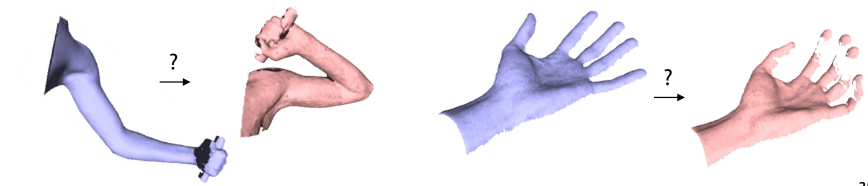
Automatic Registration for Articulated Shapes

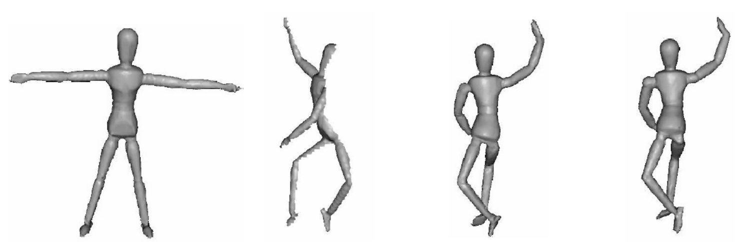
Problem Statement

- Solve pairwise registration problem
  - Develop robust method independent of initial pose
  - Do not require markers or a template
- Contributions:
  - Useful for initialization: used as preprocessing step
  - Focus on registration: does not solve for a reduced motion model



Related Work

- Correlated correspondence algorithm, requires a template (Anguelov et al. 2004)

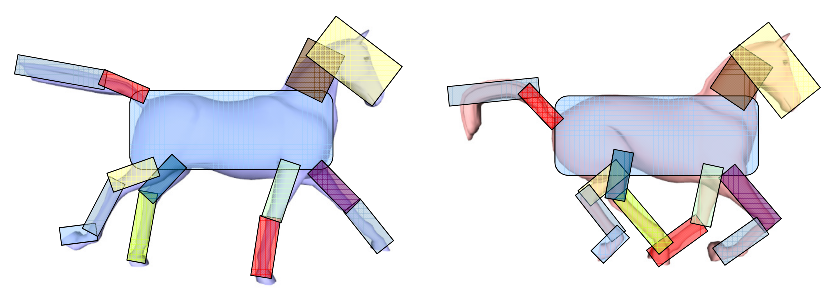


Template Model | Partial Example | Registered result | Ground Truth

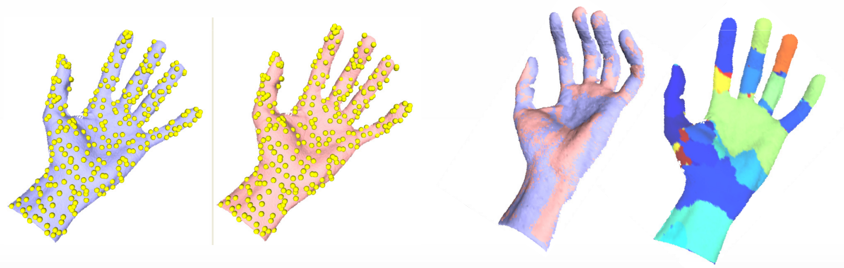
(from Anguelov et al. 2004)

Algorithm Overview

- Articulated motion → small set of transformations
- **Predetermine** a set of transformations describing the motion
- **Optimize assignment** of transformations to the points



Algorithm Description

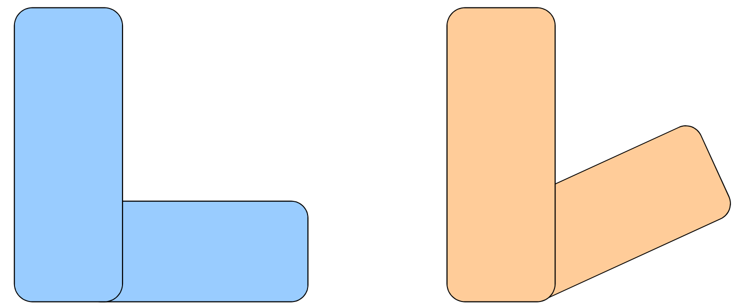


Motion Sampling

Global Motion Optimization

Motion Sampling Illustration

- Find transformations that move parts of the source to parts of the target

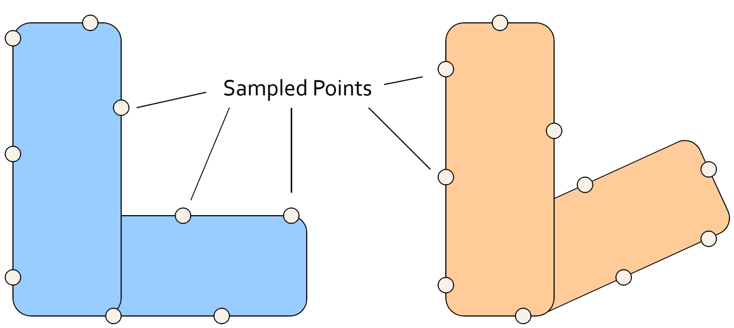


Source Shape

Target Shape

Motion Sampling Illustration

- Find transformations that move parts of the source to parts of the target

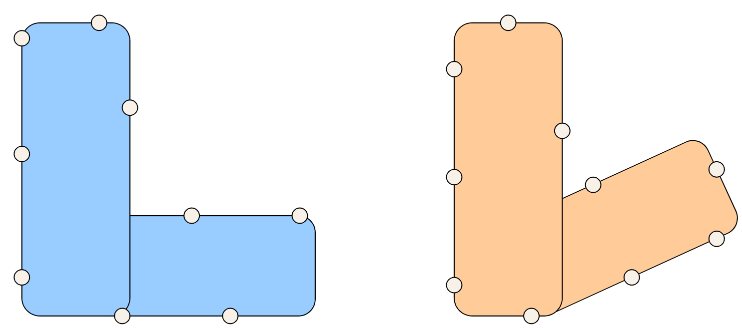


Source Shape

Target Shape

Motion Sampling Illustration

- Find transformations that move parts of the source to parts of the target

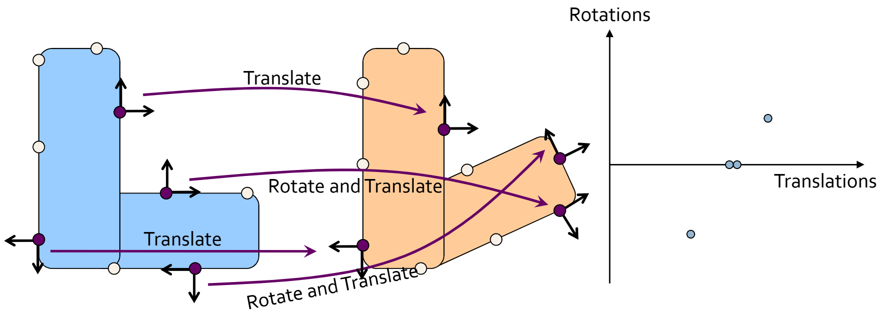


Source Shape

Target Shape

Motion Sampling Illustration

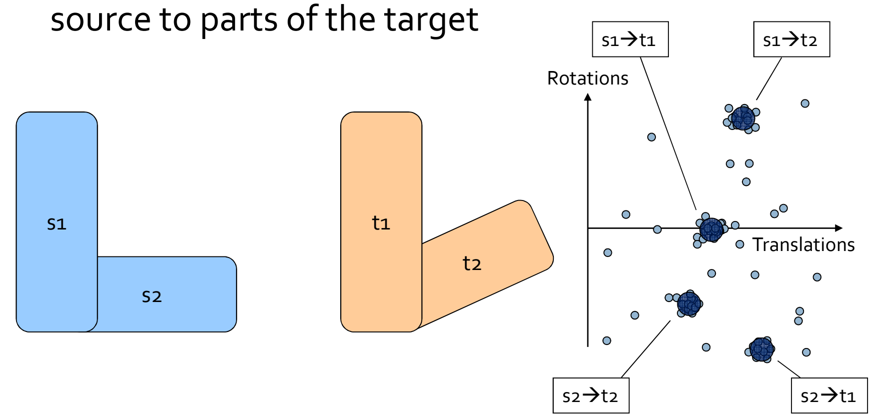
- Find transformations that move parts of the source to parts of the target



