vancouver SIGGRAPH2014
The 41st International Conference and Exhibition on Computer Graphics and Interactive Techniques

"Skinning: Real-time Shape Deformation" Course

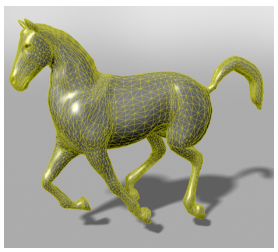
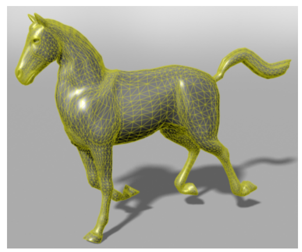
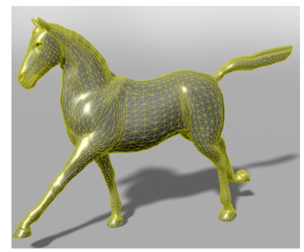
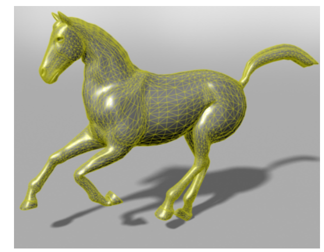
Mesh Animation Decomposition and Compression

Zhigang Deng

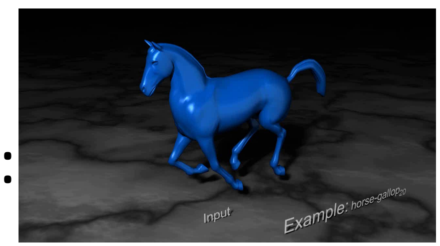
UNIVERSITY of HOUSTON

Example-based Skinning Decomposition

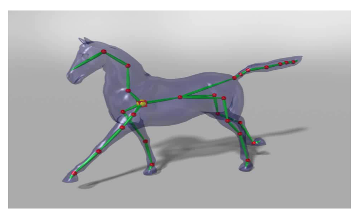
- Input

- Output
  - Without skeleton:

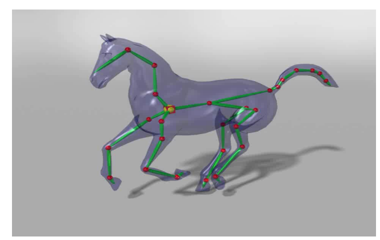


  - With skeleton:



Skeleton

+



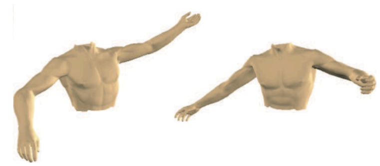
Skinning Weights

Example-based Skinning Decomposition

- Background/Motivations
  - Availability and affordability of performance capture
  - Reusing a small set of example poses for editing/

Marker-based mocap with dense markers
[Park and Hodgins 2006]

Performance Capture of Interacting Characters with Handheld Kinects
[Ye et al. 2012]

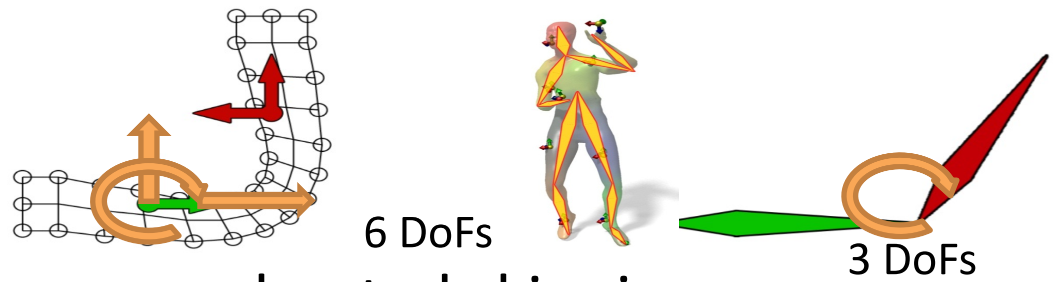


Any 3D model
Authoring tools
[Mohr and Gleicher 2003]

