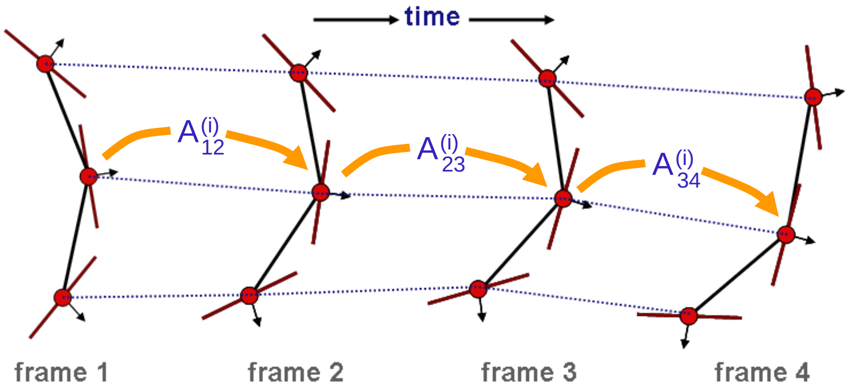
How to implement...

- The deformation priors?
  - We use one of the models previously developed
- Acceleration priors?
  - This is rather simple...

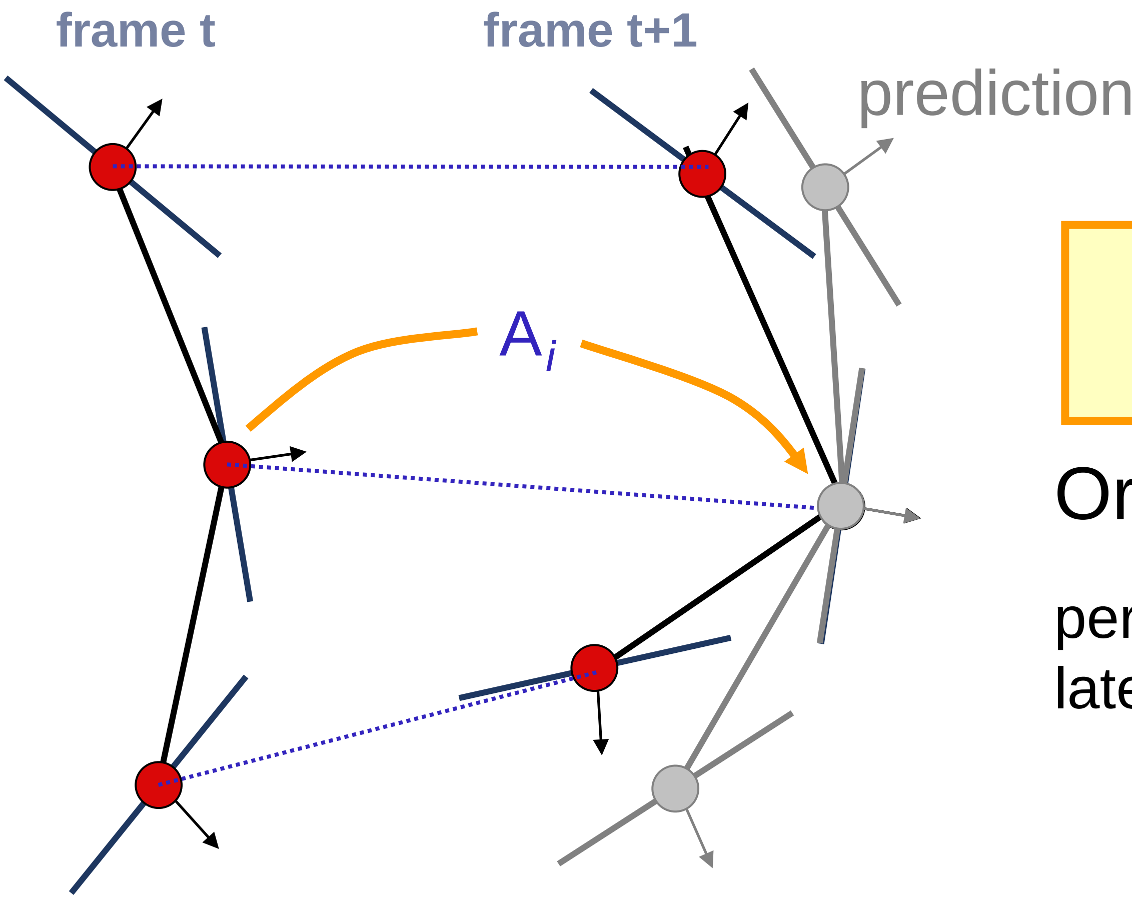
Recap: Elastic Deformation Model

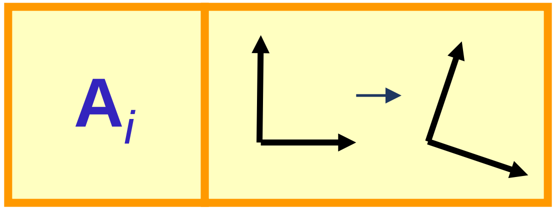
Deformation model

- Latent transformation $\mathbf{A}^{(i)}$ per surfel
- Transforms *neighborhood* of s_i
- Minimize error (both surfels and $\mathbf{A}^{(i)}$)



Recap: Elastic Deformation Model

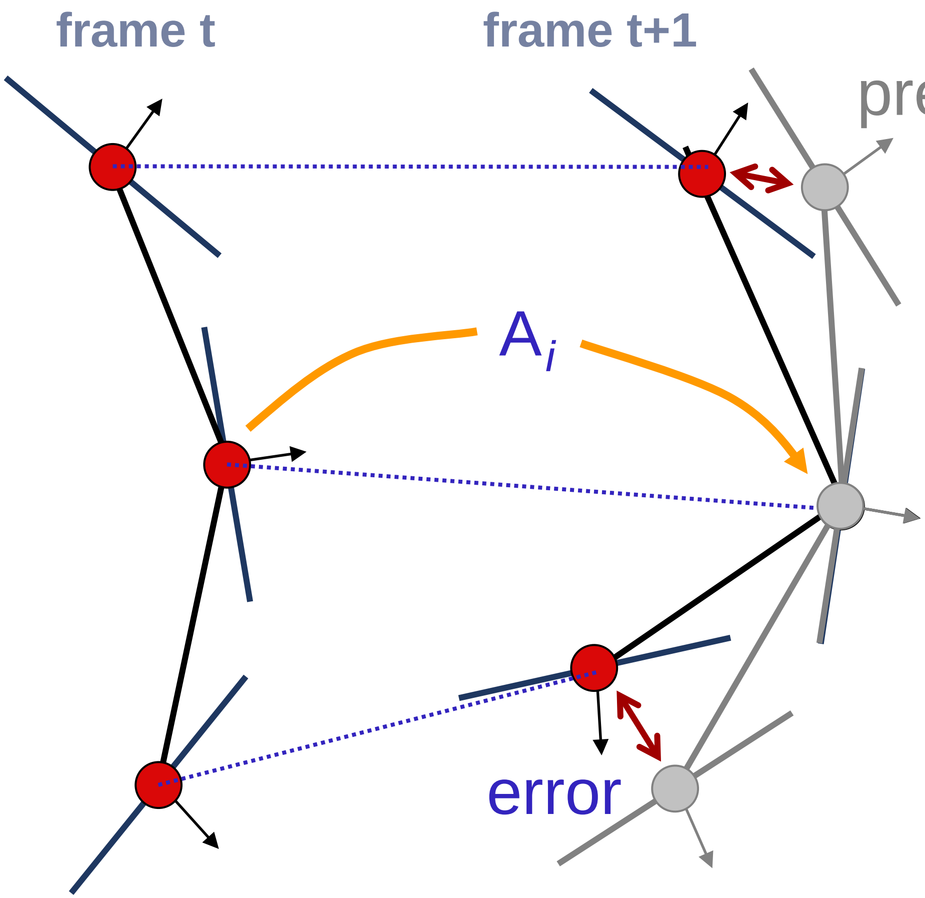


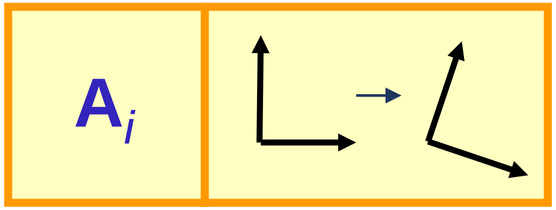


Orthonormal Matrix $\mathbf{A}_i$

per surfel (neighborhood), latent variable

Recap: Elastic Deformation Model





Orthonormal Matrix $\mathbf{A}_i$

per surfel (neighborhood), latent variable

$$E_{deform}(S) = \sum_{surfels} \sum_{neighbors} \left[\mathbf{A}_i^t \left(\mathbf{s}_i^{(t)} - \mathbf{s}_{i_j}^{(t)} \right) - \left(\mathbf{s}_i^{(t+1)} - \mathbf{s}_{i_j}^{(t+1)} \right) \right]^2$$

Recap: Unconstrained Optimization

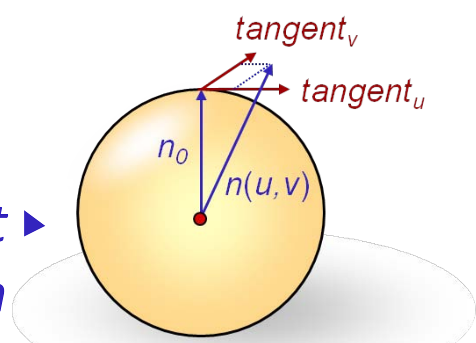
Orthonormal matrices

- Local, 1st order, non-degenerate parametrization:

$$\mathbf{C}_{\times i}^{(t)} = \begin{pmatrix} 0 & \alpha & \beta \\ -\alpha & 0 & \gamma \\ -\beta & -\gamma & 0 \end{pmatrix} \qquad \begin{aligned} \mathbf{A}_i &= \mathbf{A}_0 \exp(\mathbf{C}_{\times i}) \\ &\doteq \mathbf{A}_0 (I + \mathbf{C}_{\times i}^{(t)}) \end{aligned}$$

- Optimize parameters α, β, γ, then recompute $\mathbf{A}_0$
- Compute initial estimate using [*Horn 87*]

c.f: unconstraint normal optimization ▸



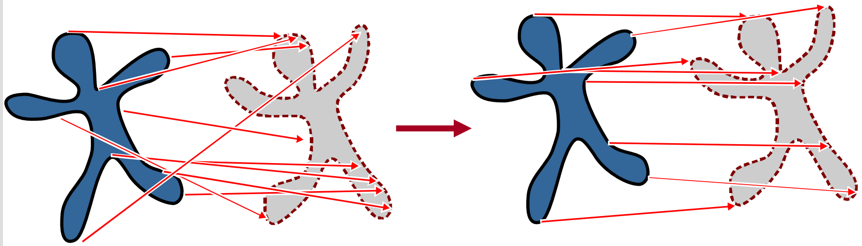
$$n(u,v) = n_0 + u \cdot tangent_u + v \cdot tangent_v$$

Animation Reconstruction

Two additional priors:

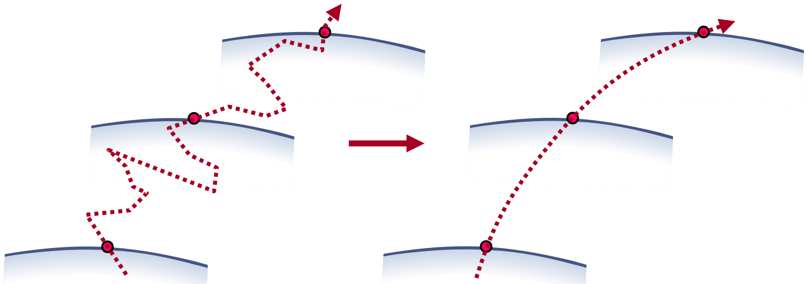
Deformation

$$E_d(S) \sim \int_S \text{deform}(S_t, S_{t+1})^2$$



Acceleration

$$E_a(S) \sim \int_{S,t} \ddot{s}(x, t)^2$$

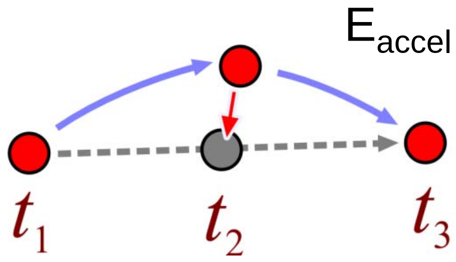


Acceleration

Acceleration priors

- Penalize non-smooth trajectories

$$E_{accel}(A) = \left[\mathbf{s}_i^{t-1} - 2\mathbf{s}_i^{t} + \mathbf{s}_i^{t+1}\right]^2$$



- Filters out temporal noise

Optimization

For optimization, we need to know:

- The surfel graph
- A (rough) initialization close to correct solution

Optimization:

- Non-linear *continuous optimization* problem
- Gauss-Newton solver (fast & stable)

How do we get the initialization?

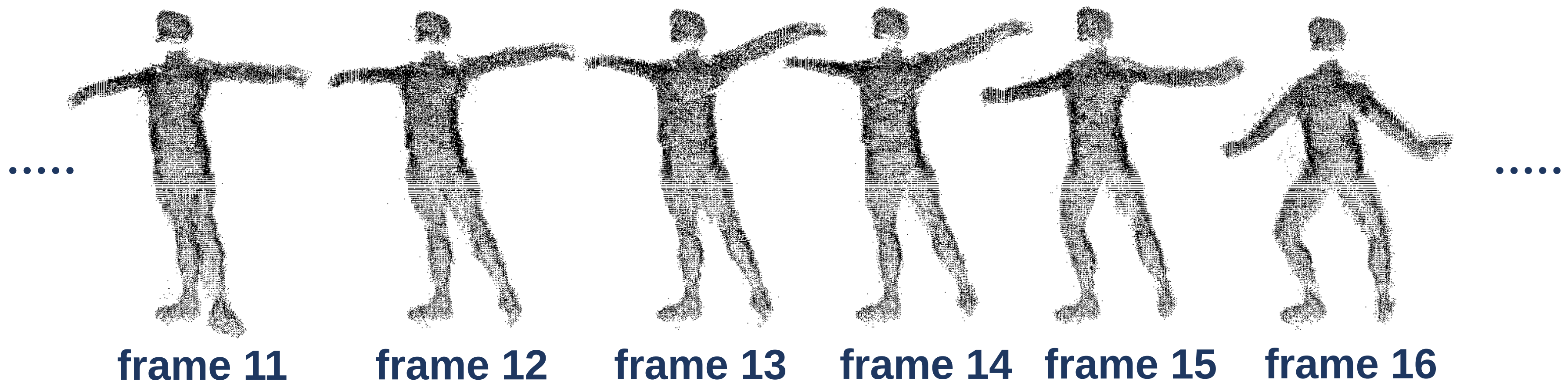
- *Iterative assembly* heuristic to build & init graph