Source Shape Target Shape Transformation Space

Motion Sampling Illustration

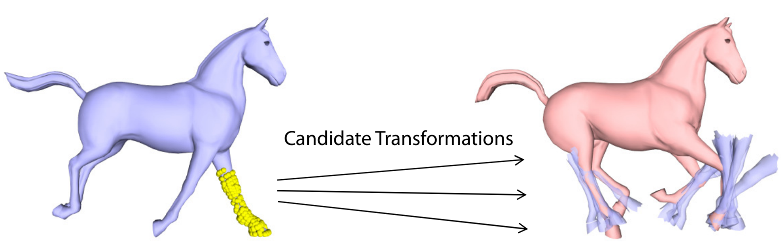
- Find transformations that move parts of the source to parts of the target



Source Shape Target Shape Transformation Space

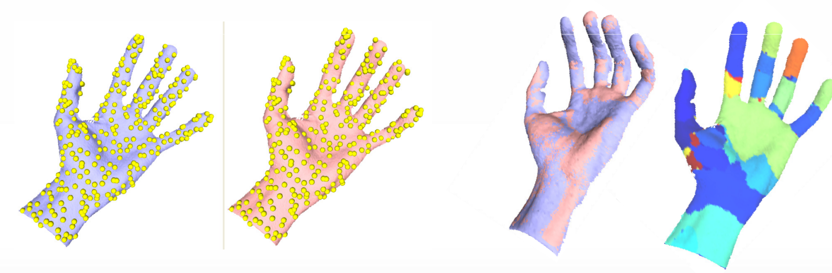
Limitations of Motion Sampling

- *Final Output:* finite set of rigid transformations
- If there are multiple similar parts
  - Does not figure out the correct part
  - Disambiguate in the optimization step



Source with Selected Region Visualized Transformations

Algorithm Description



Motion Sampling **Global Motion Optimization**

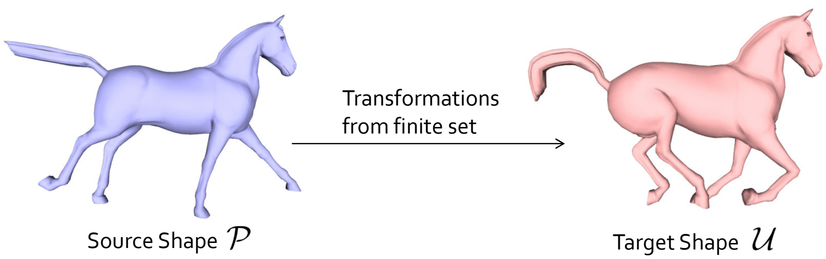
Global Motion Optimization

- Optimize an *assignment* from a finite set of transformations

argmin Data Cost + Smoothness Cost

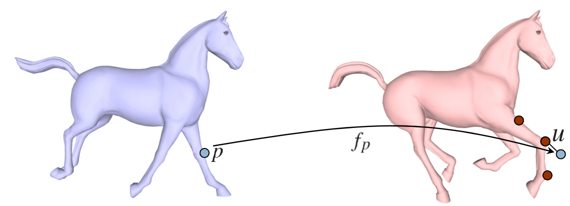
Assignment from a set of transformations

- A *discrete labelling problem* → Graph Cuts for optimization



Data Term

- Move all points as close as possible to the target
- How to measure distance to target?
  - Apply selected transformation f_p for all $p = f_p(p)$
  - Measure distance to closest point u in target

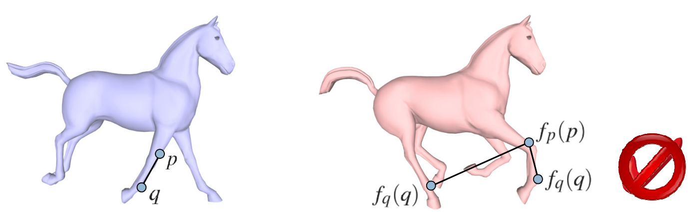


Smoothness Term

- Preserve edge length between neighboring points

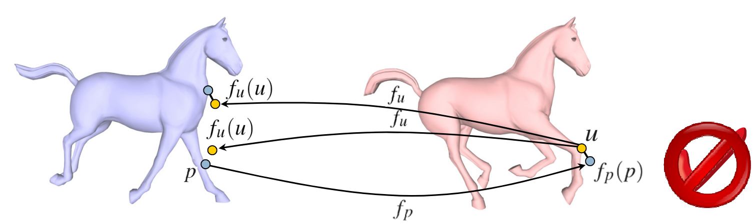
$$V(p,q,f_p,f_q) = \Big| \underbrace{\|p-q\|}_{\text{Original Length}} - \underbrace{\|f_p(p)-f_q(q)\|}_{\text{Transformed Length}} \Big|$$

- Disambiguates multiple possible mappings



Symmetric Cost Function

- Swapping source / target can give different results
  - Optimize assignment in both meshes (forward & backward)
  - Enforce consistent assignment: penalty when $f_p \neq f_u$



Optimization Using Graph Cuts

argmin *Assignment from a set of transformations* Data $_{Source}$ + Smoothness $_{Source}$ +
Data $_{Target}$ + Smoothness $_{Target}$ +
Symmetric Consistency $_{Source\ \&\ Target}$

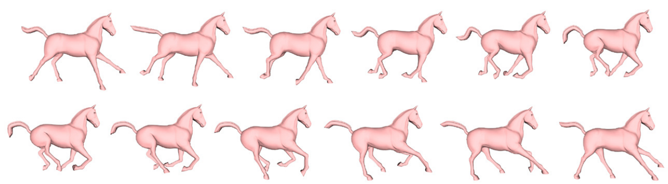
- Data and smoothness terms apply to both shapes
- Additional symmetric consistency term
- Weights to control relative influence of each term
- Use "graph cuts" to optimize assignment
  - [Boykov, Veksler & Zabih PAMI '01]