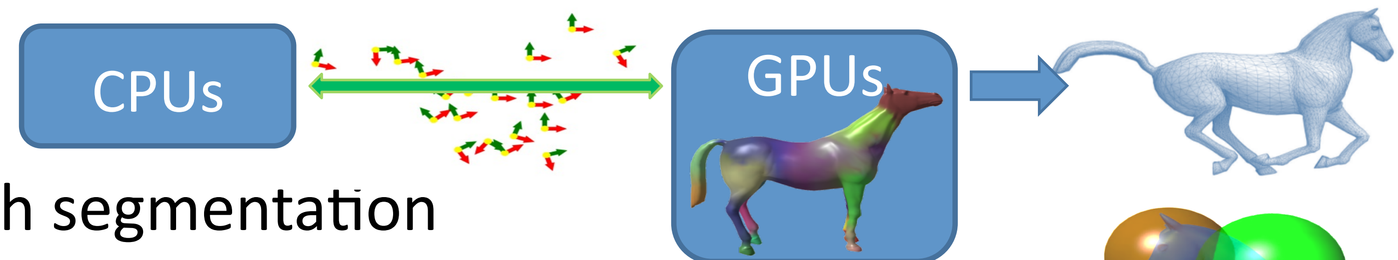
Weta Digital's Tissue System

Example-based Skinning Decomposition

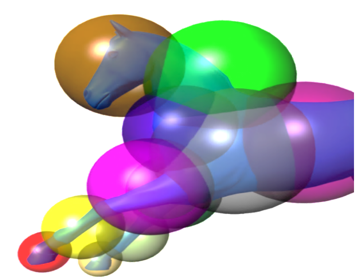
- Applications
  - Animation editing



  - Compression, hardware-accelerated skinning

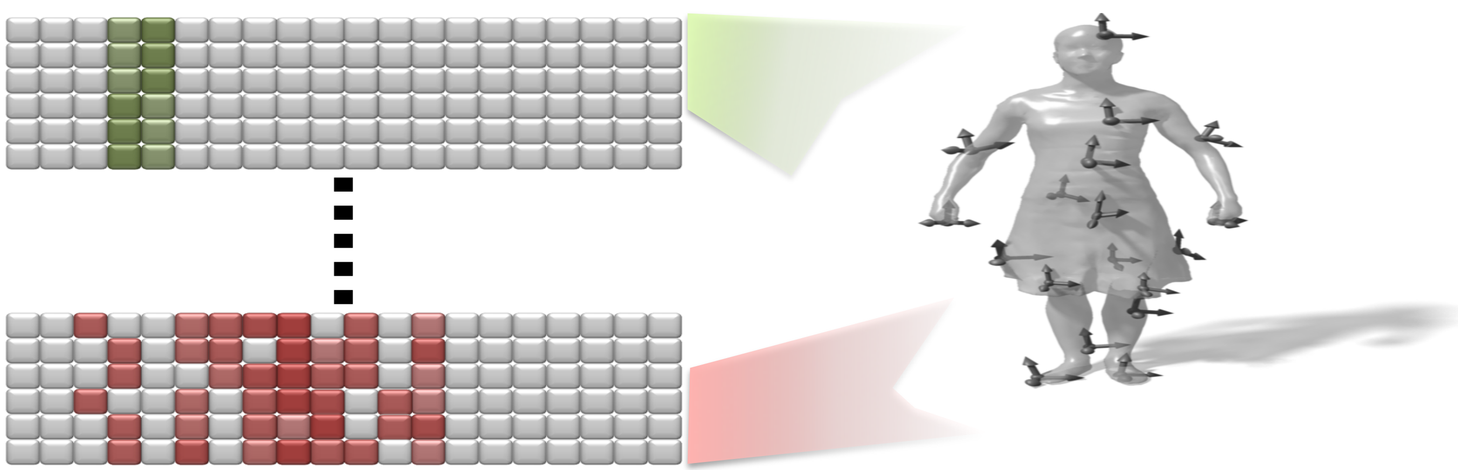


  - Mesh segmentation
  - Collision detection
  - Skin retargeting



Skinning Weight Reduction & Compression

- Motivation
  - Impose the sparseness constraint on the weights
  - Maximally retaining the visual quality