Iterative Assembly

Global Assembly

Assumption: Adjacent frames are similar

- Every frame is a good initialization for the next one
- Solve for frame pairs

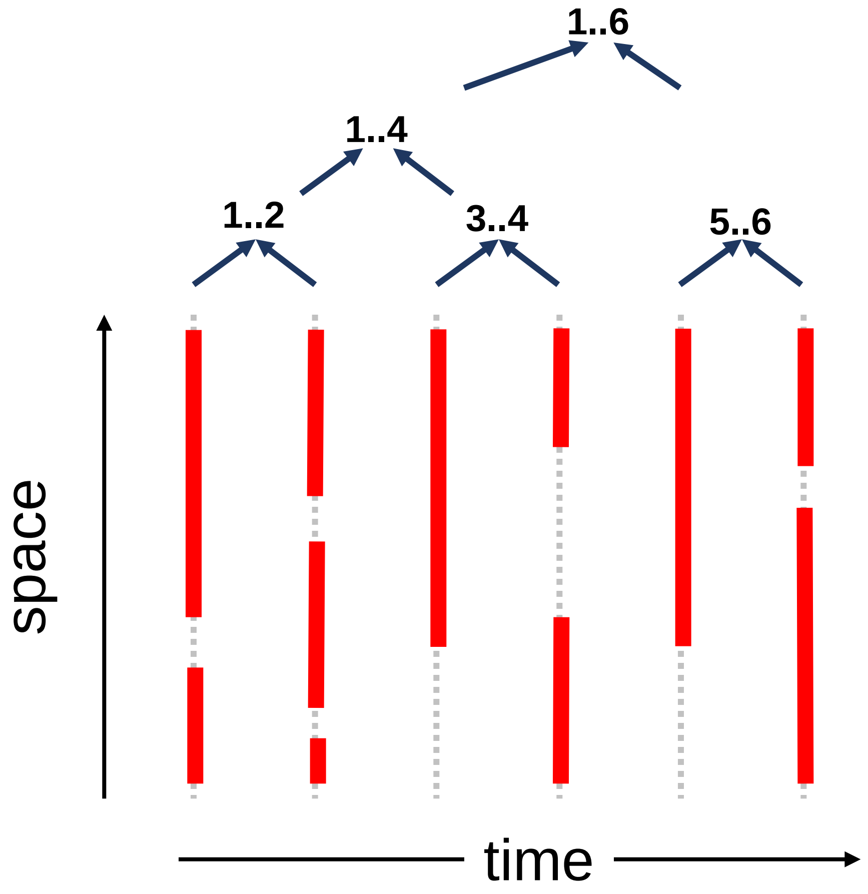


[data set courtesy of C. Theobald, MPI-Inf]

Iterative Assembly

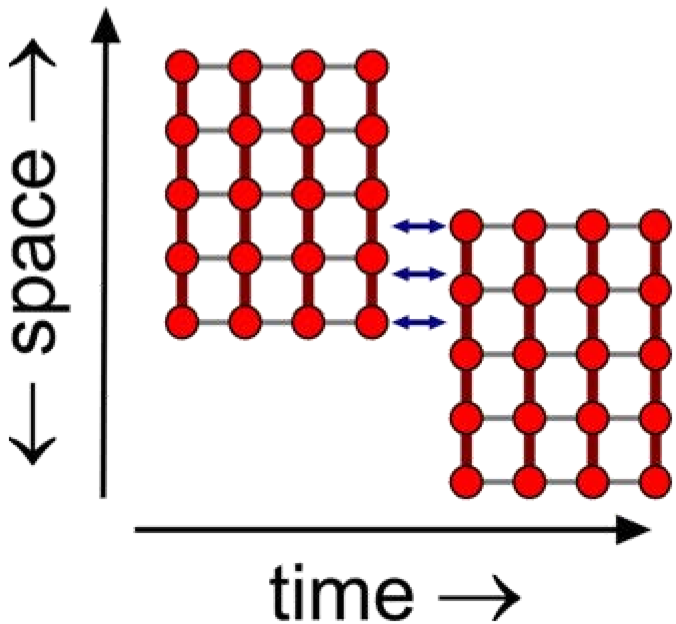
Iterative assembly

- Merge adjacent frames
- Propagate hierarchically
- Global optimization
  (avoid error propagation)

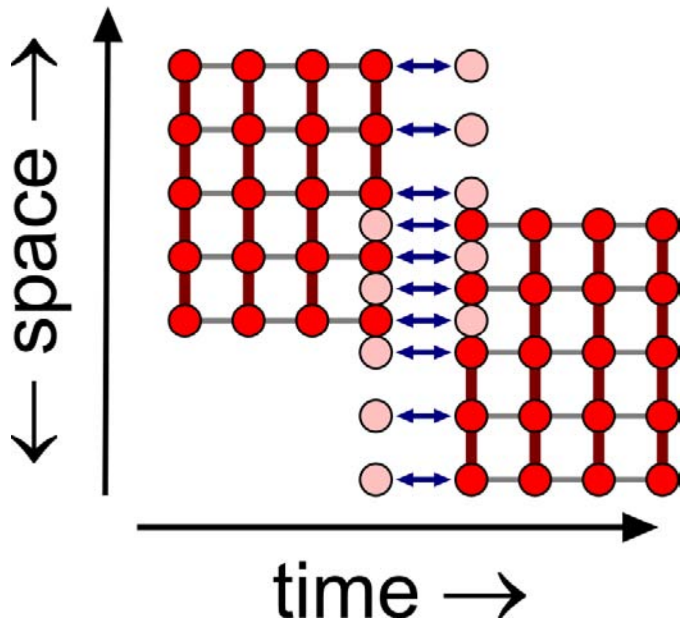


Iterative Assembly

Pairwise alignment



adjacent
trajectory sets

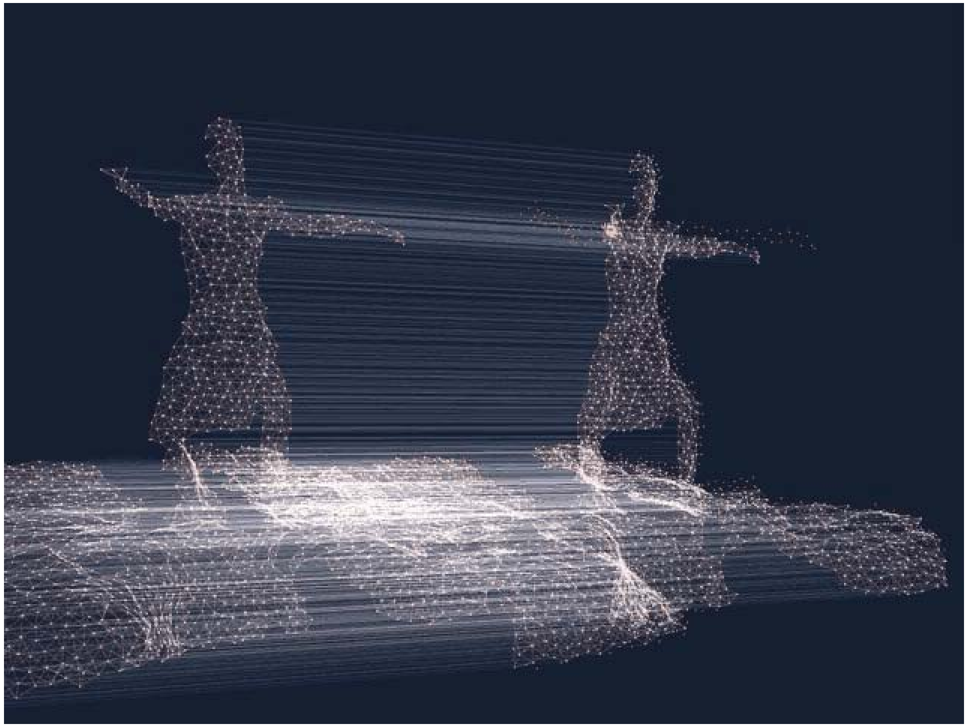


aligned
frames

Alignment

Alignment:

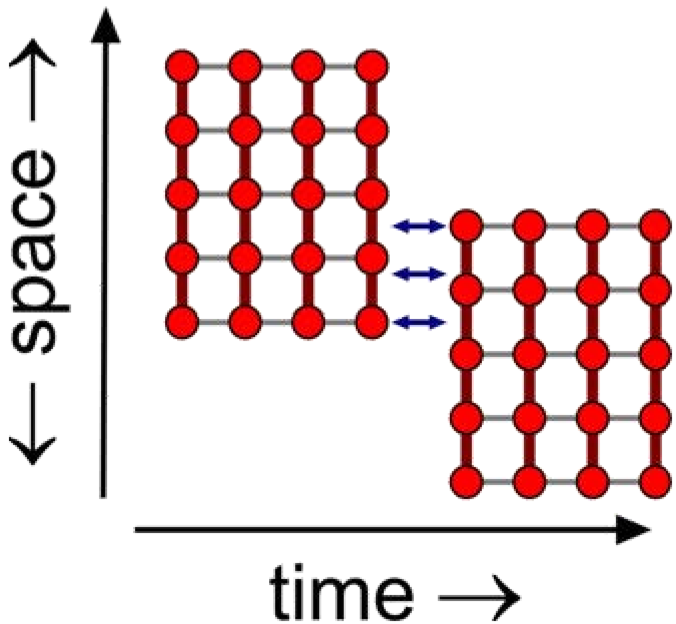
- Two frames
- Use one frame as initialization
- Second frame as “data points”
- Optimize



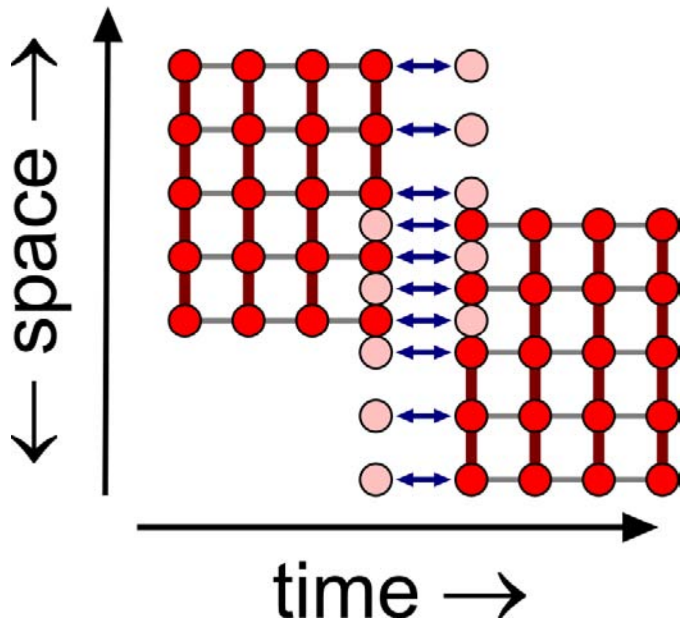
[data set: Zitnick et al., Microsoft Research]

Iterative Assembly

Pairwise alignment



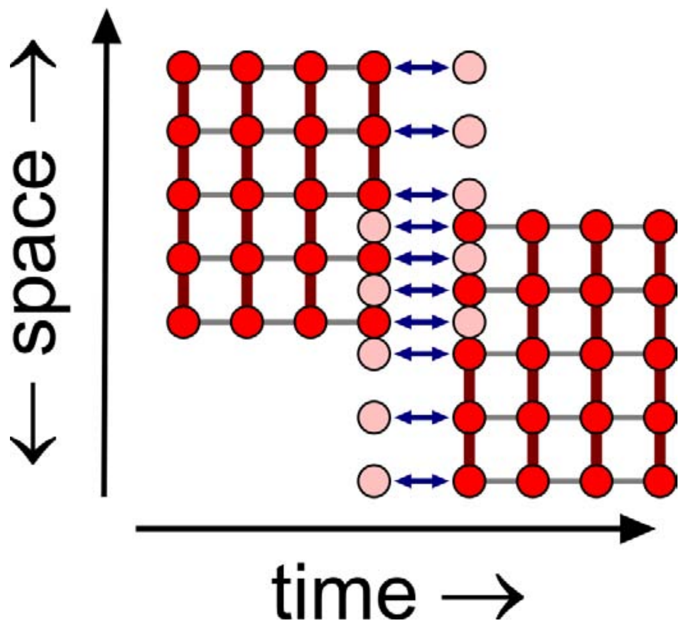
adjacent
trajectory sets



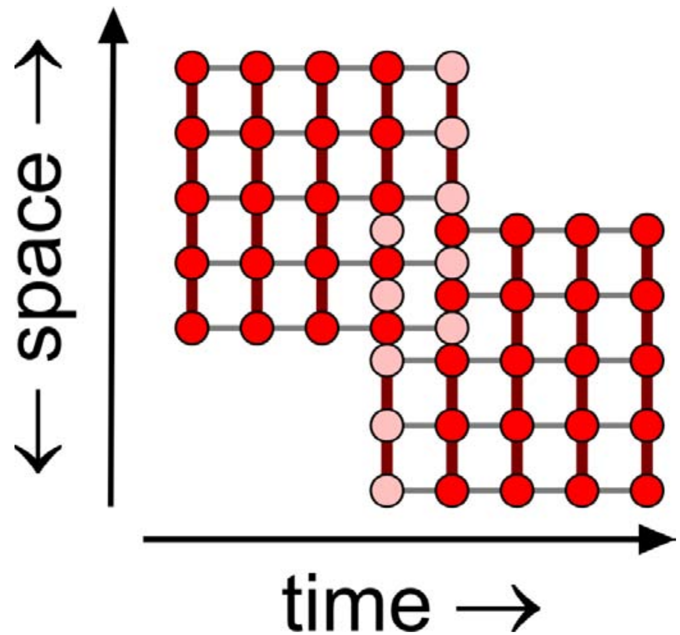
aligned
frames

Iterative Assembly

Topology stitching



aligned
frames

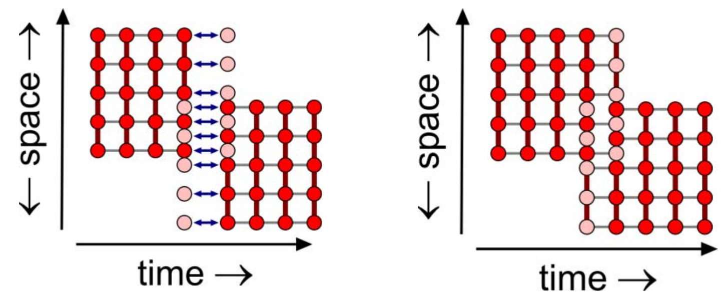


merged
topology

Topology Stitching

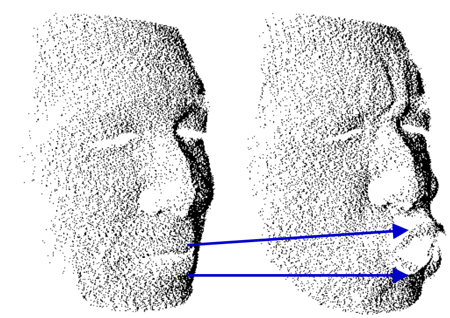
Recompute Topology

- Recompute kNN/ε-graph
- Topology is global



Sanity Check:

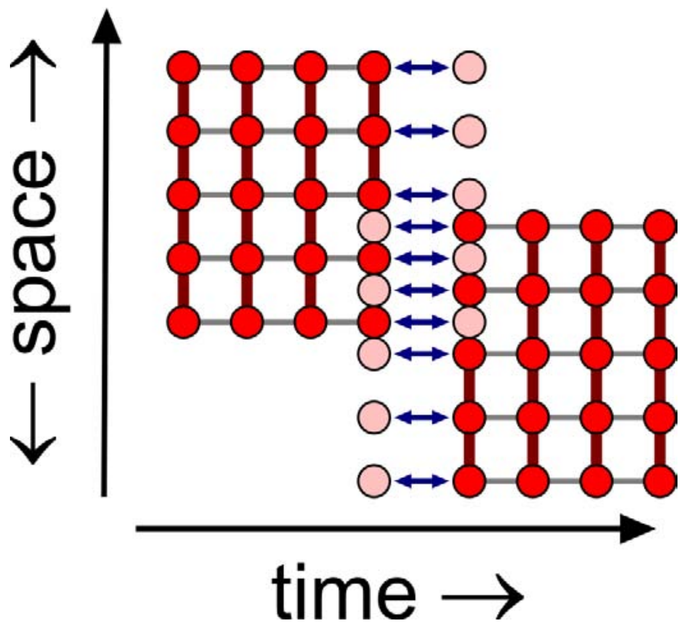
- No connection if distance changes



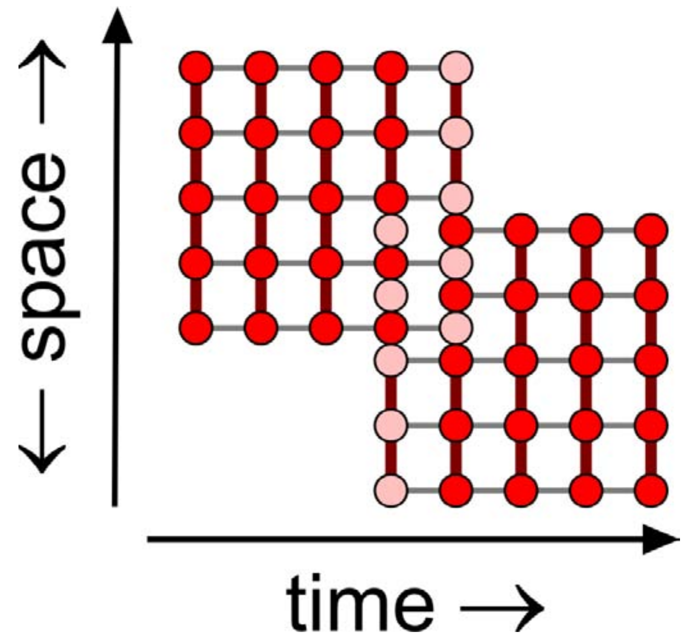
[data set courtesy of S. König, S. Gumhold, TU Dresden]

Iterative Assembly

Topology stitching



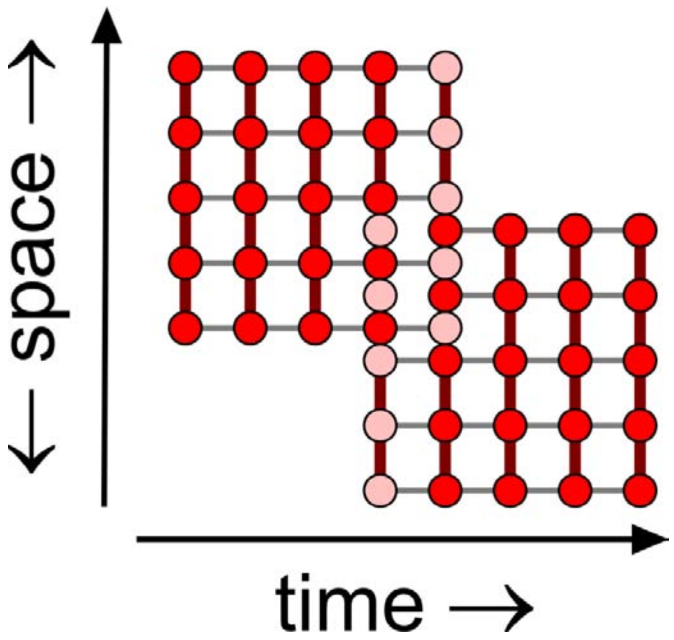
aligned
frames



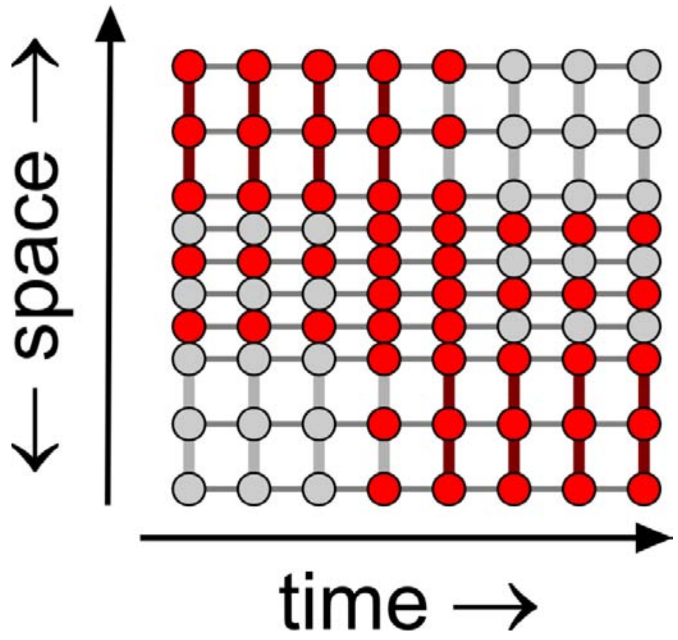
merged
topology

Iterative Assembly

Problem: incomplete trajectories



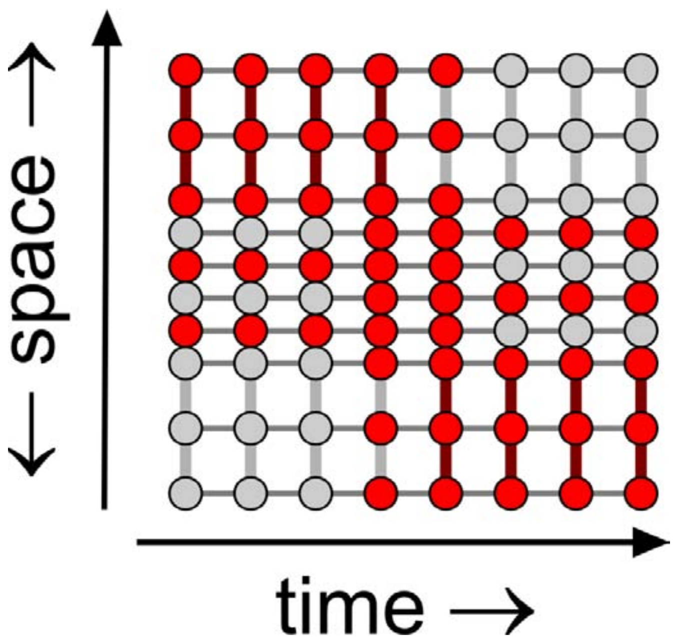
merged
topology



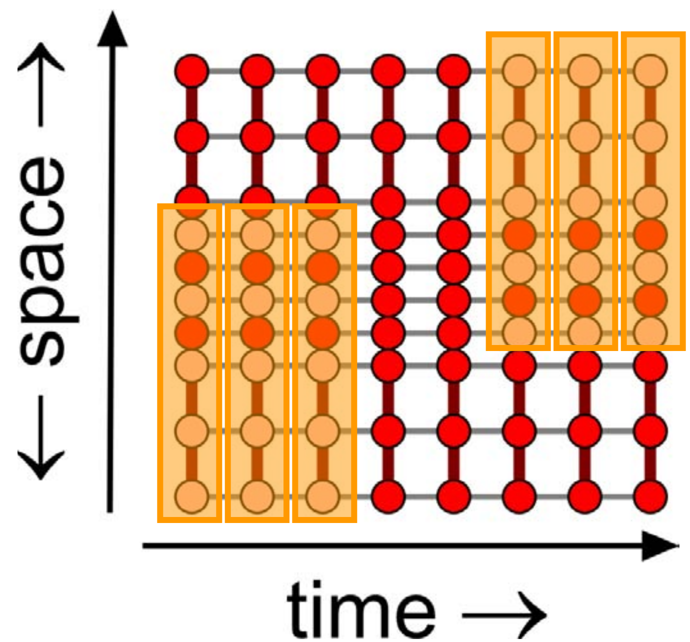
uninitialized
surfels

Iterative Assembly

Hole filling



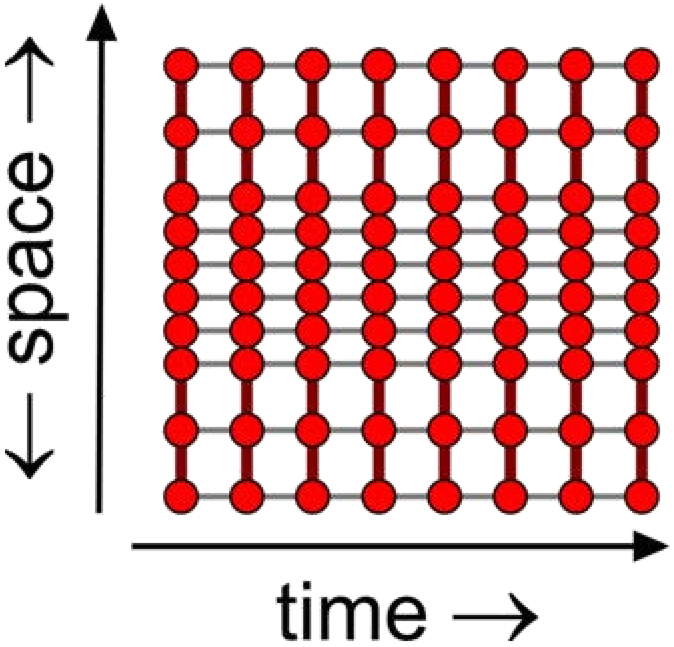
uninitialized
surfels



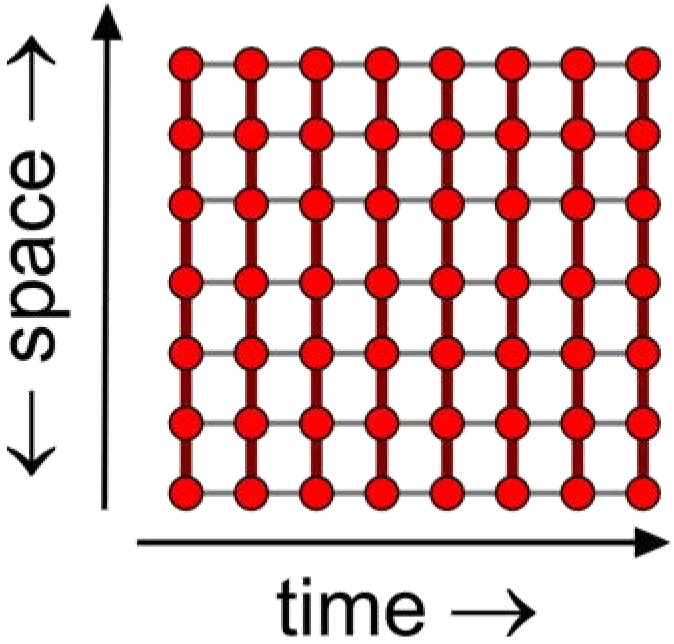
copy from neighbors,
optimize

Iterative Assembly

Resampling



hole filled
result

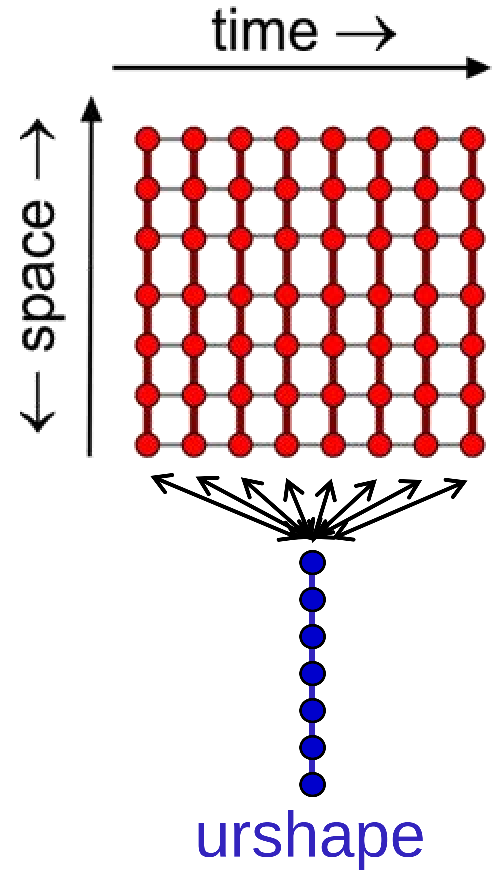


remove dense surfels
(constant complexity)

Global Optimization

Last step:

- Global optimization
- Optimize over all frames simultaneously



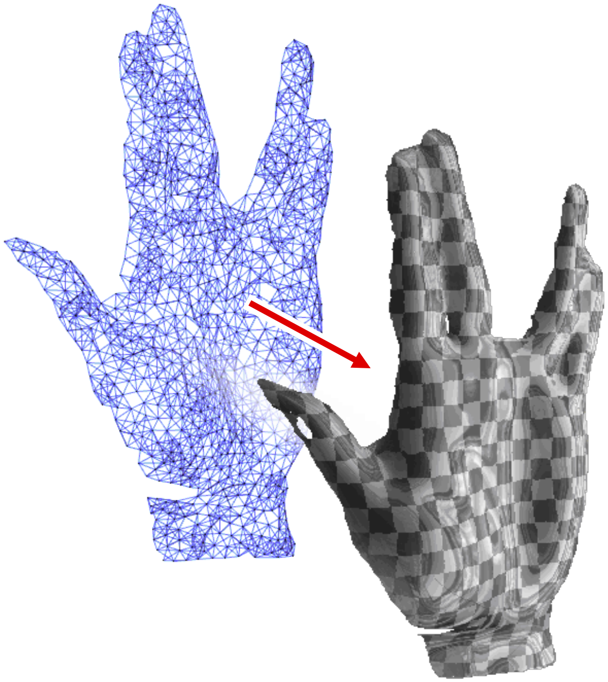
Improve stability: Urshapes

- Connect hidden “latent” frame to all other frames (deformation prior only)
- Initialize with one of the frames