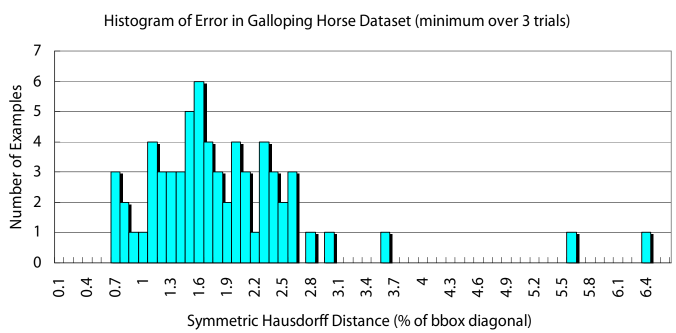
Results

Horse Dataset
Arm Dataset
Hand Dataset

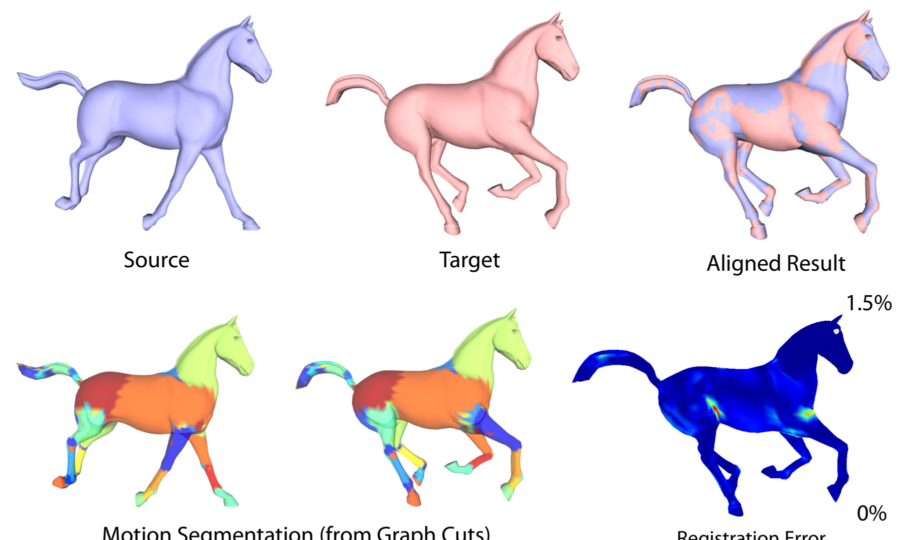
Horse Dataset Results

12 poses of galloping horse: total of 66 pairs, correct leg matched in 64 pairs

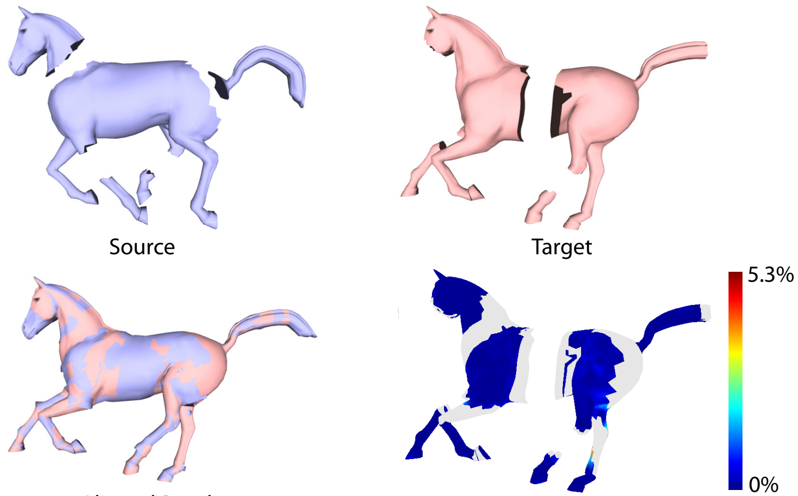




Synthetic Dataset Example



Synthetic Dataset w/ Holes



Source

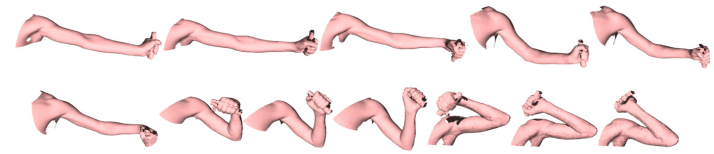
Target

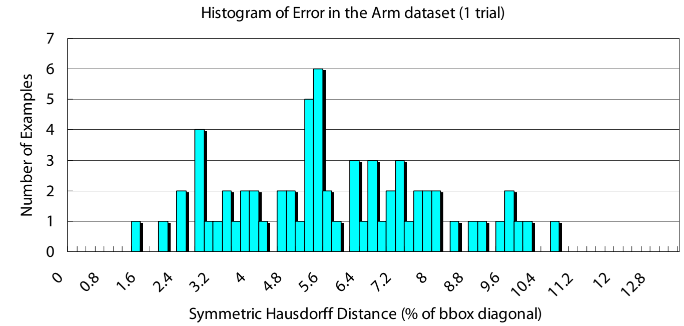
Aligned Result

Distance (from Target) to the closest point (% bounding box diagonal)

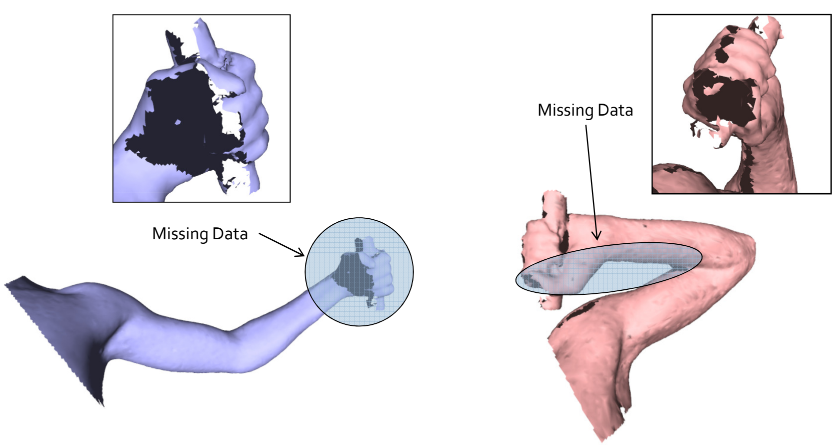
Arm Dataset Results

12 poses of arm scans: total of 66 pairs, arm & hand orientation matched in all pairs





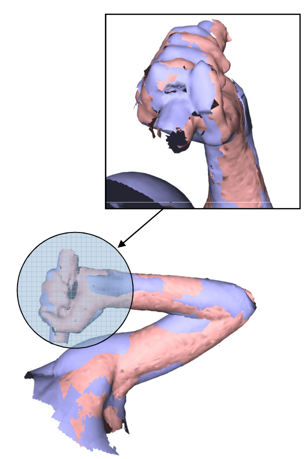
Arm Dataset Example

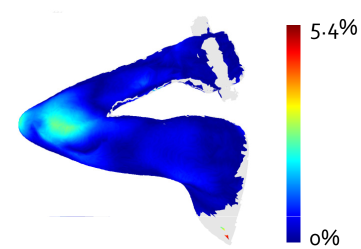


Source

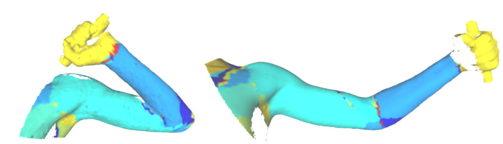
Noisy Target

Arm Dataset Example





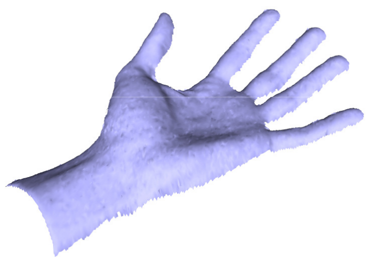
Distance (from Target) to the closest point (% bounding box diagonal)



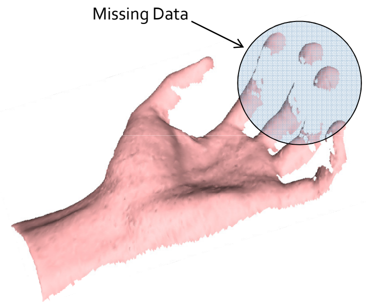
Aligned Result

Motion Segmentation

Hand Dataset Example

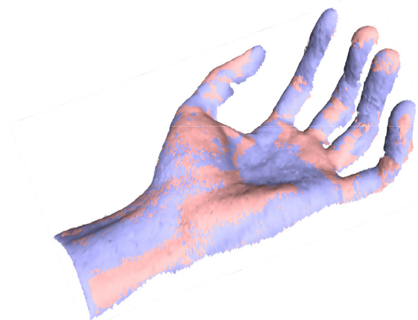


Source

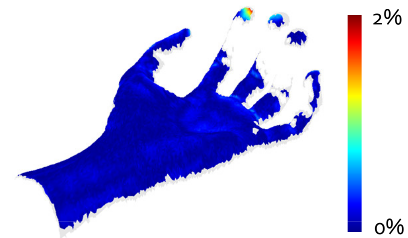


Target

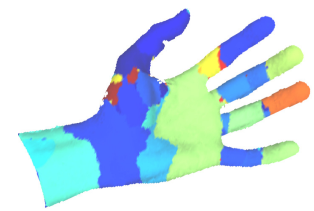
Hand Dataset Example



Aligned Result



Distance (from Target) to the closest point (% bounding box diagonal)



Motion Segmentation

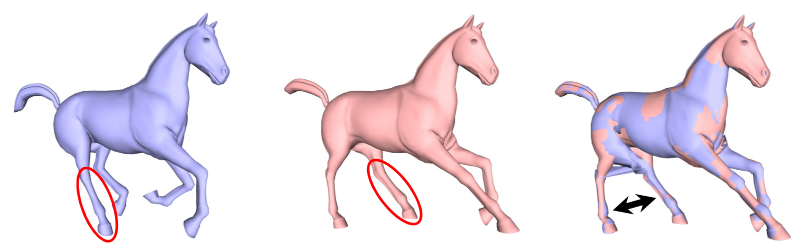
Performance

Dataset	#Points	# Labels	Matching	Clustering	Pruning	Graph Cuts
Horse	8431	1500	2.1 min	3.0 sec	(skip) 1.6 sec	1.1 hr
Arm	11865	1000	55.0 sec	0.9 sec	12.4 min	1.2 hr
Hand (Front)	8339	1500	14.5 sec	0.7 sec	7.4 min	1.2 hr
Hand (Back)	6773	1500	17.3 sec	0.9 sec	9.4 min	1.6 hr