Road Map

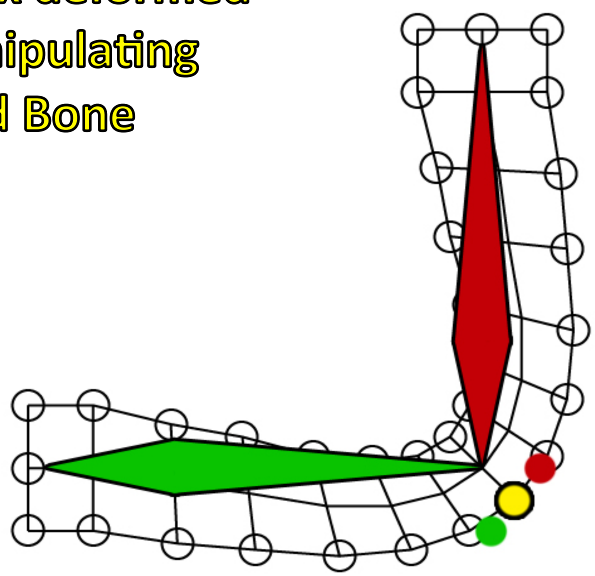
Part I:
Example-based Skinning Decomposition

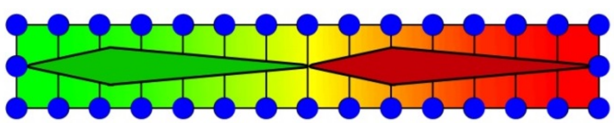
Part II:
Example-based Skeleton Extraction

Part III:
Skinning Weights Reduction & Compression

Review: Linear Blend Skinning

Yellow Vertex deformed while manipulating the Red Bone





Skinning Weights

The deformed position of vertex *i*

Rotation of bone *j*

Rest pose position on vertex *i*

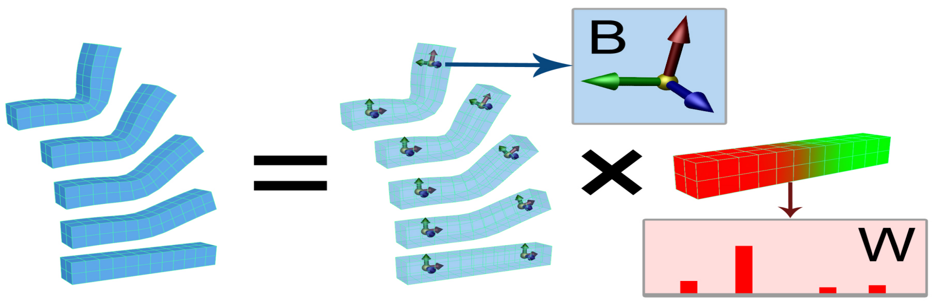
$$\mathbf{v}'_{t,i} = \sum_{j=1}^{m} w_{ij}(\mathbf{R}_{t,j}\mathbf{v}_i + \mathbf{T}_{t,j})$$

Influence of bone *j* on vertex *i*

Translation of bone *j*

Example-based Skinning Decomposition

- Definition (inverse of LBS problem)
  – Automatically extract the LBS model from example poses



- Several selected skinning decomposition methods
  – Skinning mesh animation (SMA) [James and Twigg 2005]
  – Fast and efficient skinning of animated meshes (FSD) [Kavan et al. 2010]
  – Smooth skinning decomposition with rigid bones (SSDR) [Le and Deng 2012]

Problem Formulation

- A constrained least squares optimization problem

$$\min_{\mathbf{W},\mathbf{R},\mathbf{T}} E = \min_{\mathbf{W},\mathbf{R},\mathbf{T}} \sum_{t=1}^{S} \sum_{i=1}^{n} \left\| \mathbf{v}'_{t,i} - \sum_{j=1}^{m} w_{ij} (\mathbf{R}_{t,j} \mathbf{v}_i + \mathbf{T}_{t,j}) \right\|^2$$

$\text{Subject to: } w_{ij} \geq 0, \forall i, j$ Non-negativity constraint

$$\sum_{j=1}^{m} w_{ij} = 1, \forall i$$ Affinity constraint

$$|\{w_{ij} | w_{ij} \neq 0\}| \leq K, \forall i$$ Sparseness constraint

$$\mathbf{R}_{t,j}{}^{\mathsf{T}} \mathbf{R}_{t,j} = \mathbf{I}, \det \mathbf{R}_{t,j} = 1, \forall t, j$$ Orthogonal constraint