Meshing

Last step: create mesh

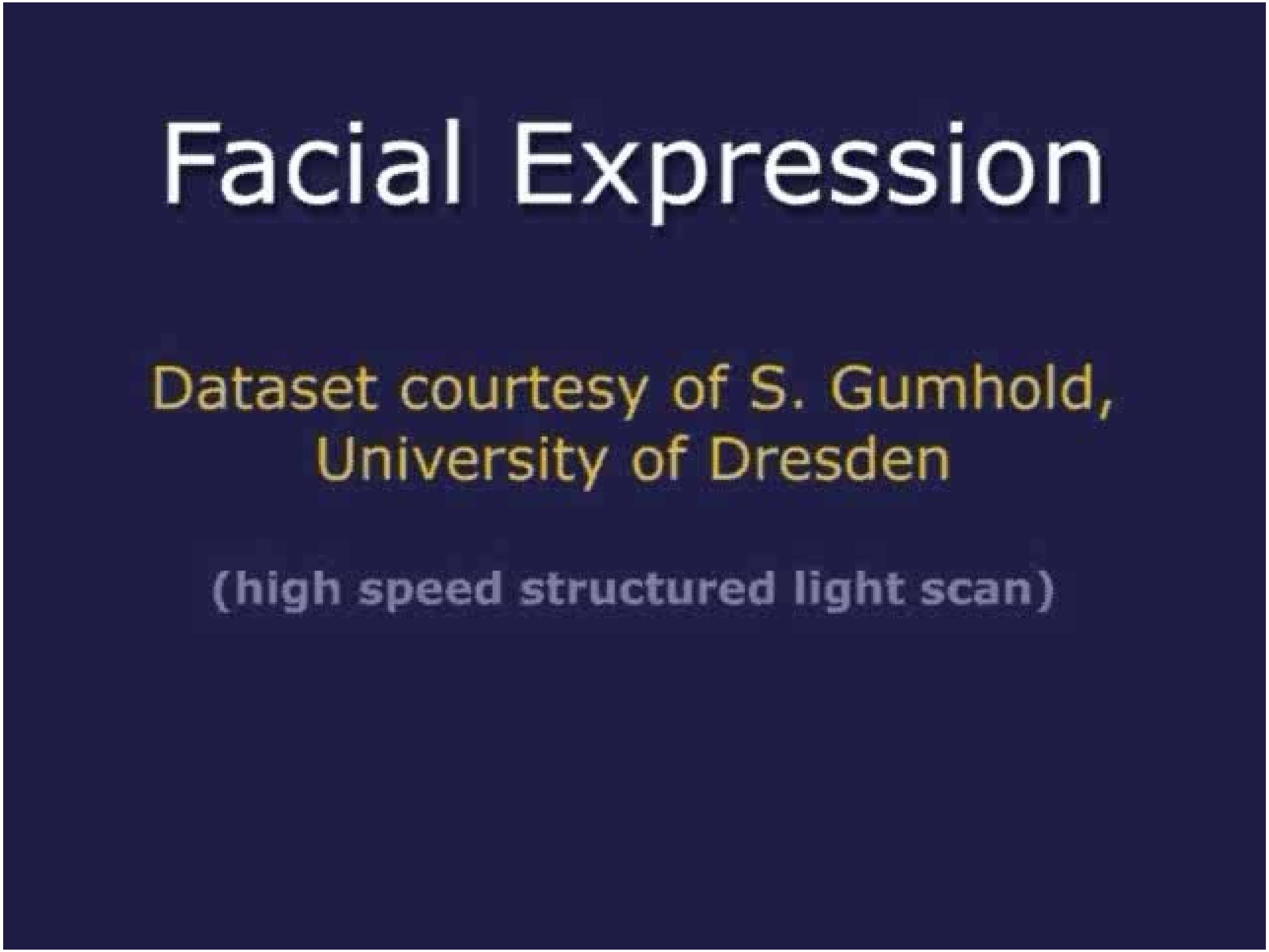
- After complete surfel graph is reconstructed
- Pick one frame (or urshape)
- "Marching cubes" meshing [*Hoppe et al. 92*, *Shen et al. 04*]
- Morph according to trajectories (local weighted sum)



[data set courtesy of O. Schall, MPI Informatik Saarbrücken]

Results

frames	surfels	data pts	preprocessing	reconstruction	
20	49,500	963,671	6 min 52 sec	4 h 25 min	[Pentium-4, 3.4GHz]



frames	surfels	data pts	preprocessing	reconstruction	
20	32,740	400,000	6 min 59 sec(*)	7 h 31 min	[Pentium-4, 3.4GHz / (*)3.0GHz]

Improved Algorithm

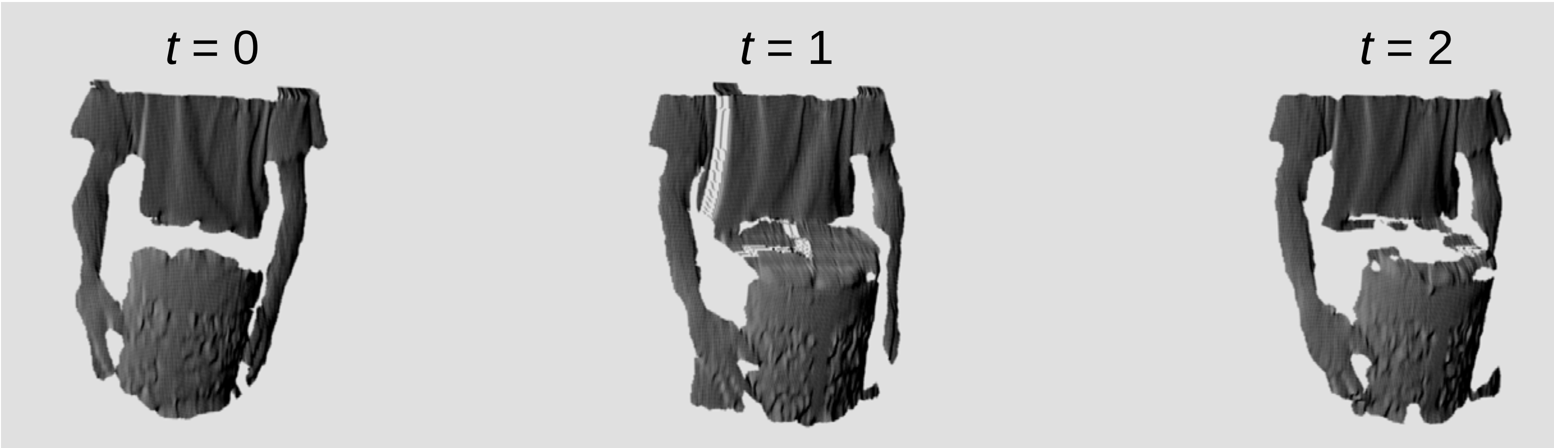
Urshape Factorization

Improved Version

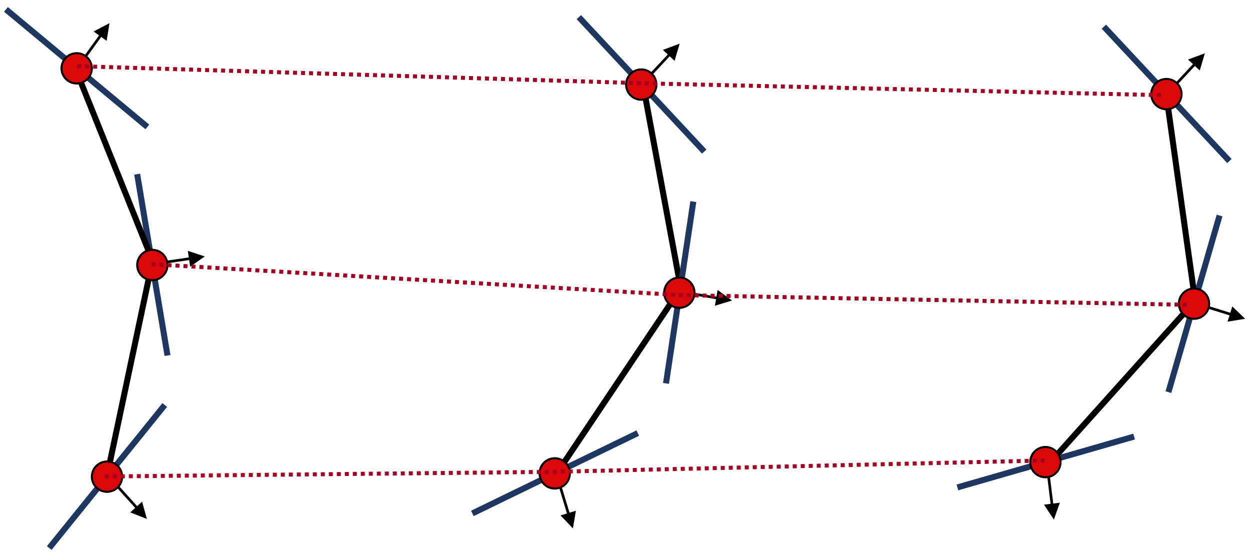
Factorization Model:

- Solving for the geometry in every frame wastes resources
- Store one urshape and a deformation field
  - High resolution geometry
  - Low resolution deformation (adaptive)
- Less memory, faster, and much more stable
- Streaming computation (constant working set)

We have so far...

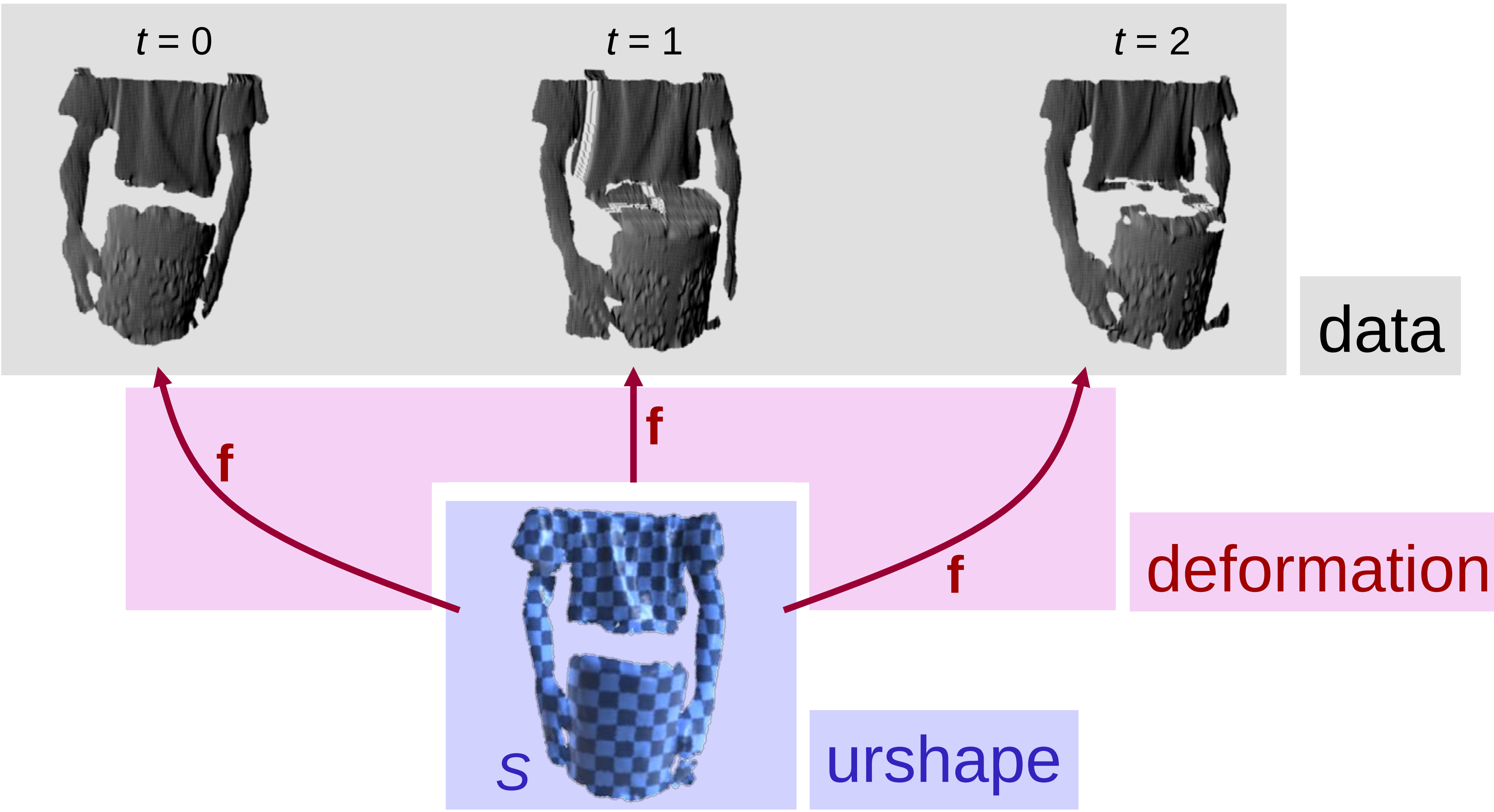


data



trajectories

New: Factorization



Components

Variational Model

- Given an initial estimate,
  improve *urshape* and *deformation*

Numerical Discretization

- *Shape*
- *Deformation*

Domain Assembly

- Getting an initial estimate
- *Urshape* assembly

Components

Variational Model

- Given an initial estimate, improve *urshape* and *deformation*

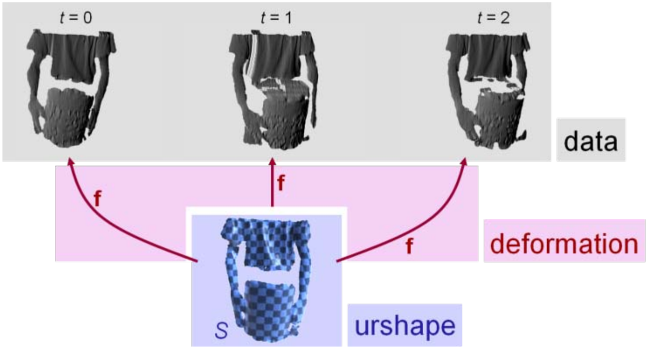
Numerical Discretization

- *Shape*
- *Deformation*

Domain Assembly

- Getting an initial estimate
- *Urshape* assembly

Energy Minimization



Energy Function

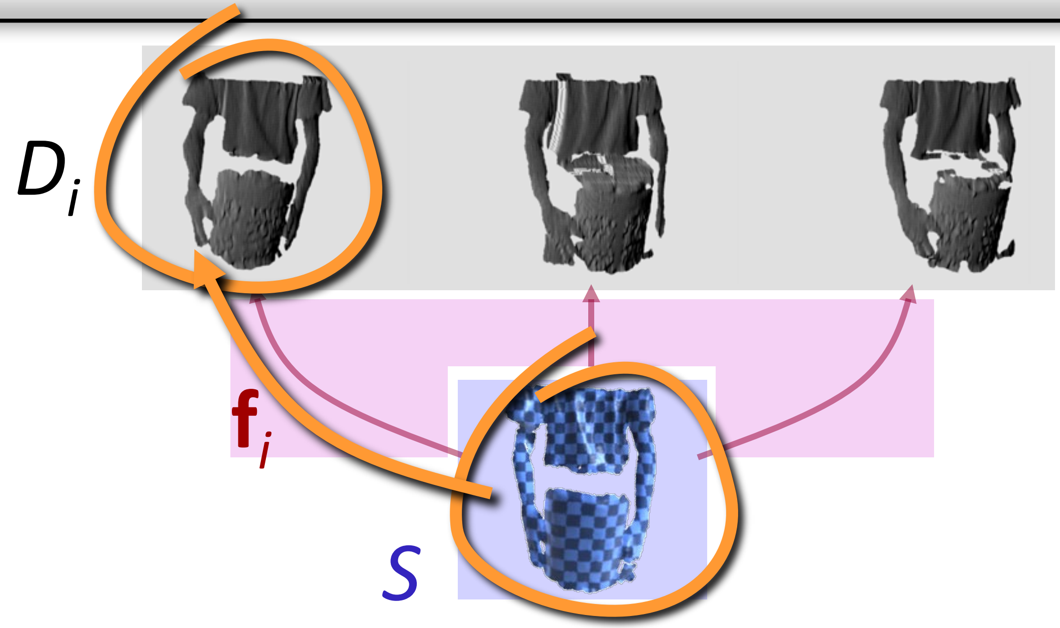
$E(\mathbf{f}, S) = E_{data} + E_{deform} + E_{smooth}$

Components

- $E_{data}(\mathbf{f}, S)$– data fitting
- $E_{deform}(\mathbf{f})$ – elastic deformation, smooth trajectory
- $E_{smooth}(S)$– smooth surface