- Graph cuts optimization is most time-consuming step
  - Symmetric optimization doubles variable count
  - Symmetric consistency term introduces many edges

Limitations

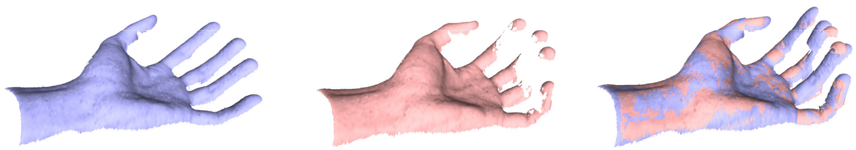
- Errors in registration
  - Trade-off between data and smoothness costs
    - Data weight too high → May break smoothness
    - Smoothness weight too high → Prefer bad alignment



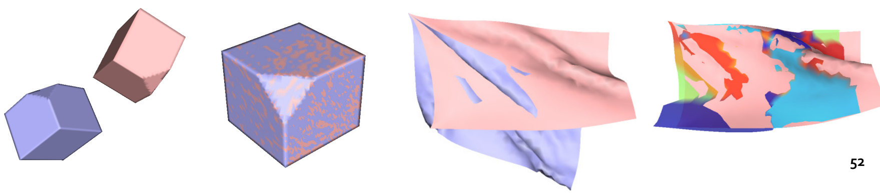
Source Target Registration

Limitations

- Errors in registration
  - Motion sampling: may fail to sample properly when too much missing data, non-rigid motion
  - Hard assignment of transformations



Source Target Registration



Conclusions

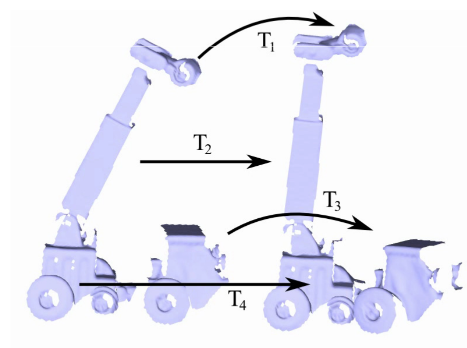
- Automatic method for registering articulated shapes
  - No template, markers, or manual segmentation needed
  - Explicitly sample a discrete set of motion
  - Optimize the assignment of transformations
  - Graph cut result gives intuitive segmentation

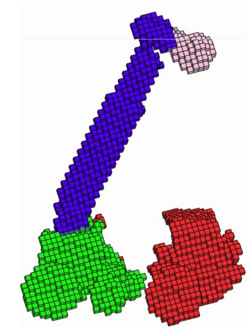
- Useful for obtaining a robust initialization of the registration
  - Does not provide an articulated motion model

Range Scan Registration Using Reduced Deformable Models

Problem Statement

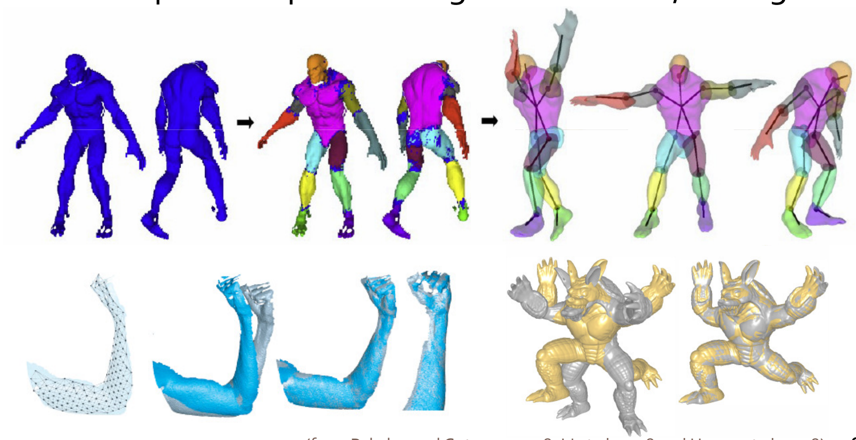
- Fit a model of the surface motion to a pair of scans
  - Articulated model (e.g. joints, smooth weights)
  - Serves as the basis for fitting on multiple frames





Related Work

- User provided segmentation: Pekelny08
- Unsupervised pairwise registration: Li08, Huang08

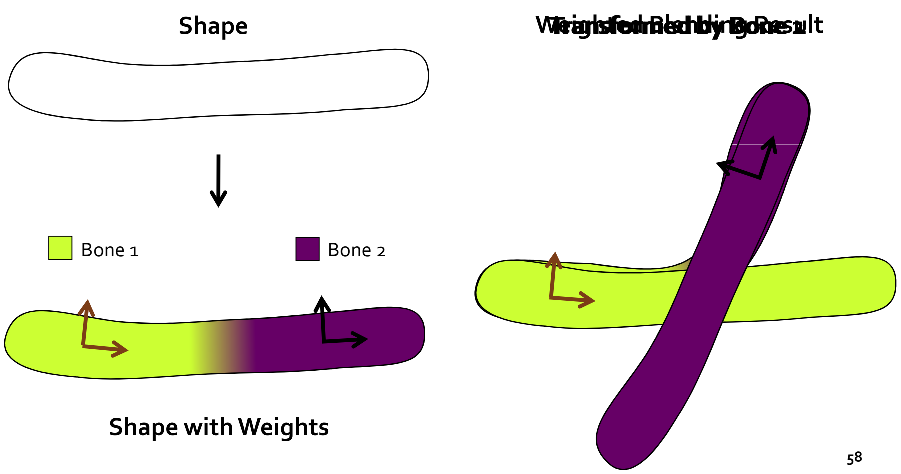


(from Pekelny and Gotsman 2008, Li et al. 2008 and Huang et al. 2008)

Problem Formulation

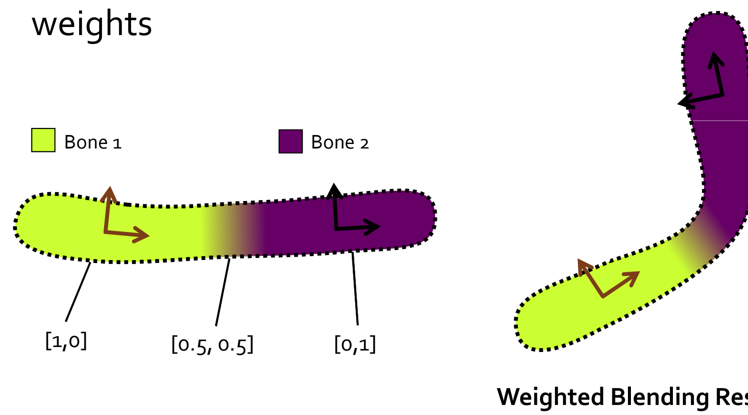
Model: Linear Blend Skinning

- Transformations (bones) and weights



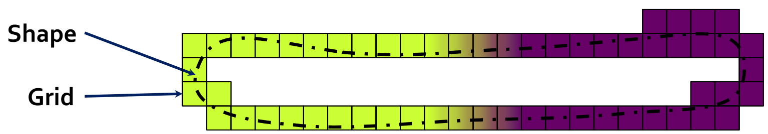
Model: Linear Blend Skinning

- Each point assigned weights in reference pose
- Transformations move each point according to its weights