- Constraints used in different methods
  - SSDR and SMA can handle the orthogonal constraint

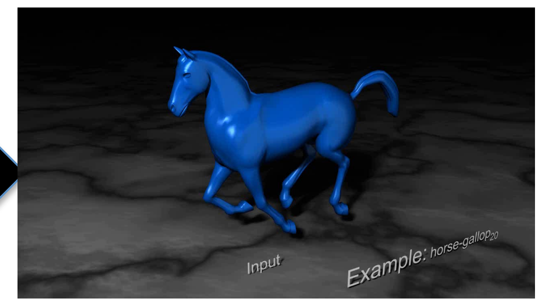
General Pipeline

Clustering-based Bone Initialization

Optimization of Skinning Parameters

Bone transformations update

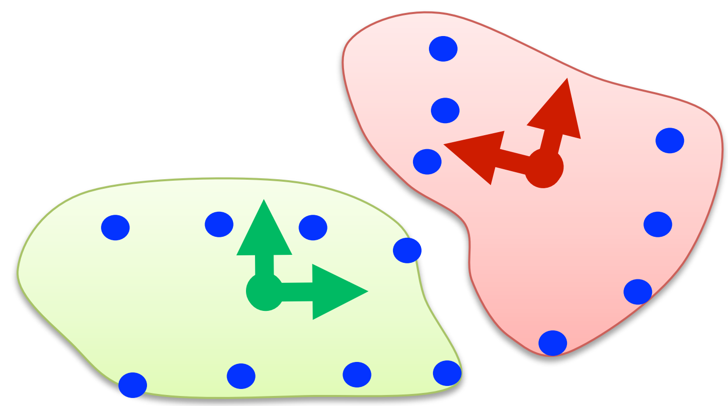
Skinning weights update



Clustering-based Bone Initialization

- Goal
  - Initialize proxy bones from example poses
- Algorithms
  - *Mean shift clustering* (**SMA**): without explicitly specifying # of bones.
  - *Multiple source region growing* (**FSD**): efficient.
  - *K-means clustering* (**SSDR**): assuming each vertex is driven by one bone only.

image courtesy of [kavan et al 2010]



K-means clustering in SSDR method

Optimization of Skinning Parameters

- Single-pass strategy in the SMA method
  - Identify influence bones per vertex (a fixed number)
    - Having the smallest squared error when predicting deformed position of vertex *i* alone.

$$e_{ij} = \sum_{t=1}^{S} ||\mathbf{v}'_{t,i} - [\mathbf{R}_{t,j}|\mathbf{T}_{t,j}]\mathbf{v}_i||_2^2, \qquad j = 1 \cdots m$$

  - Estimate bone-vertex weights
    - A constrained least squares problem

$$\sum_{j \in \mathfrak{B}_i} ([\mathbf{R}_{t,j}|\mathbf{T}_{t,j}]\mathbf{v}_i) w_{ij} = \mathbf{v}'_{t,i}, \qquad t = 1 \cdots S \qquad \sum_j w_{ij} = 1$$

- Multi-pass (Iterative) strategy in FSD and SSDR
  - Iteratively update bone transformations and skinning weights
    - only describe the algorithm in SSDR in this course

Iterative Optimization in SSDR

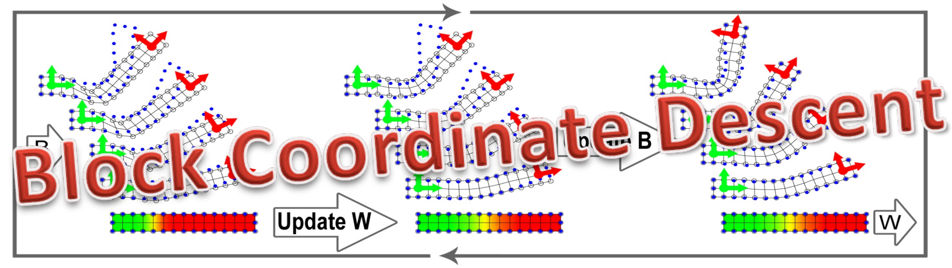
$$\min_{\mathbf{W},\mathbf{R},\mathbf{T}} E = \min_{\mathbf{W},\mathbf{R},\mathbf{T}} \sum_{t=1}^{S} \sum_{i=1}^{n} \left\| \mathbf{v}'_{t,i} - \sum_{j=1}^{m} w_{ij} (\mathbf{R}_{t,j} \mathbf{v}_i + \mathbf{T}_{t,j}) \right\|^2$$