Optimize S, $\mathbf{f}$ alternatingly

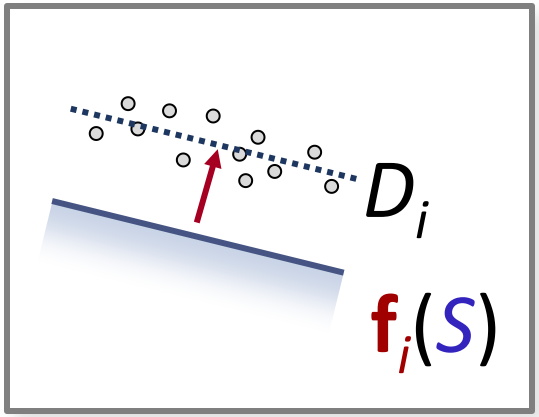
Data Fitting



$E_{data}(\mathbf{f}, S)$

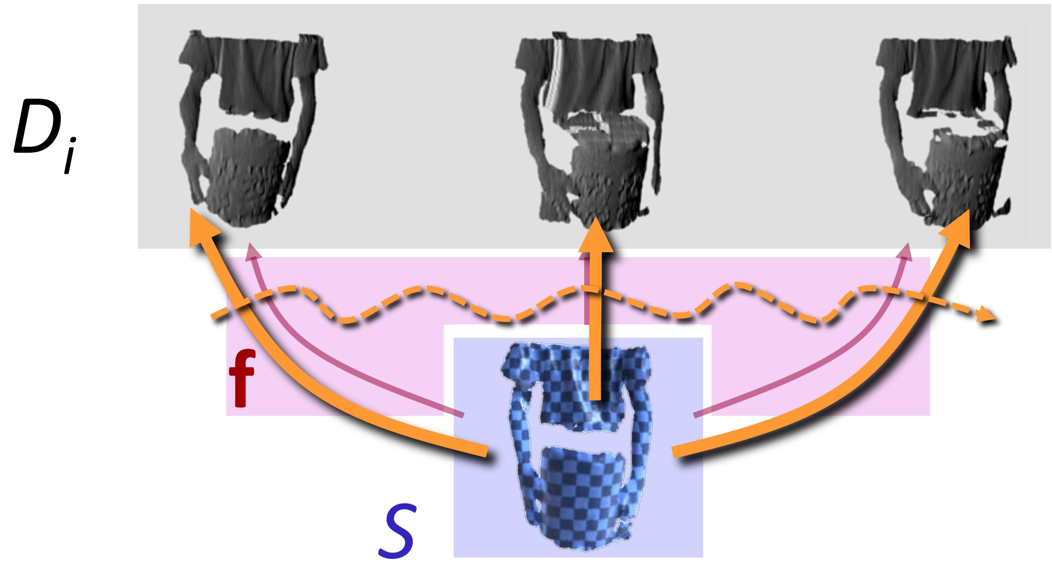
Data fitting

- Necessary: $\mathbf{f}_i(S) \approx D_i$
- Truncated squared distance function (point-to-plane)



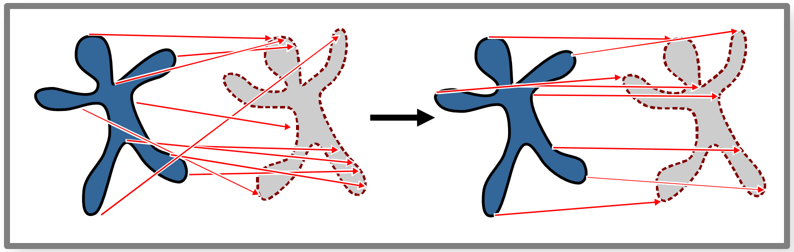
Elastic Deformation Energy

$E_{deform}(\mathbf{f})$

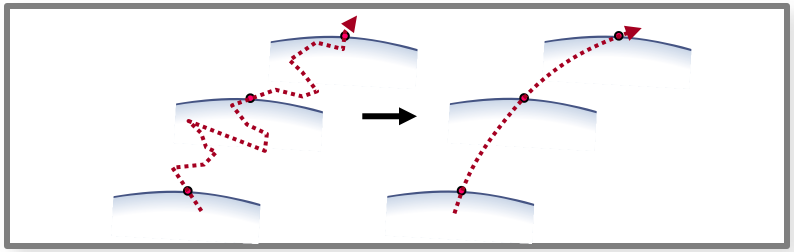


Regularization

- Elastic energy

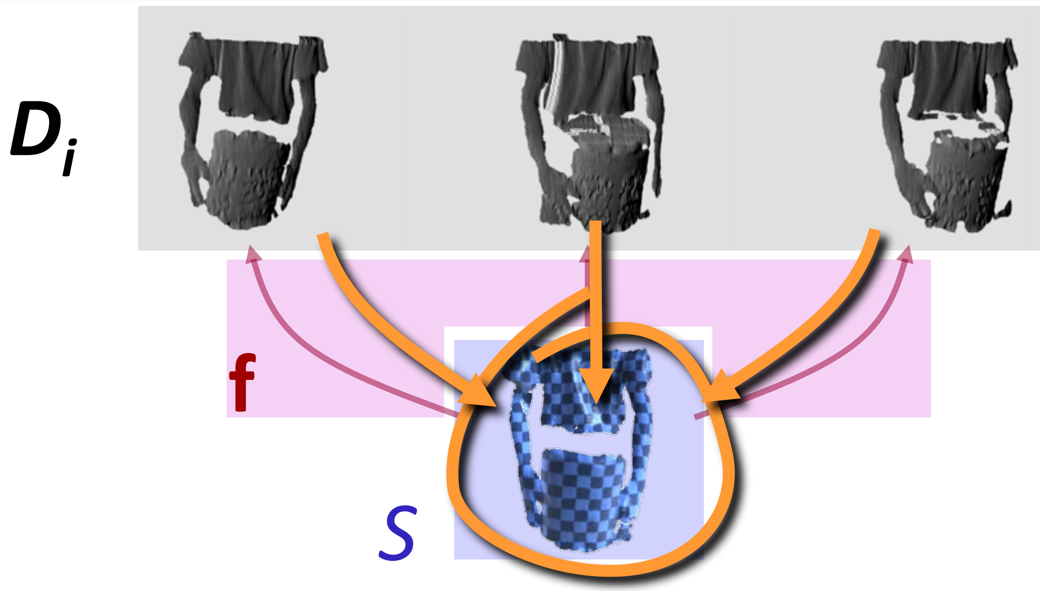


- Smooth trajectories



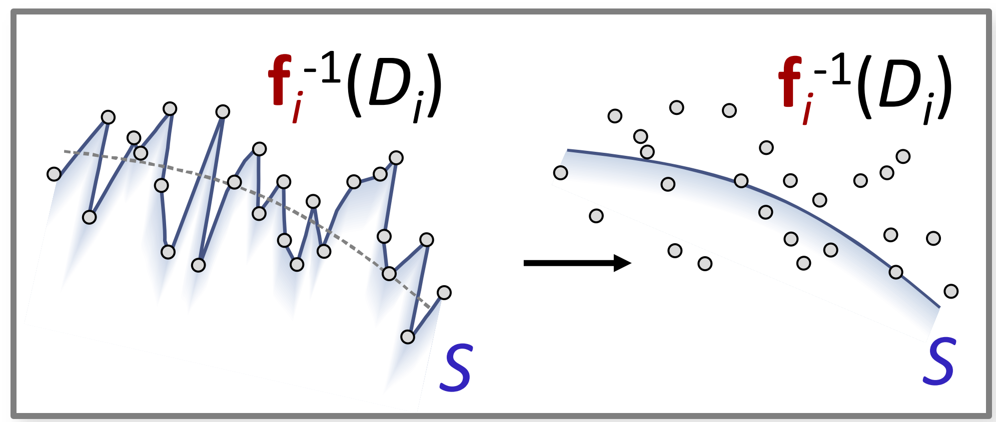
Surface Reconstruction

$E_{smooth}(S)$

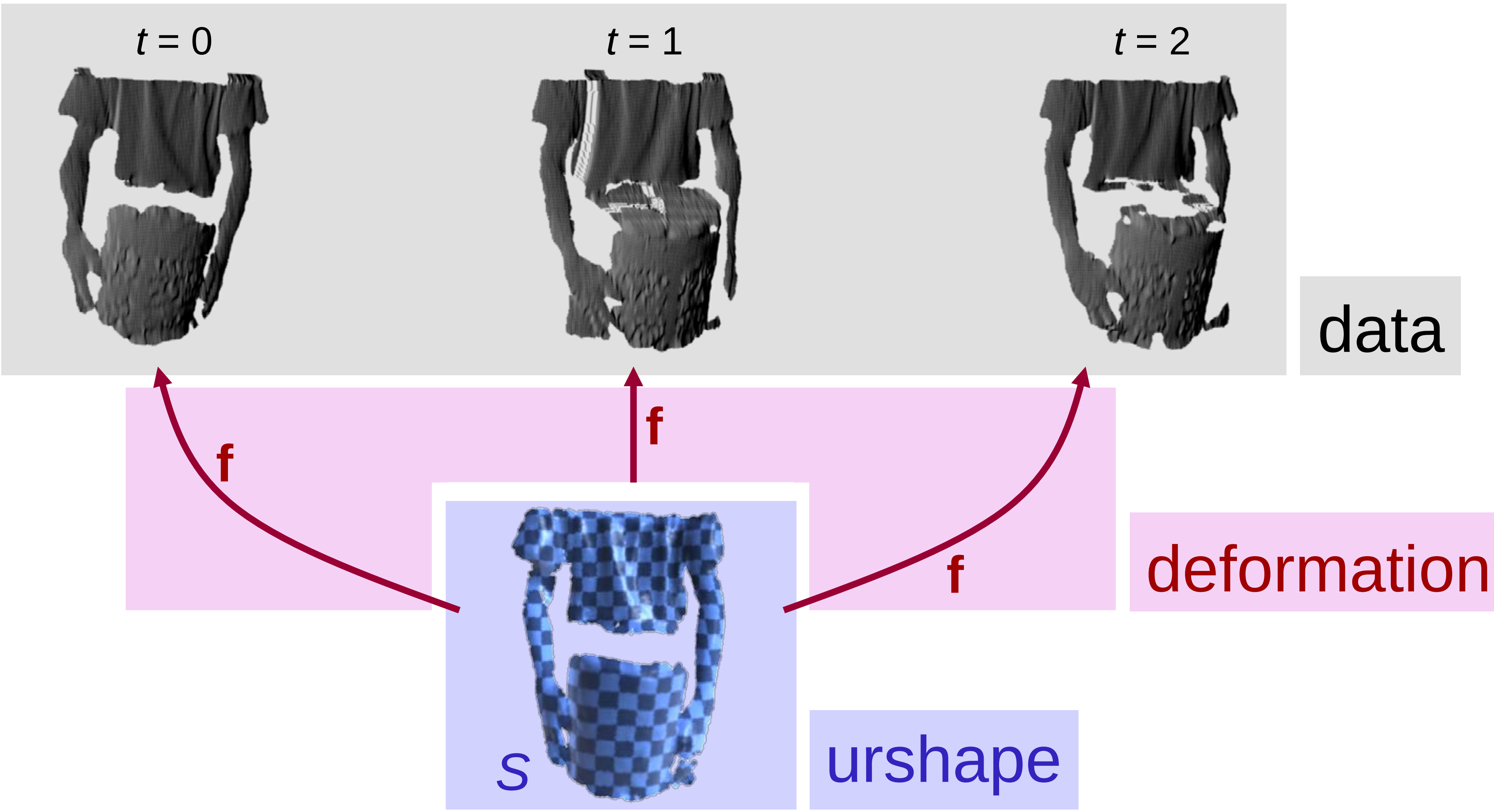


Data fitting

- Smooth surface
- Fitting to noisy data



Factorization



Components

Variational Model

- Given an initial estimate,
  improve *urshape* and *deformation*

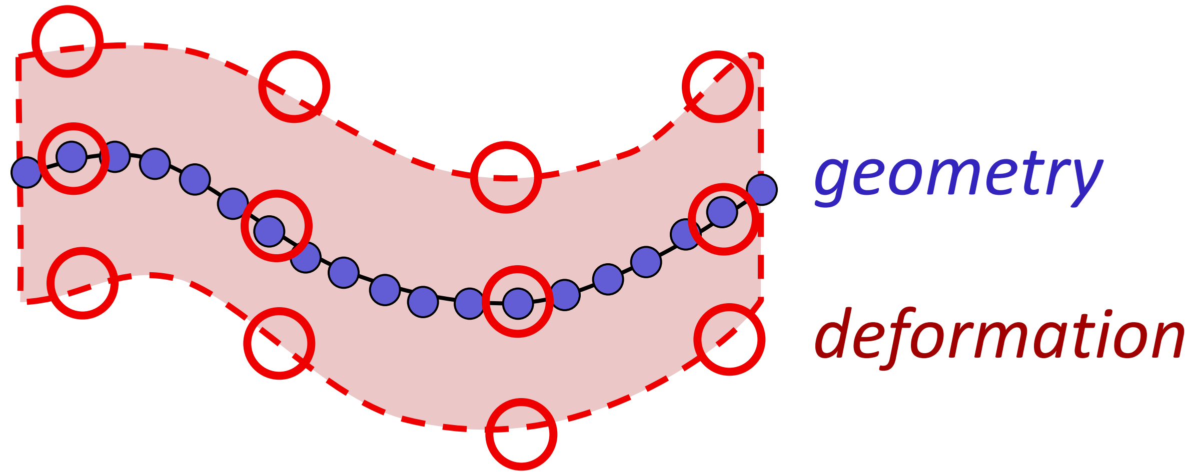
Numerical Discretization

- *Shape*
- *Deformation*

Domain Assembly

- Getting an initial estimate
- *Urshape* assembly

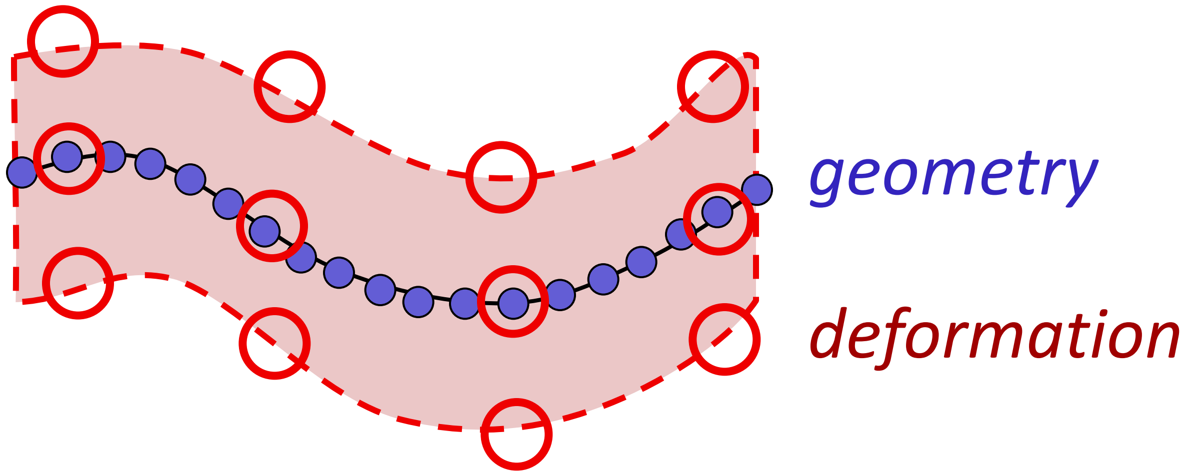
Discretization



Sampling:

- Full resolution *geometry*
- Subsample *deformation*

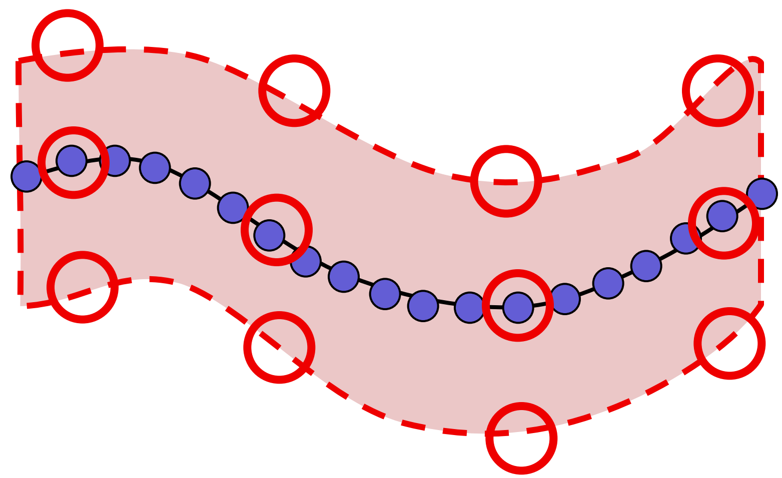
Discretization



Sampling:

- Full resolution *geometry*
  - High frequency
- Subsample *deformation*
  - Low frequency

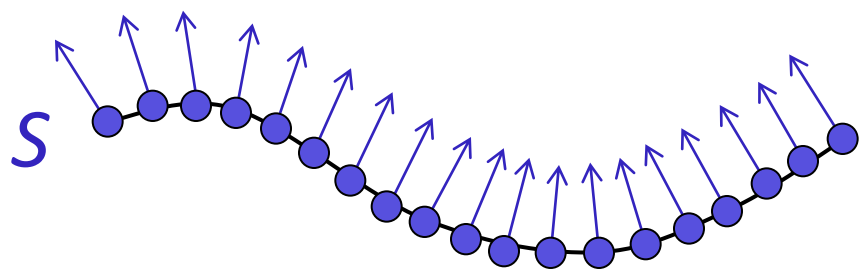
Discretization



Sampling:

- Full resolution *geometry*
  - High frequency, stored once
- Subsample *deformation*
  - Low frequency, all frames ⇒ more costly

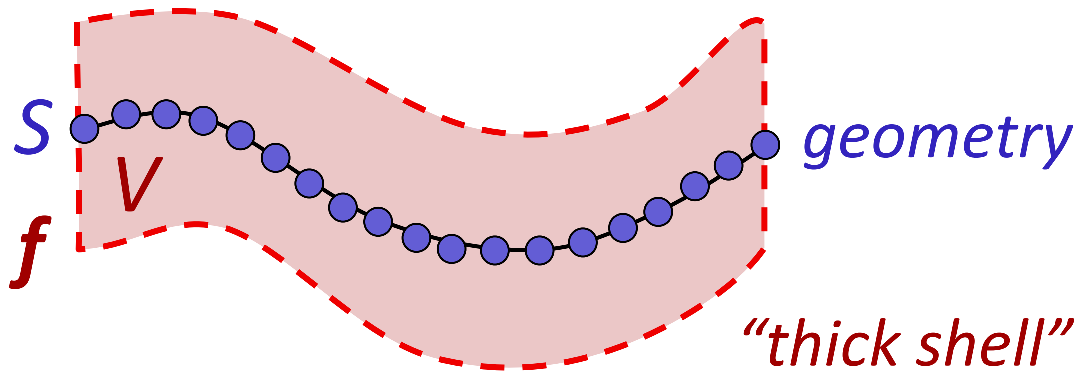
Shape Representation



Shape Representation:

- Graph of *surfels* (point + normal + local connectivity)
- E_{smooth} – neighboring planes should be similar
- Same as before...

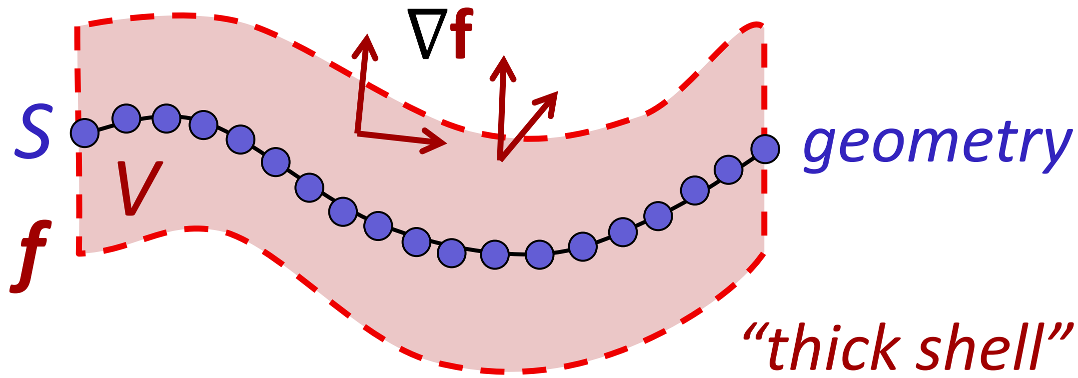
Deformation



Volumetric Deformation Model

- Surfaces embedded in "stiff" volumes
- Easier to handle than "thin-shell models"
- General – works for non-manifold data

Deformation



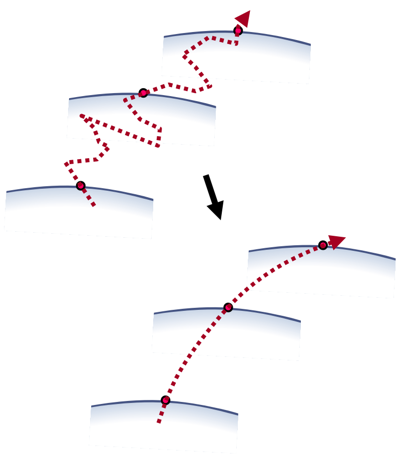
Deformation Energy

- Keep deformation gradients $\nabla \mathbf{f}$ as-rigid-as-possible
- This means: $\nabla \mathbf{f}^T \nabla \mathbf{f} = \mathbf{I}$
- Minimize: $E_{deform} = \int_T \int_V || \nabla \mathbf{f}(\mathbf{x},t)^T \nabla \mathbf{f}(\mathbf{x},t) - \mathbf{I} ||^2 \, d\mathbf{x} \, dt$

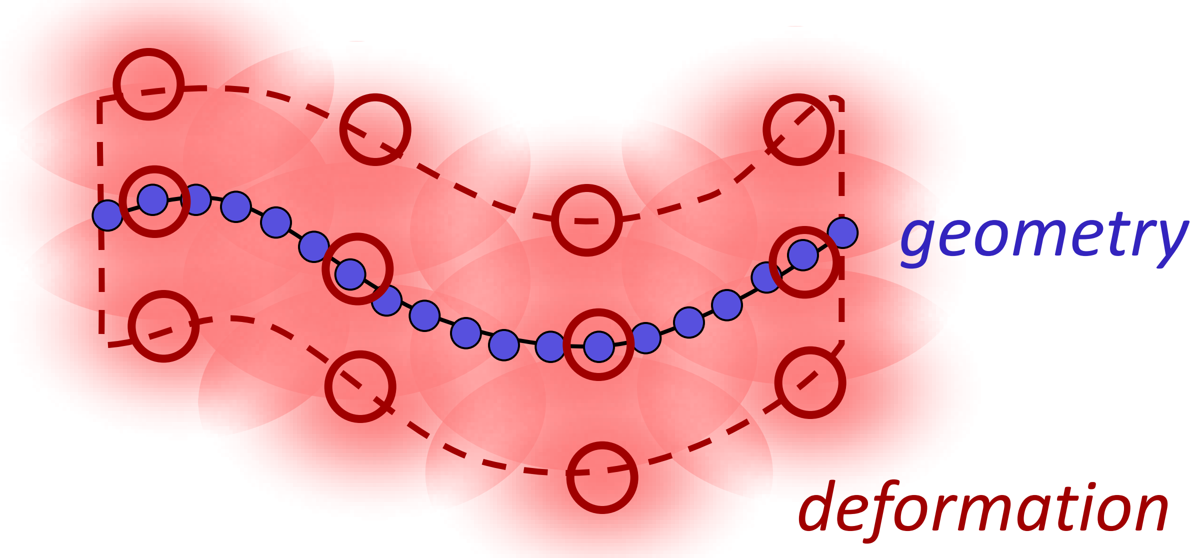
Additional Terms

More Regularization

- Volume preservation: $E_{vol} = \int_T \int_V ||\det(\nabla \mathbf{f}) - 1||^2$
  - Stability
- Acceleration: $E_{acc} = \int_T \int_V ||\partial_t^2 \ \mathbf{f}||^2$
  - Smooth trajectories
- Velocity (weak): $E_{vel} = \int_T \int_V ||\partial_t \ \mathbf{f}||^2$
  - Damping



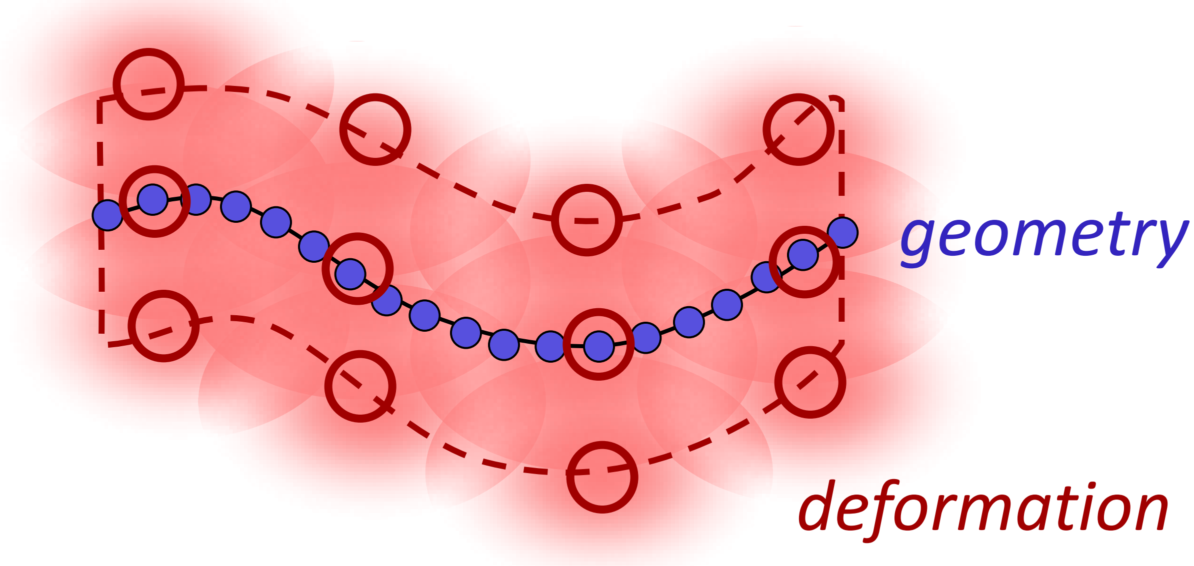
Discretization



How to represent the deformation?

- Goal: efficiency
- Finite basis:
  As few basis functions as possible

Discretization



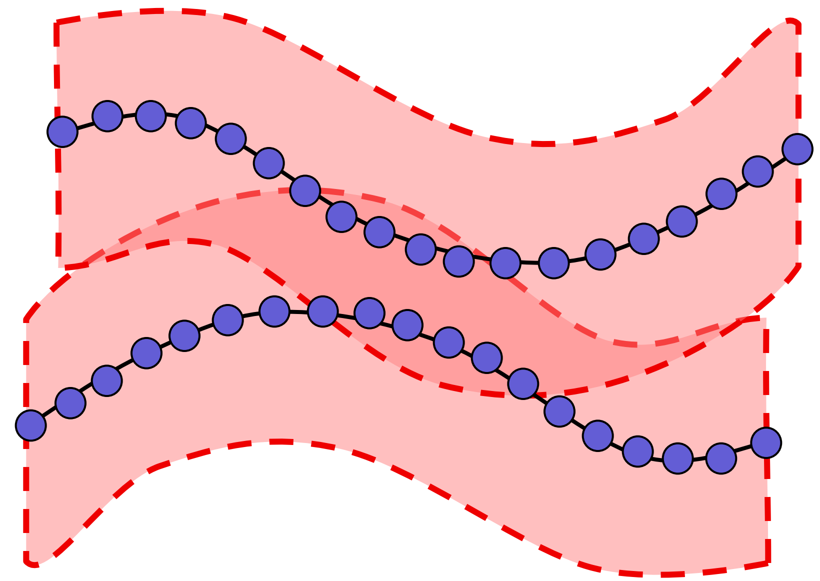
Meshless finite elements

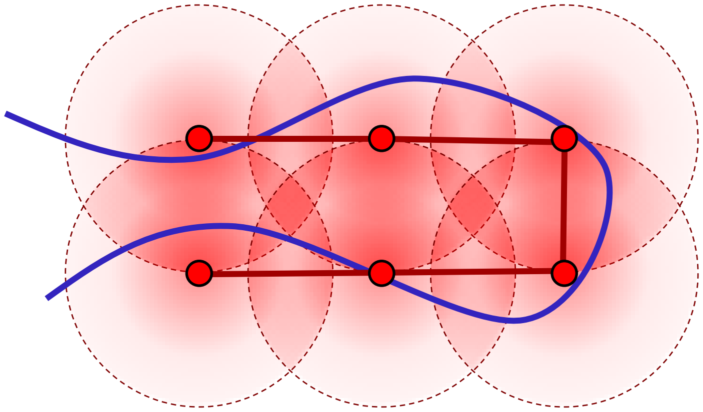
- Partition of unity, smoothness
- Linear precision
- Adaptive sampling is easy

Meshless Finite Elements

Topology:

- Separate deformation nodes for disconnected pieces
- Need to ensure
  - Consistency
  - Continuity
- Euclidean / intrinsic distance-based coupling rule
  - See references for details

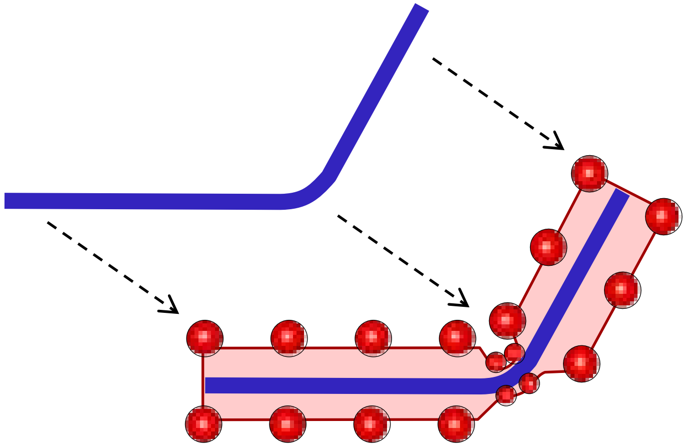




Adaptive Sampling

Adaptive Sampling

- Bending areas
  - Decrease rigidity
  - Decrease thickness
  - Increase sampling density
- Detecting bending areas: residuals over many frames



Components

Variational Model

- Given an initial estimate,
  improve *urshape* and *deformation*

Numerical Discretization

- *Deformation*
- *Shape*

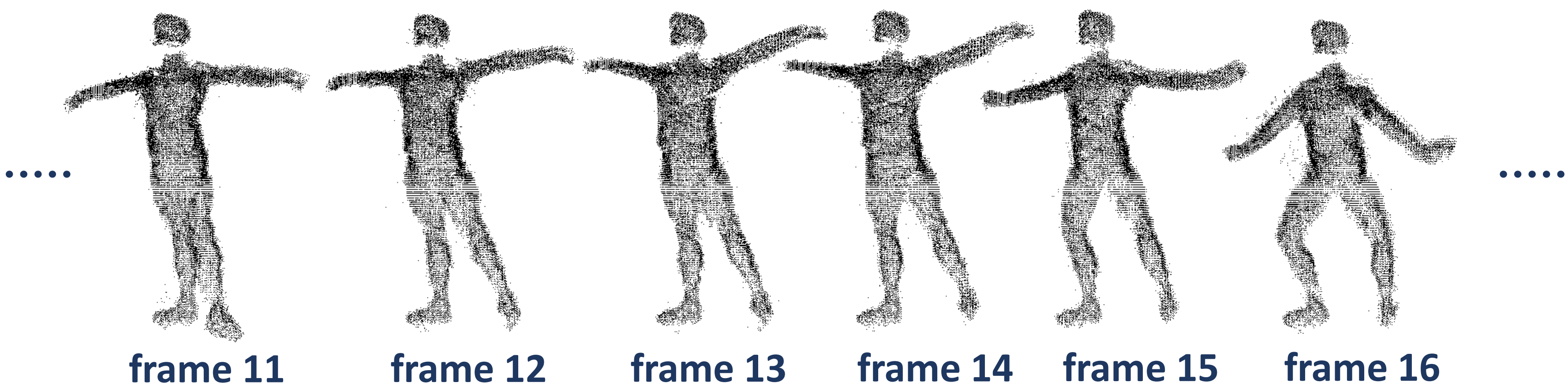
Domain Assembly

- Getting an initial estimate
- *Urshape* assembly

Urshape Assembly

Adjacent frames are similar

- Solve for frame pairs first
- Assemble urshape step-by-step



[data set courtesy of C. Theobald, MPC-VCC]