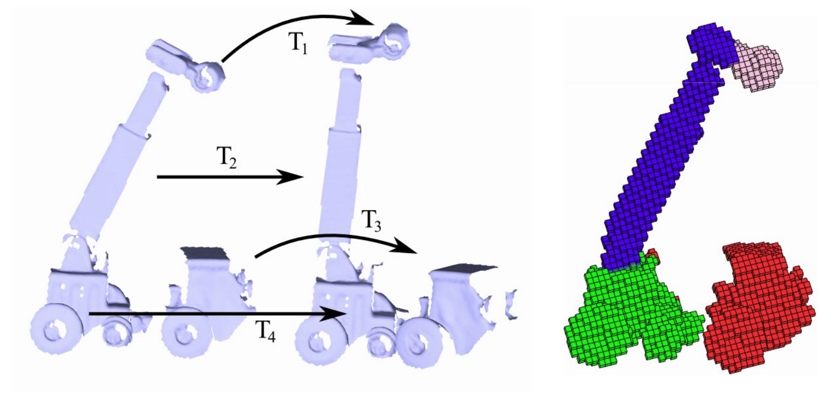
Weight Grid

- Define weights on grid enclosing surface
  - Covers small holes, reduces variables
  - Provides regular structure for optimization

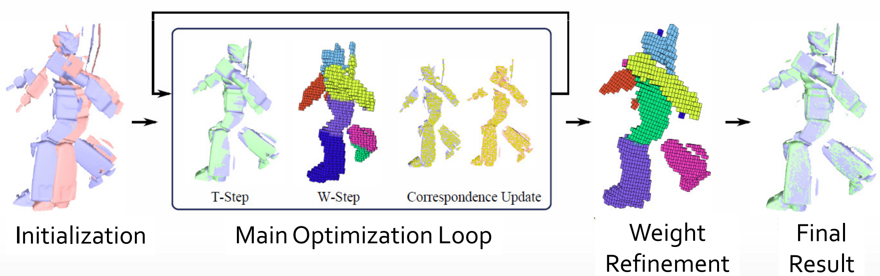


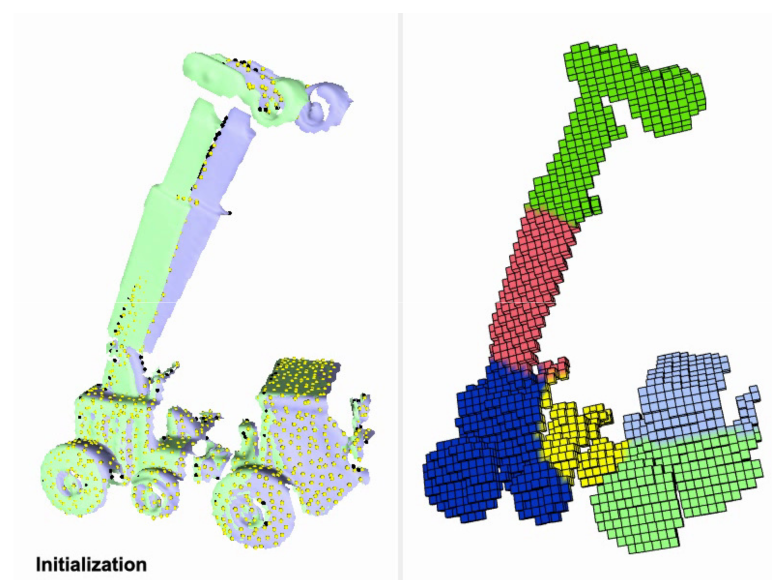
LBS for scan registration

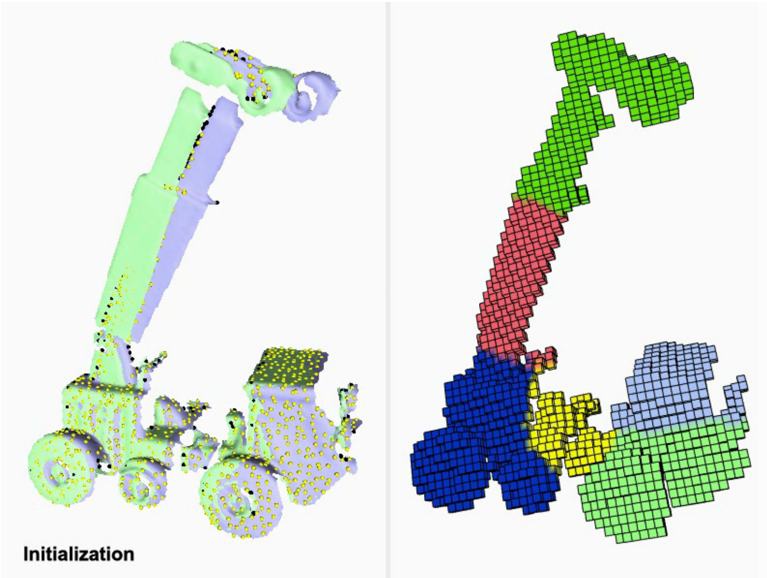
- Fit the transformations and weights to align a pair of range scans