Subject to: $w_{ij} \geq 0, \forall i, j$

$$\sum_{j=1}^{m} w_{ij} = 1, \forall i$$

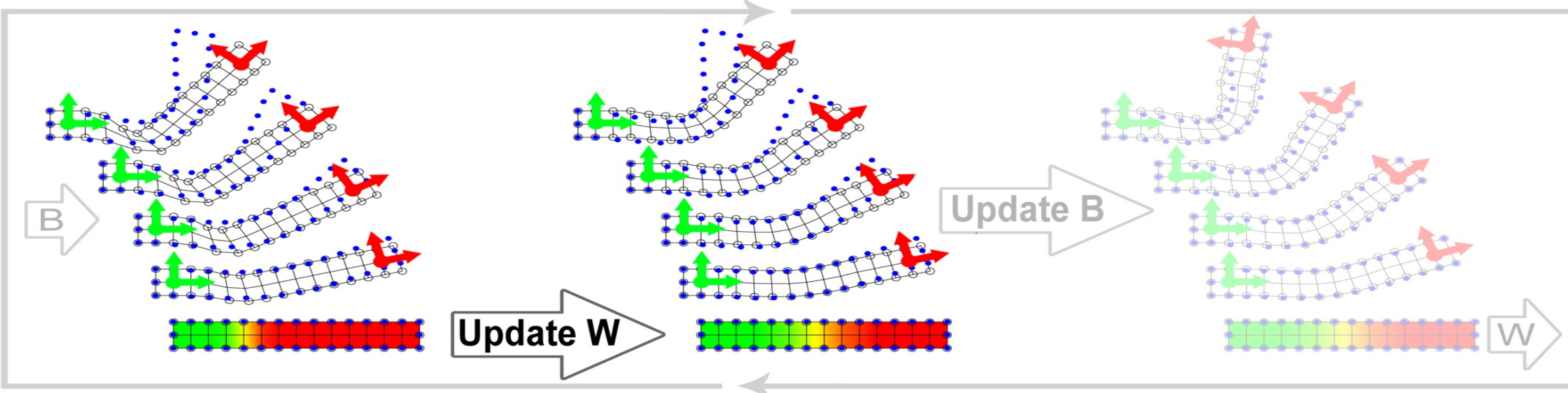
$$|\{w_{ij} | w_{ij} \neq 0\}| \leq K, \forall i$$

$$\mathbf{R}_{t,j}{}^{\mathsf{T}} \mathbf{R}_{t,j} = \mathbf{I}, \det \mathbf{R}_{t,j} = 1, \forall t, j$$