Hierarchical Merging

data

f(*S*)

f

S

Hierarchical Merging

data

f(S)

f

S

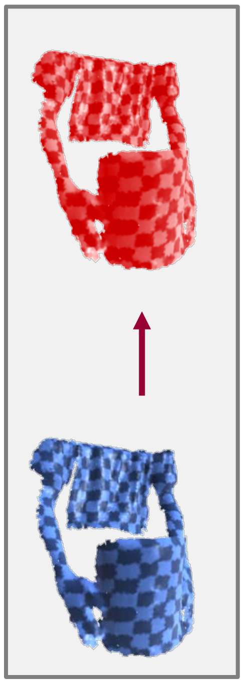
Initial Urshapes

data

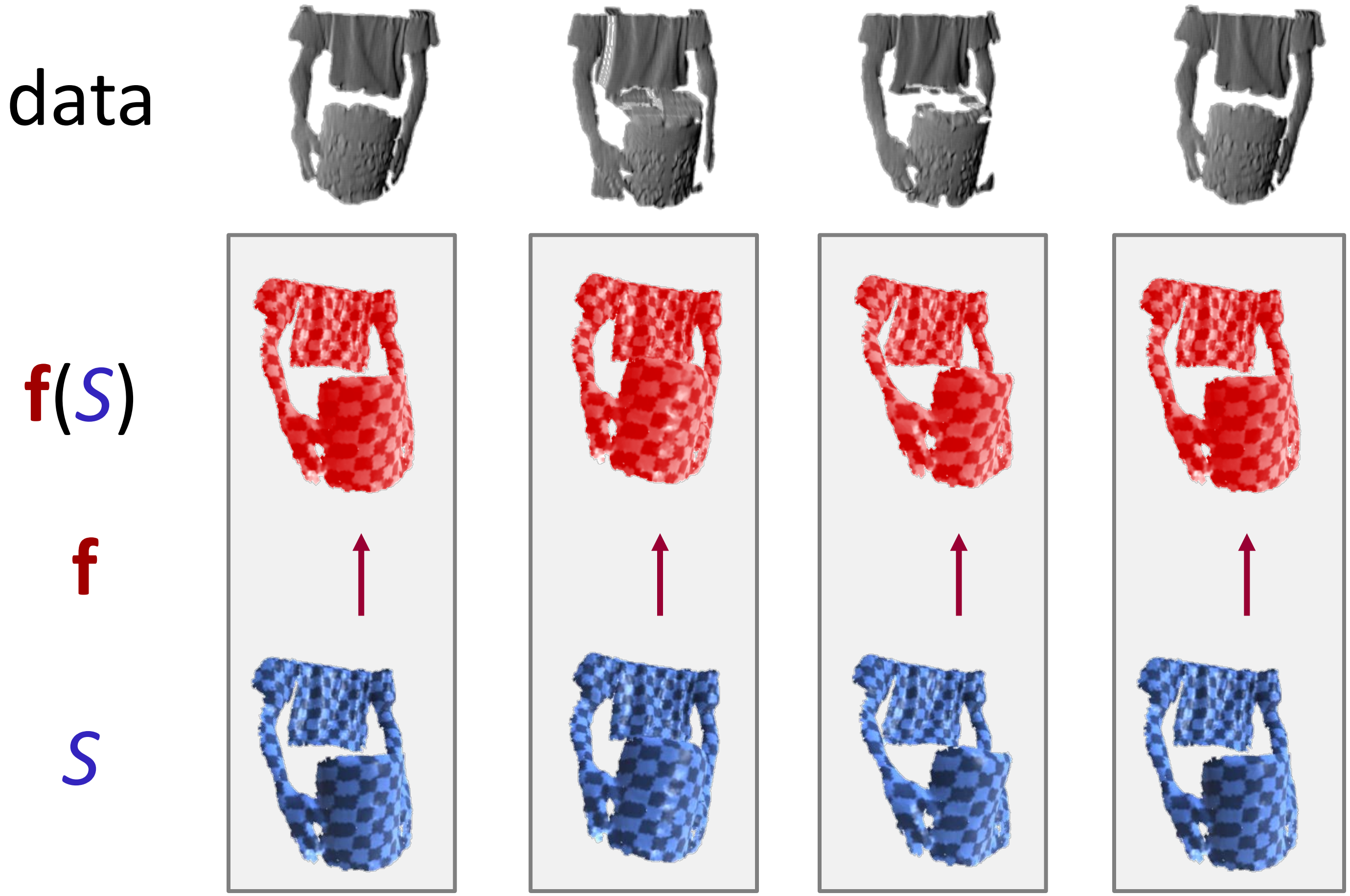
$f(S)$

f

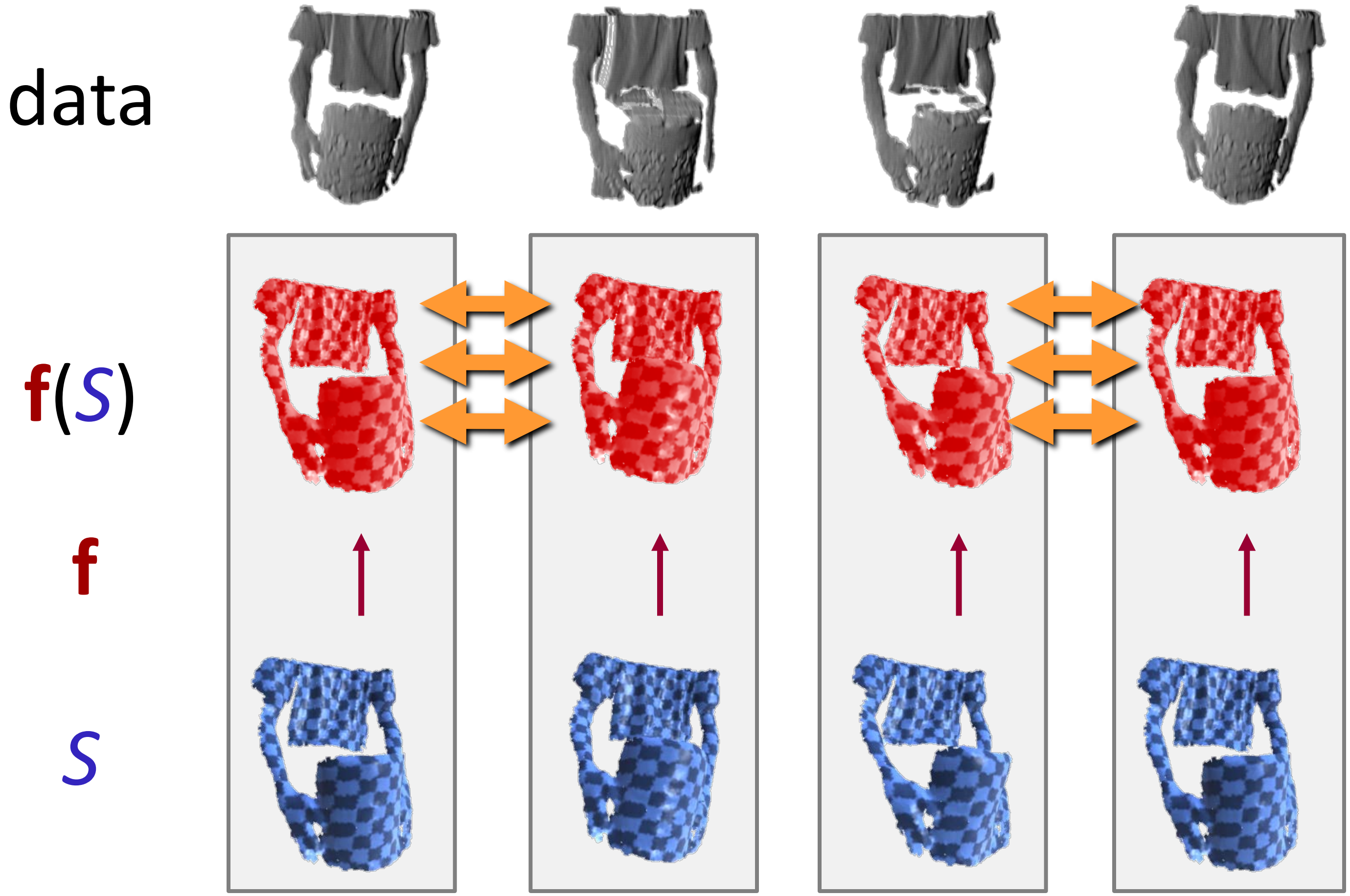
S



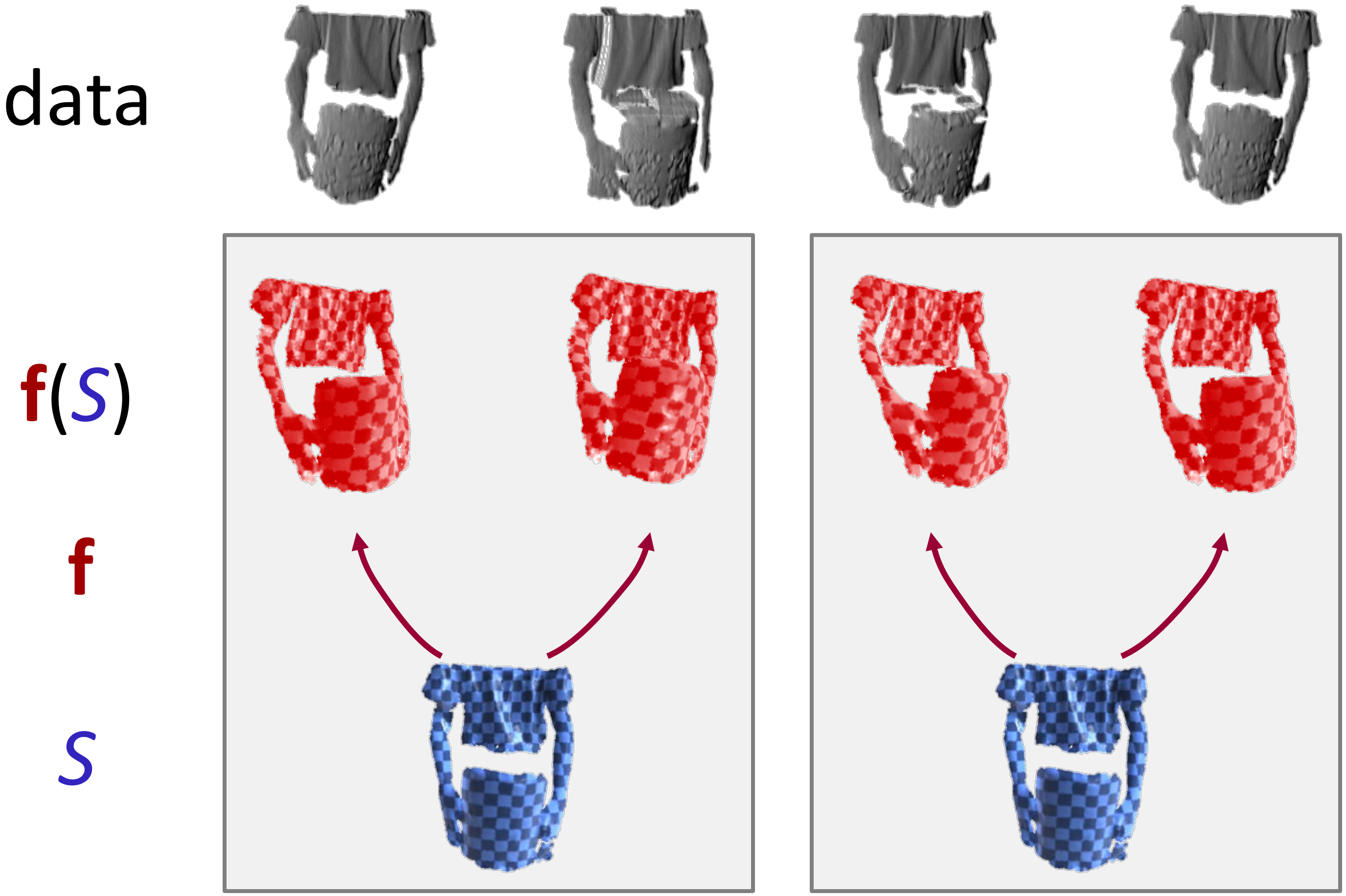
Initial Urshapes



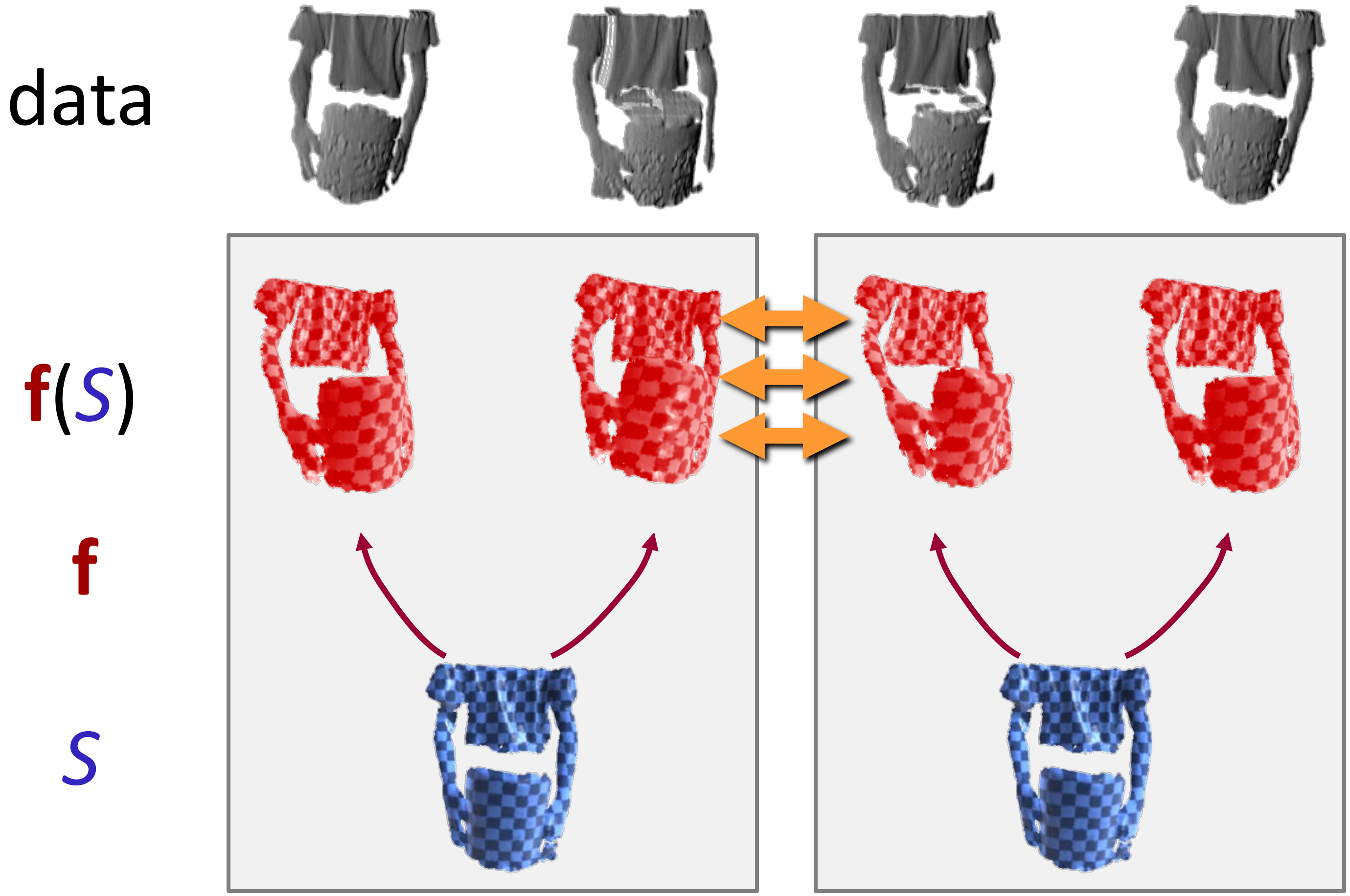
Alignment



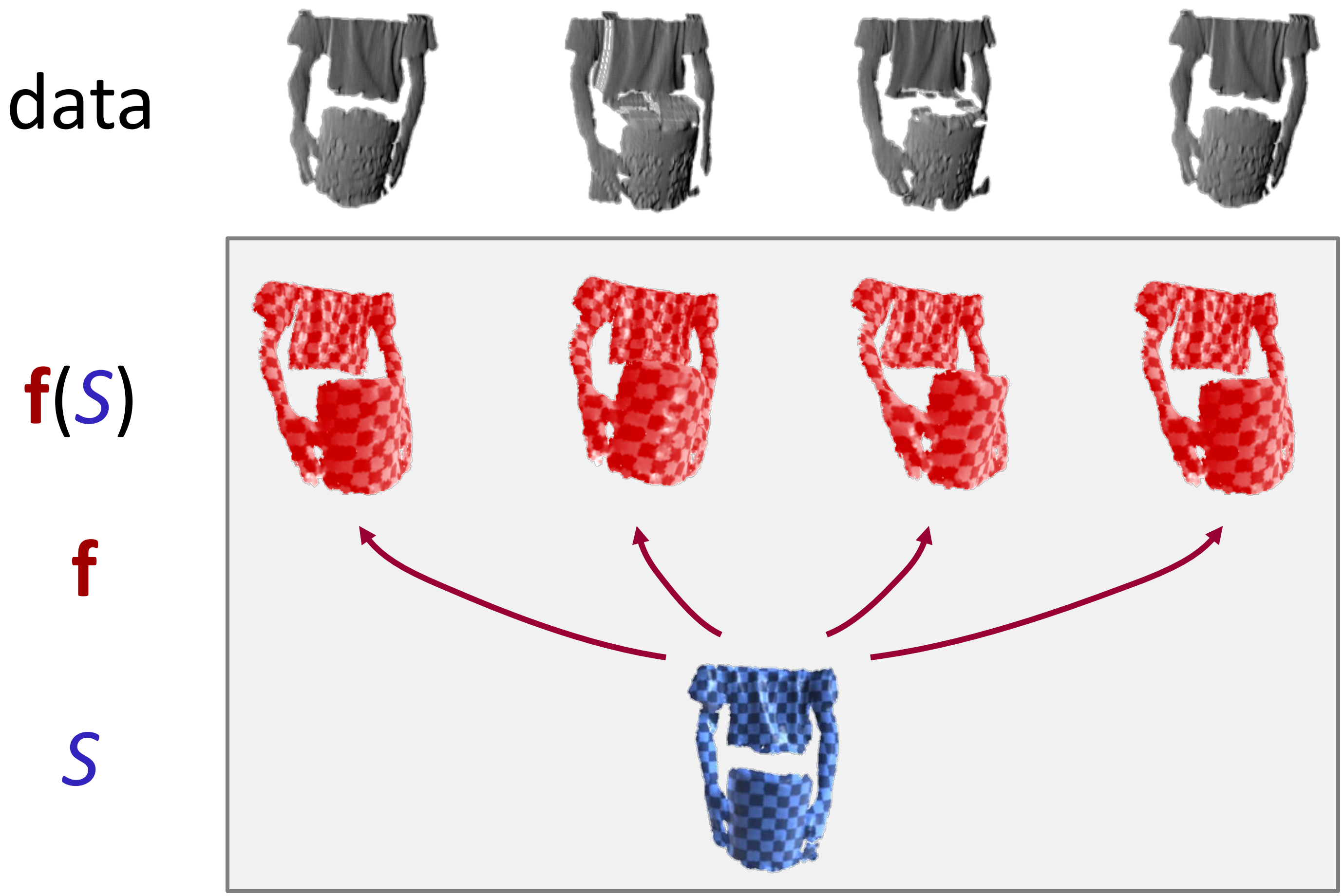
Align & Optimize



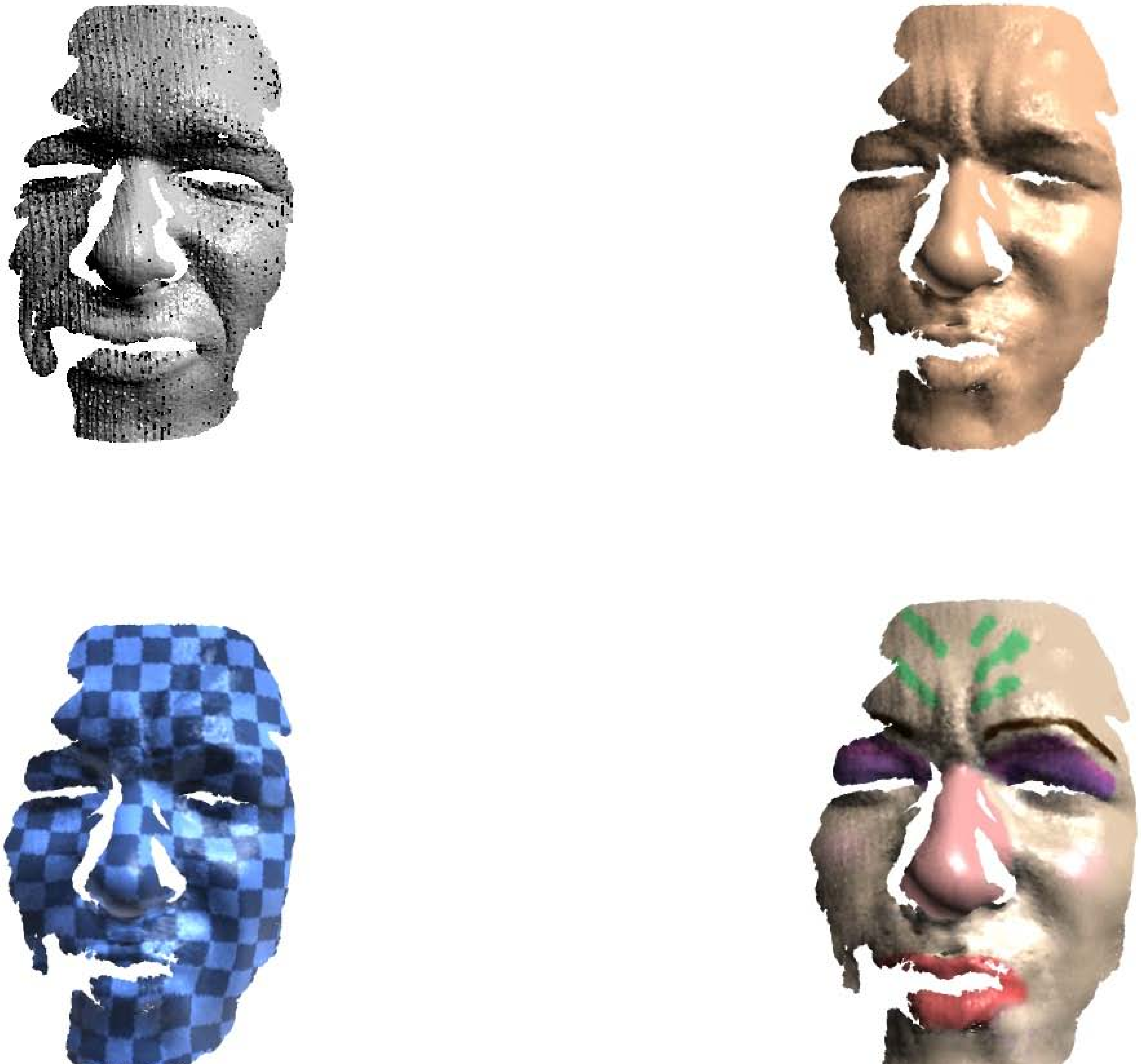