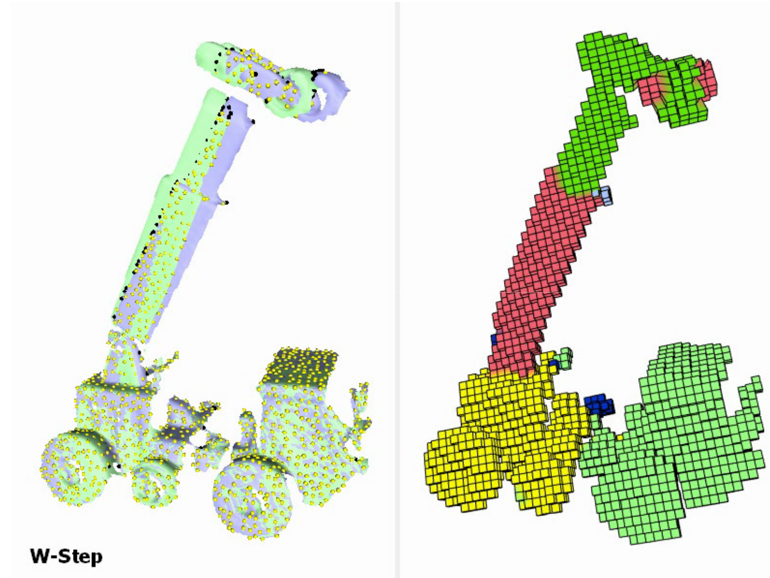
Algorithm Description

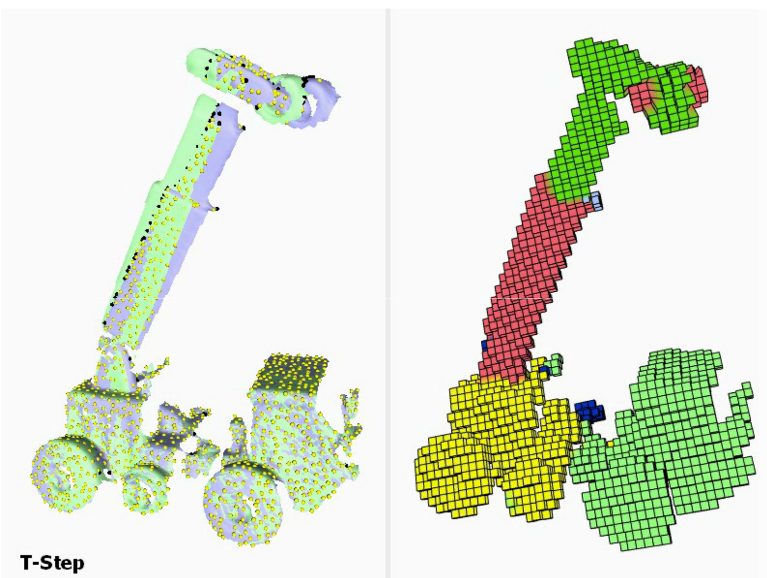




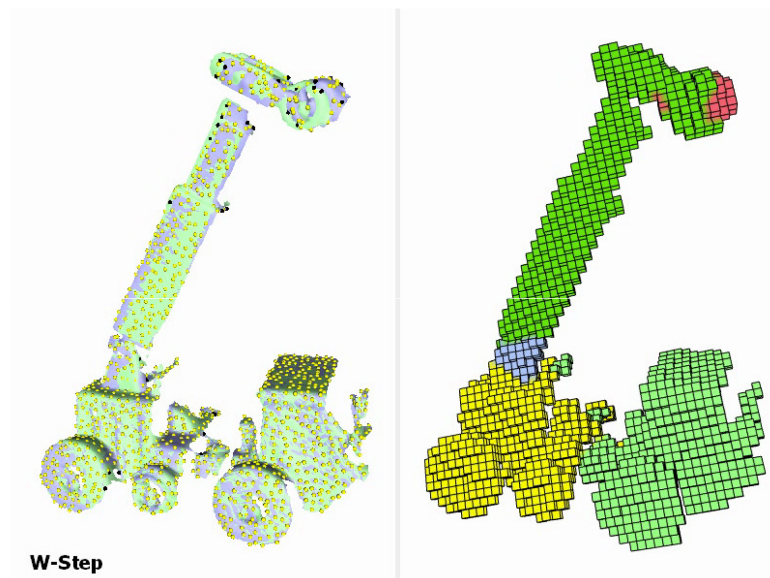


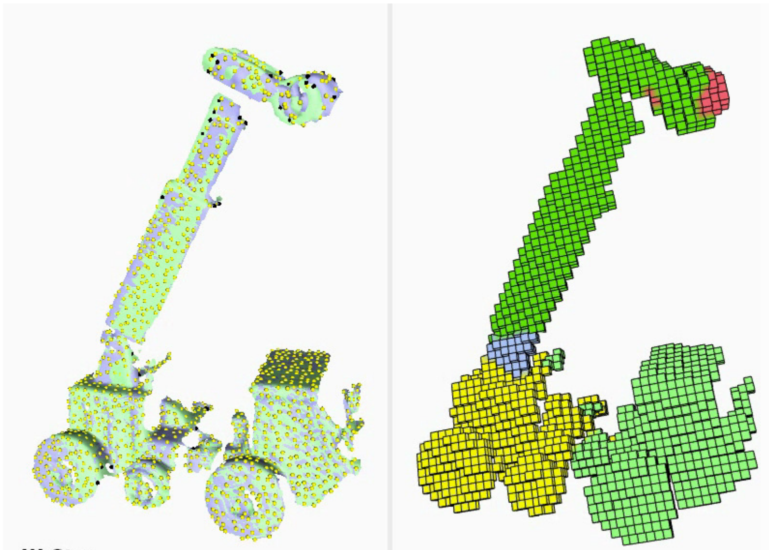
(Converged)





(Converged)

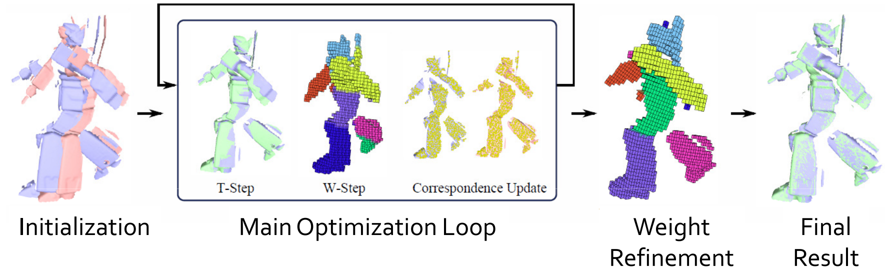




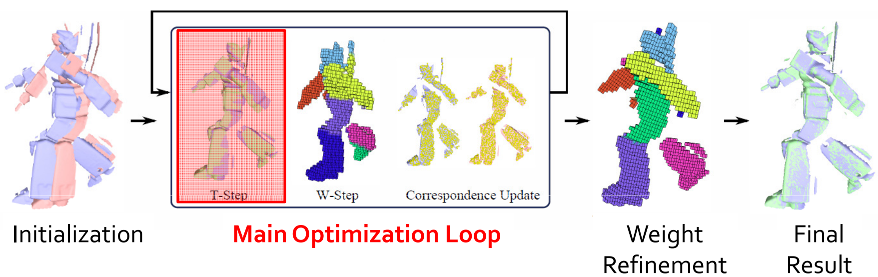
W-Step

(Finished)

Optimization overview



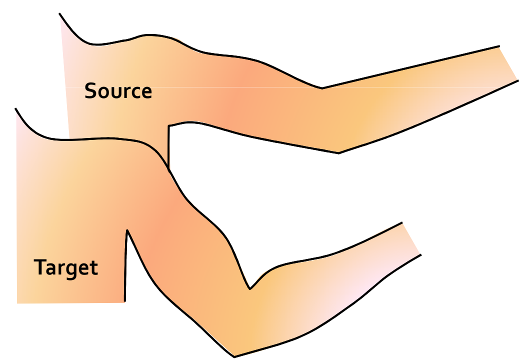
Optimization overview



- **T-Step: Optimize Alignment**
  - Distance Term
  - Joint Constraint Term

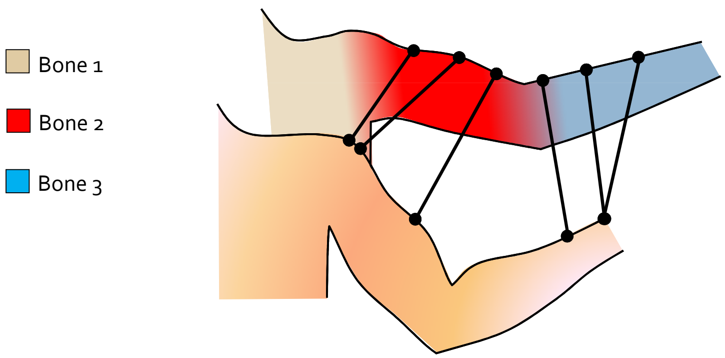
T-Step: Distance Term

- Fix weights & solve for transformations



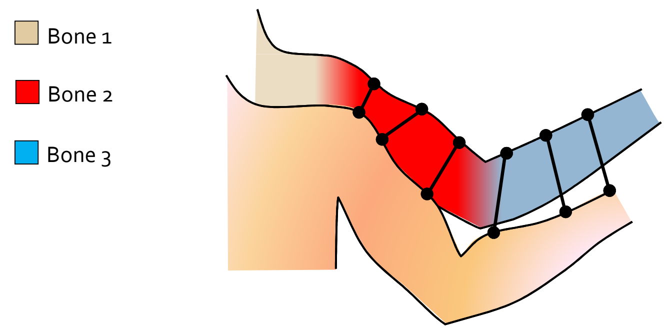
T-Step: Distance Term

- Fix weights & solve for transformations
  - Use closest point correspondences



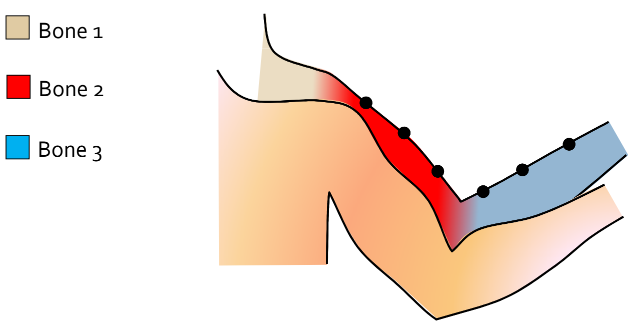
T-Step: Distance Term

- Fix weights & solve for transformations
  - Use closest point correspondences



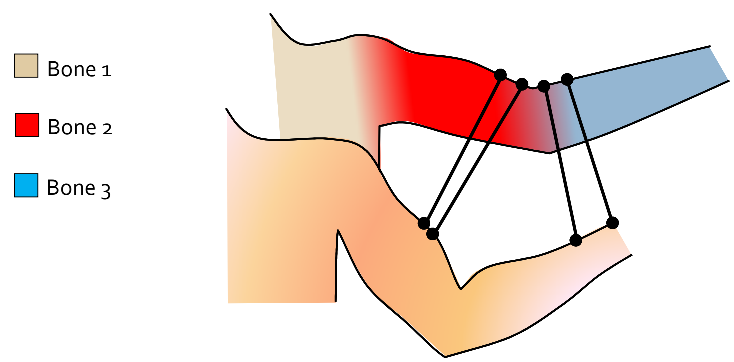
T-Step: Distance Term

- Fix weights & solve for transformations
  - Use closest point correspondences
  - Iterate further until convergence