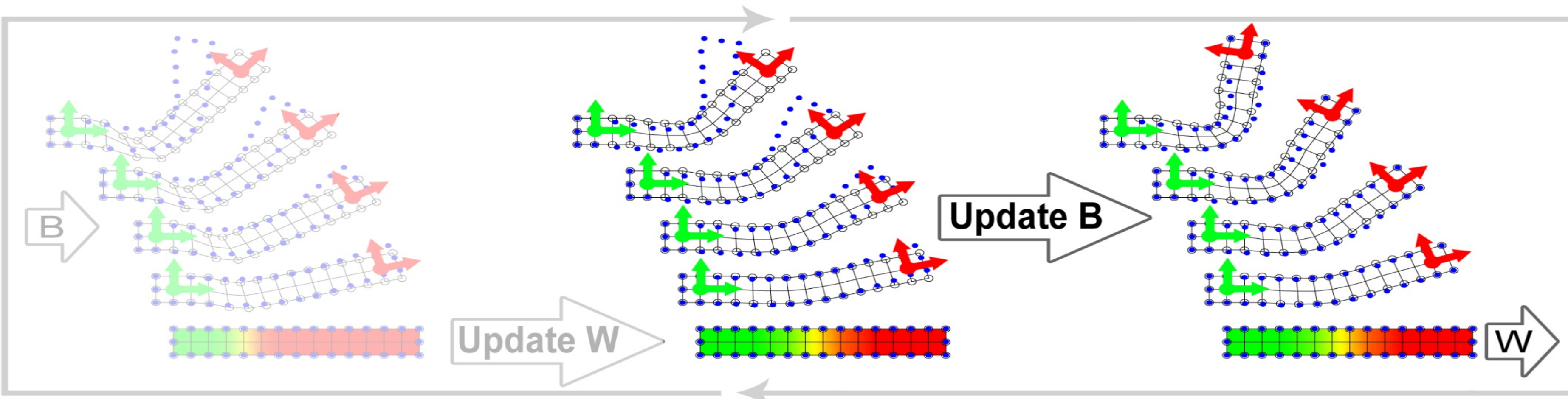
SSDR: Update Skinning Weights

$$\min_{\mathbf{W},\mathbf{R},\mathbf{T}} E = \min_{\mathbf{W},\mathbf{R},\mathbf{T}} \sum_{t=1}^{S} \sum_{i=1}^{n} \left\| \mathbf{v}'_{t,i} - \sum_{j=1}^{m} \boxed{w_{ij}} (\mathbf{R}_{t,j}\mathbf{v}_i + \mathbf{T}_{t,j}) \right\|^2$$



SSDR: Update Bone Transformations

$$\min_{\mathbf{W},\mathbf{R},\mathbf{T}} E = \min_{\mathbf{W},\mathbf{R},\mathbf{T}} \sum_{t=1}^{S} \sum_{i=1}^{n} \left\| \mathbf{v}'_{t,i} - \sum_{j=1}^{m} w_{ij} (\boxed{\mathbf{R}_{t,j}} \mathbf{v}_i + \boxed{\mathbf{T}_{t,j}}) \right\|^2$$



SSDR: Skinning Weights Update

- Per vertex solver: Constrained Linear Least Squares

$$W_i^{\mathsf{T}} = \arg\min_x ||Ax - b||^2$$

$$\text{Subject to: } \boxed{x \geq 0} \longrightarrow \text{Non-negativity Constraint}$$

$$\boxed{||x||_1 = 1} \longrightarrow \text{Affinity Constraint}$$

$$||x||_0 \leq |K|$$

- Speed up the Active Set Method [Lawson and Hanson]
  - Pre-compute LU factorization of $A^{\mathsf{T}}A$ and $A^{\mathsf{T}}b$
  - Pre-compute QR decomposition of $\begin{bmatrix}1 & 1 & \cdots & 1\end{bmatrix}^{\mathsf{T}}$

SSDR: Skinning Weights Update

- Per vertex solver: Constrained Linear Least Squares

$$W_i{}^{\top} = \arg\min_{x} ||Ax - b||^2$$

$$\text{Subject to: } x \geq 0$$

$$||x||_1 = 1$$

$$\boxed{||x||_0 \leq |K|} \rightarrow \text{Sparseness Constraint}$$

- Weight pruning of bones with small contribution

$$e_{ij} = ||w_{ij}(\mathbf{R}_{t,j}\mathbf{v}_i + \mathbf{T}_{t,j})||^2$$

Keep $|K|$ bones with largest e_{ij} and solve the LS again

Refer to the Course Note for more detailed explanation of the non-negative least squares solver with affinity constraint

SSDR: Bone Transformations Update

- Per example pose solver

$$E^t = \min_{\mathbf{R}_{t,j}|\mathbf{T}_{t,j},j=1\cdots m} \sum_{i=1}^{n} \left\| \mathbf{v}'_{t,i} - \sum_{j=1}^{m} w_{ij} (\mathbf{R}_{t,j}\mathbf{v}_i + \mathbf{T}_{t,j}) \right\|^2$$

Subject to: $\boxed{{\mathbf{R}_{t,j}}^{\mathsf{T}}\mathbf{R}_{t,j} = \mathbf{I}}, \det \mathbf{R}_{t,j} = 1, \forall t, j \rightarrow$ Nonlinear Constraint

◆ Levenberg-Marquardt optimization [Marquardt 1963]