Hierarchical Alignment



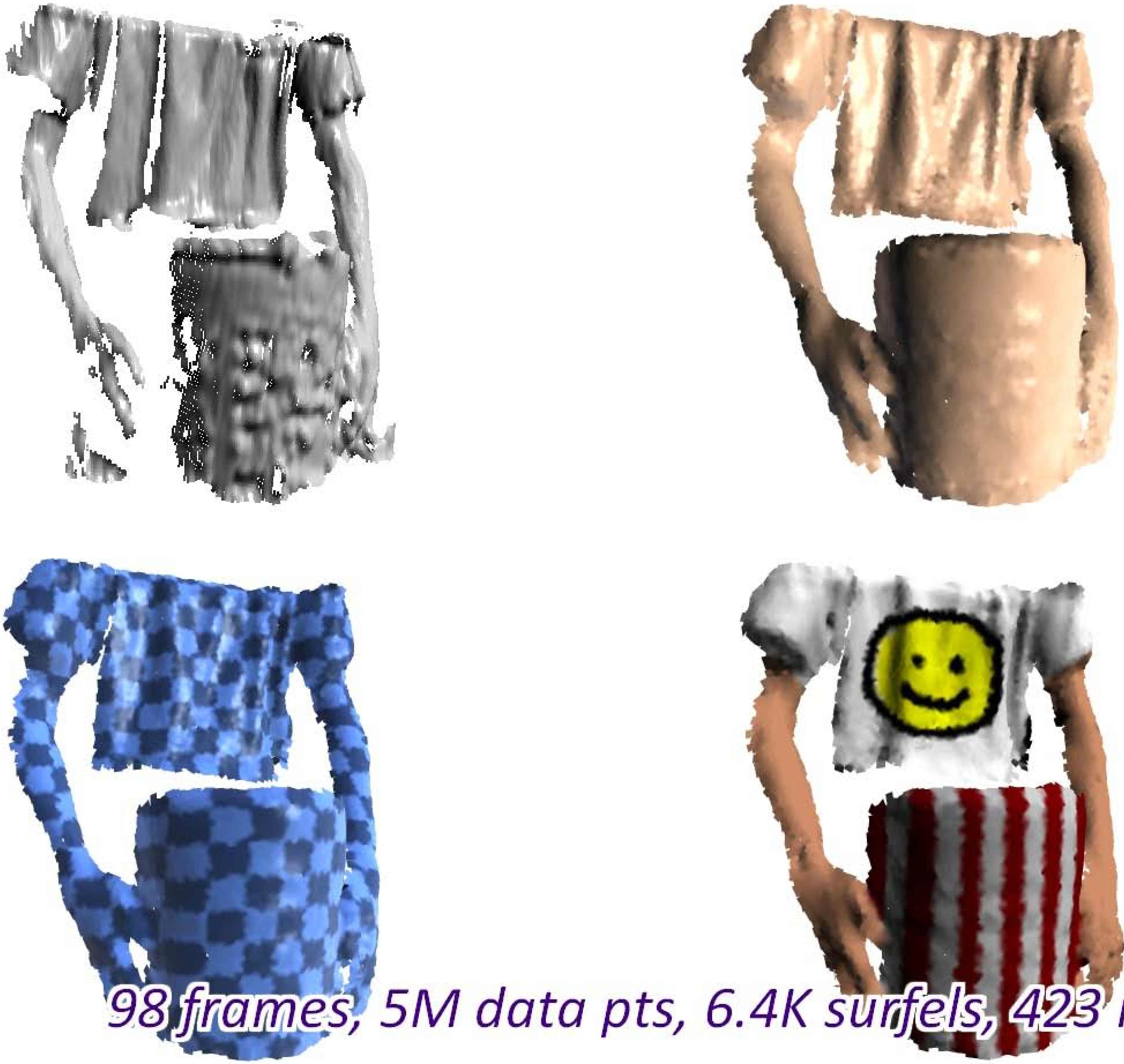
Hierarchical Alignment



Results



79 frames, 24M data pts, 21K surfels, 315 nodes



98 frames, 5M data pts, 6.4K surfels, 423 nodes

120 frames,

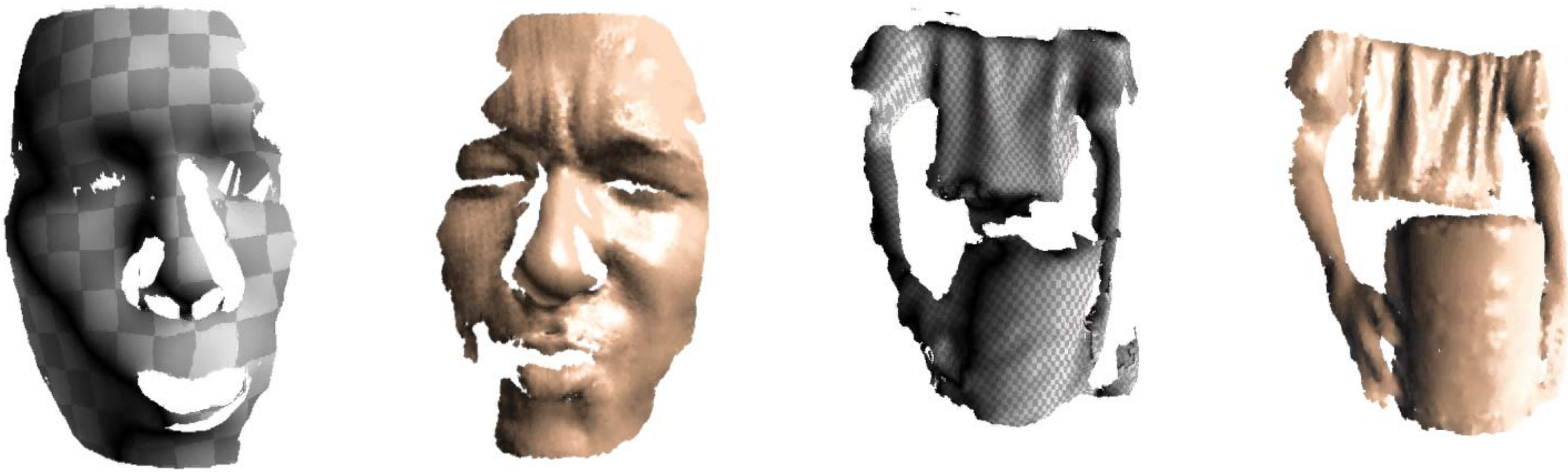
30M data pts,

17K surfels,

1,939 nodes

34 frames,
4M data pts,
23K surfels,
414 nodes

Quality Improvement



old version **new result** old version **new result**