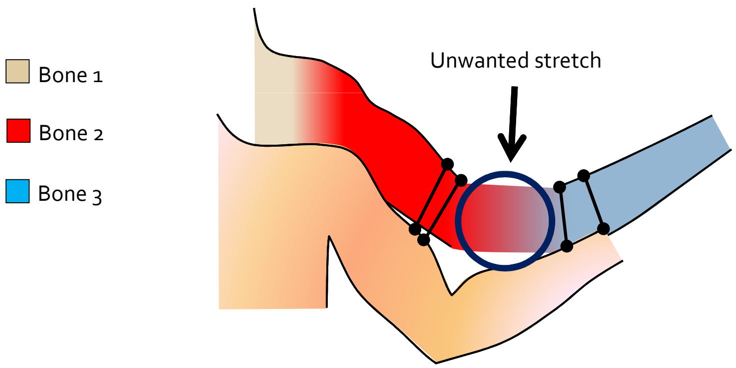
T-Step: Joint Constraint Term

- Prevent neighboring bones from separating



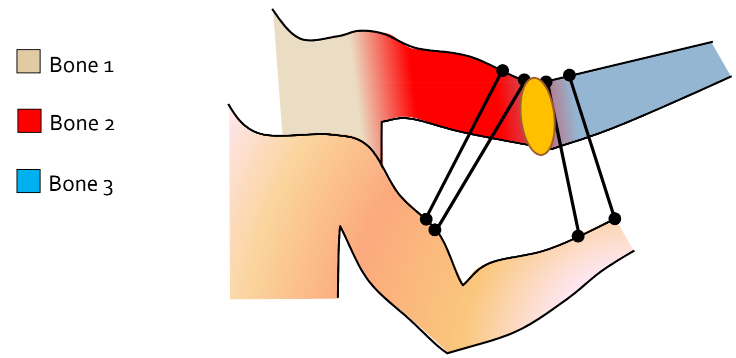
T-Step: Joint Constraint Term

- Prevent neighboring bones from separating
  - Constrain overlapping weight regions



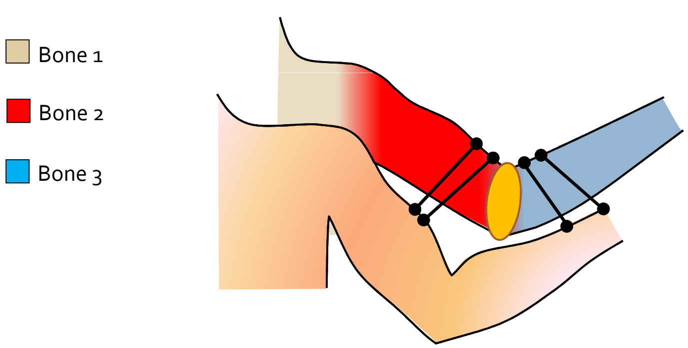
T-Step: Joint Constraint Term

- Prevent neighboring bones from separating
  - Constrain overlapping weight regions



T-Step: Joint Constraint Term

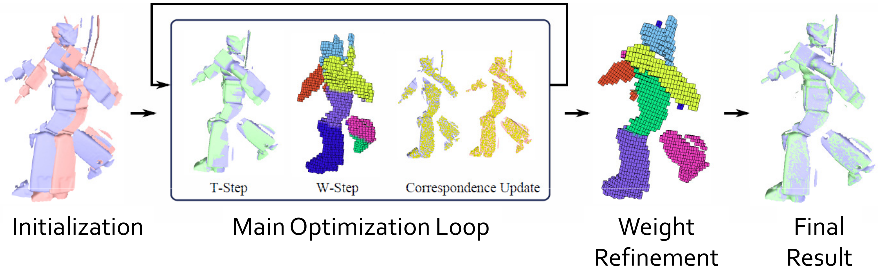
- Prevent neighboring bones from separating
  - Constrain overlapping weight regions



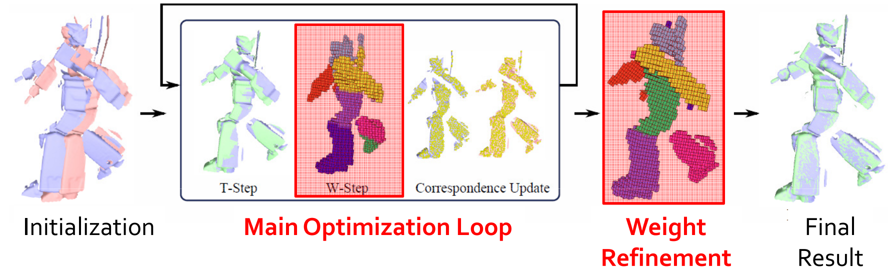
T-Step: Optimization summary

- Like rigid registration
  - Except multiple parts & joint constraints
- Non-linear least squares optimization
  - Solving for a rotation matrix
  - Gauss-Newton algorithm
  - Solve by iteratively linearizing solution
- Few variables → Fast performance
  - # variables = 6 x #bones
  - Typically 5~10 bones in our examples

Optimization overview



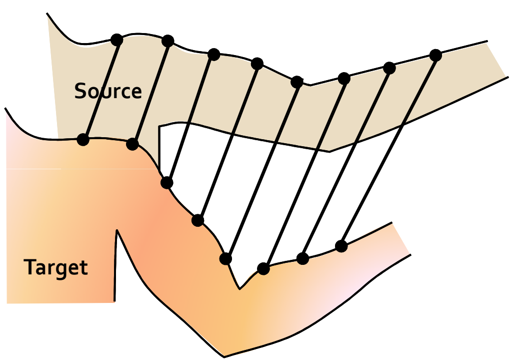
Optimization overview



- **W-Step: Optimize Weights**
  - Use Discrete Labelling
  - Continuous Weight Refinement

W-Step: Optimizing weights

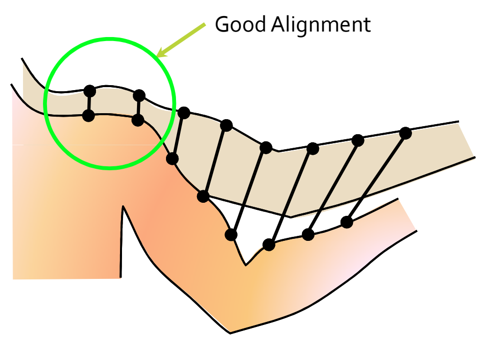
- Fix transformations, solve for continuous weights



Correspondences from last T-Step

W-Step: Optimizing weights

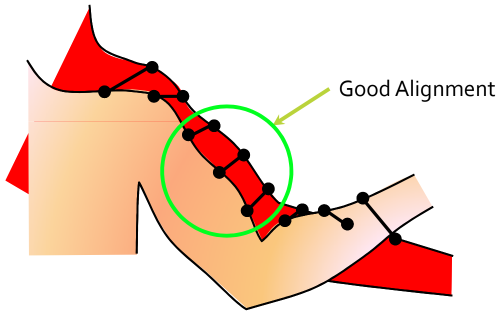
- Fix transformations, solve for continuous weights



Bone 1
(Applied to entire shape)

W-Step: Optimizing weights

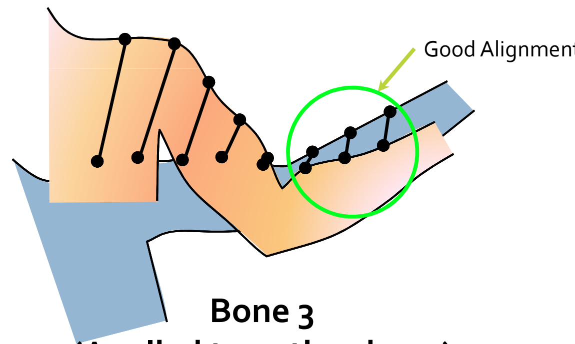
- Fix transformations, solve for continuous weights



Bone 2
(Applied to entire shape)