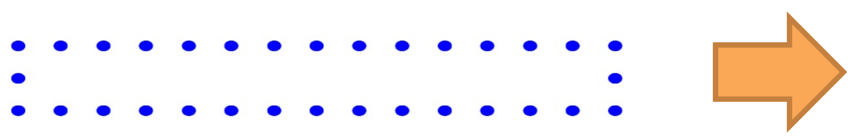
✓ Optimized solution ✖ Slow

◆ Absolute Orientation (a.k.a. Procrustes Analysis) [Kabsch 1978; Horn 1987]

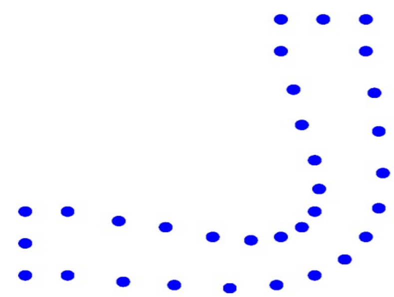
✓ Fast ✖ Approximate solution

SSDR: Bone Transformations Solver

- Solution: Solve bone transformation **one-by-one** to minimize the deformation residual of remaining bones
  - ✓ Linear solver, fast, and simple
  - ✓ Near optimized solution



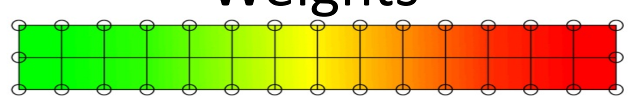
Rest pose



Example pose

Weights

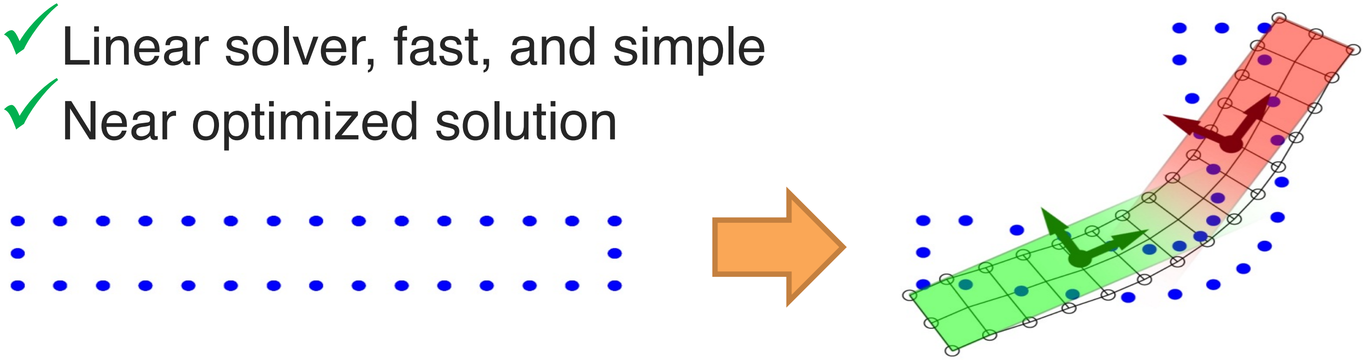
Green bone

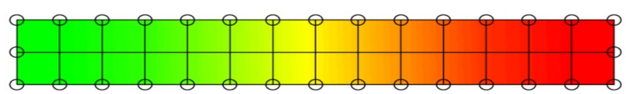


Red bone

SSDR: Bone Transformations Solver

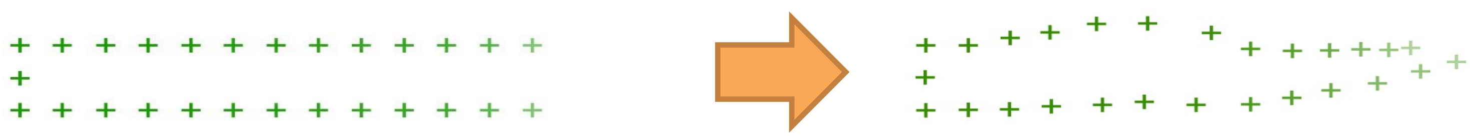
- Solution: Solve bone transformation **one-by-one** to minimize the deformation residual of remaining bones
  - ✓ Linear solver, fast, and simple
  - ✓ Near optimized solution