W-Step: Optimizing weights

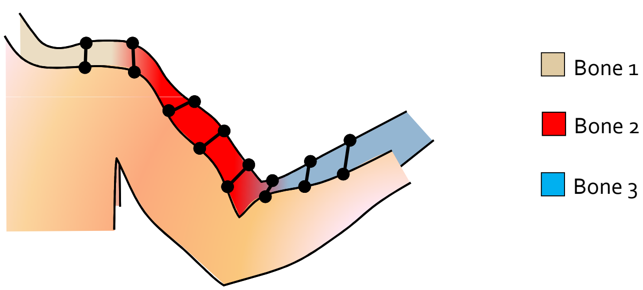
- Fix transformations, solve for continuous weights



Bone 3
(Applied to entire shape)

W-Step: Optimizing weights

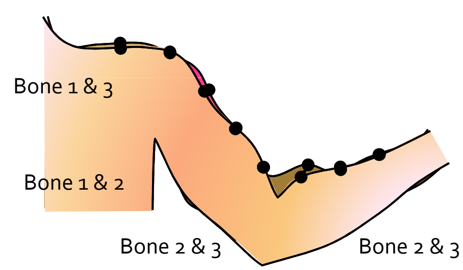
- Fix transformations, solve for continuous weights



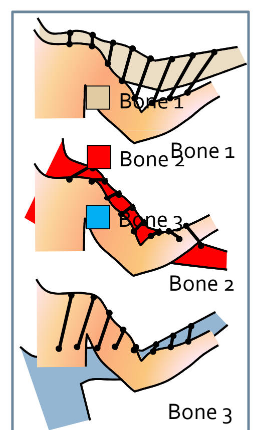
"Ideal" solved result

W-Step: Optimizing weights

- Without additional constraints, problem is underconstrained

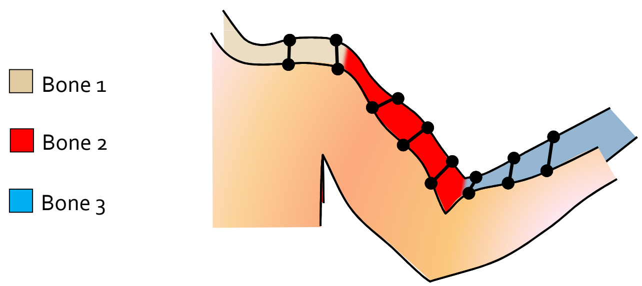


Typical solved result



Use discrete labeling

- **Our solution**: one transformation per location
  - Bones = labels
  - Becomes discrete labeling problem



Closer to "Ideal" solved result!

W-Step: Optimization Summary

- Use "graph cuts" to optimally label grid cells
  - [Boykov, Veksler & Zabih PAMI '01]
- Distance term + Smoothness term
  - Distance: measures alignment for a given label
  - Smoothness: penalizes different labels for adjacent cells
- Good Performance
  - Only ~ 1000 grid cells (graph nodes) in our examples
  - Fast performance for graph cuts

Results

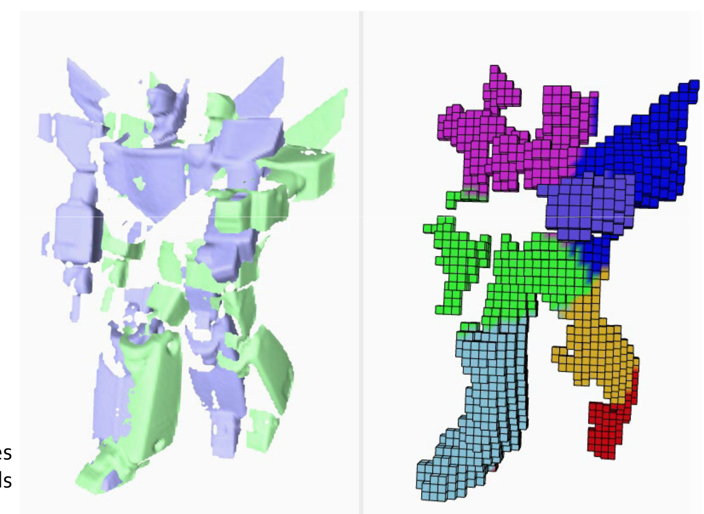
Robot, torso video

Interactive posing video

Additional results & statistics

Robot video (real-time recording)

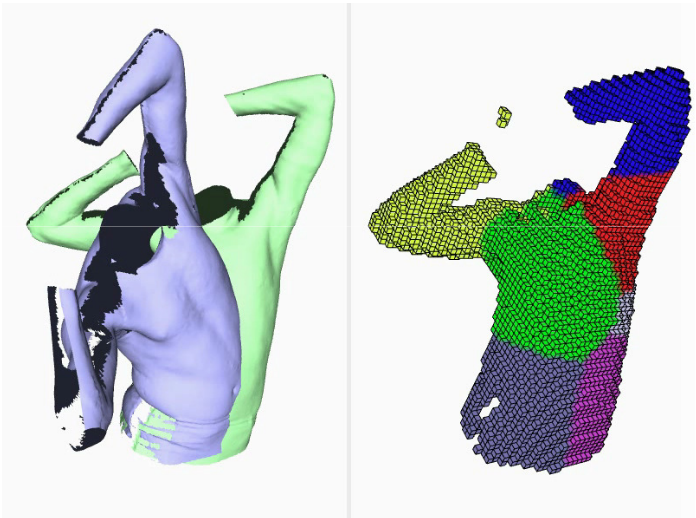
7 bones
1454 cells



Alignment Result

Solved Weights

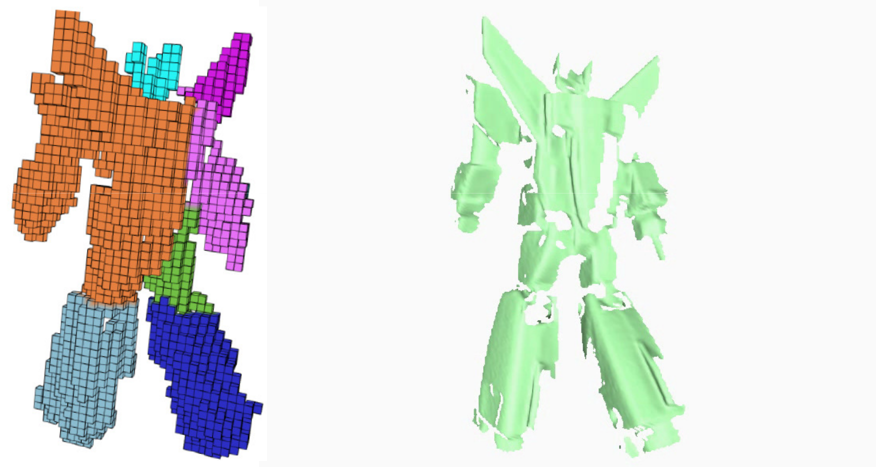
Torso video (2x speed recording)



7 bones
4890 cells

Alignment Result Solved Weights

Interactive posing (real-time recording)



Solved Weights
(7 bones, 1598 cells)

Interactive Posing Result

Average performance statistics

	Car	Robot	Walk	Hand
Bones	7	7	10	12
Corresp.	1200	1200	1000	1500
Vertices	5389	9377	4502	34342
Max Dist	20	40	20	30
Grid Res	60	65	50	40
Grid Cells	1107	1295	1014	814
Grid Points	2918	3366	2553	1884
Setup	0.185 sec	0.234 sec	0.136s ec	0.078 sec
RANSAC	8.089 sec	20.001 sec	5.517 sec	N/A
Align	9.945 sec	19.644 sec	23.092 sec	49.918 sec
Weight	6.135 sec	10.713 sec	10.497 sec	3.689 sec
Total Time	24.355 sec	50.591 sec	39.242 sec	53.684 sec

Limitations

- Discussion
  - Topology issues with grid
    - Improve in next section using graph-based approach
  - Limited to a pair of scans
    - Simultaneously register multiple frames in next section
  - Limitations with LBS
    - Optimize better model (e.g. DLB)

Conclusion

- A new algorithm to align range scans by modeling the motion with a reduced deformable model
  - Use LBS to represent the motion
  - Represent weight function using a 3D grid
  - Solve for the parameters using alternating optimization
  - No marker, template, segmentation information
  - Robust to occlusion & missing data

- **Next:** extend this method to handle multiple frames