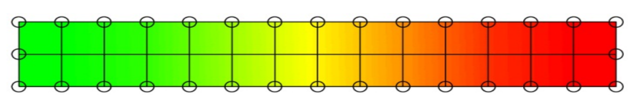




SSDR: Bone Transformations Solver

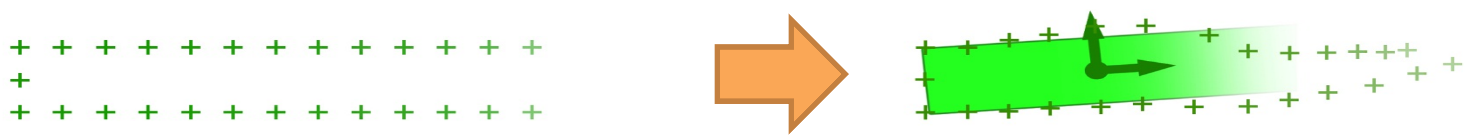
- Solution: Solve bone transformation **one-by-one** to minimize the deformation residual of remaining bones
  - ✓ Linear solver, fast, and simple
  - ✓ Near optimized solution

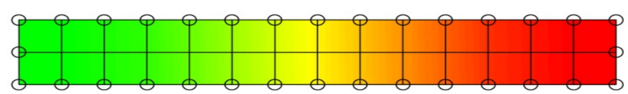




SSDR: Bone Transformations Solver

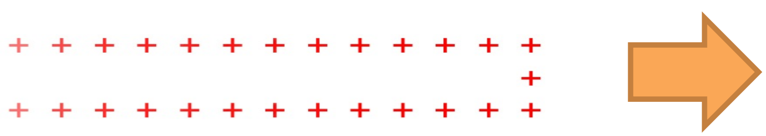
- Solution: Solve bone transformation **one-by-one** to minimize the deformation residual of remaining bones
  - ✓ Linear solver, fast, and simple
  - ✓ Near optimized solution

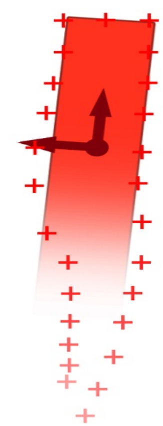


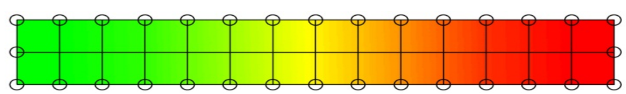


SSDR: Bone Transformations Solver

- Solution: Solve bone transformation **one-by-one** to minimize the deformation residual of remaining bones
  - ✓ Linear solver, fast, and simple
  - ✓ Near optimized solution

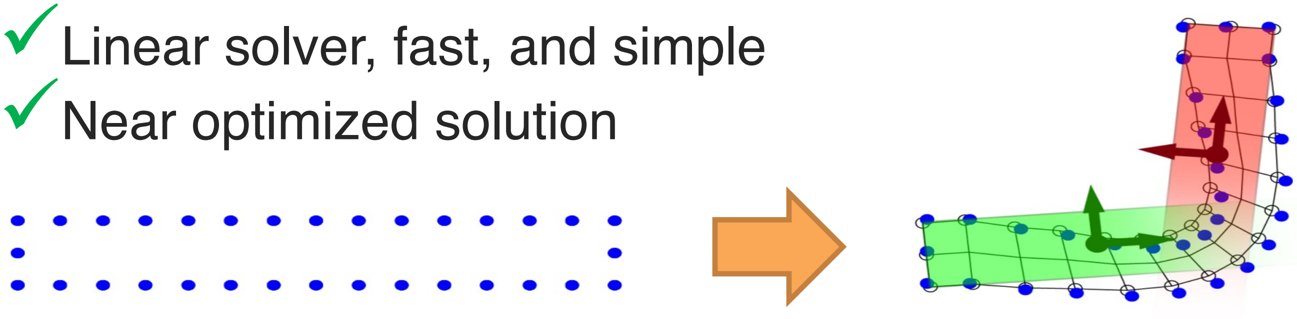


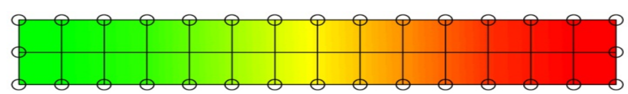




SSDR: Bone Transformations Solver

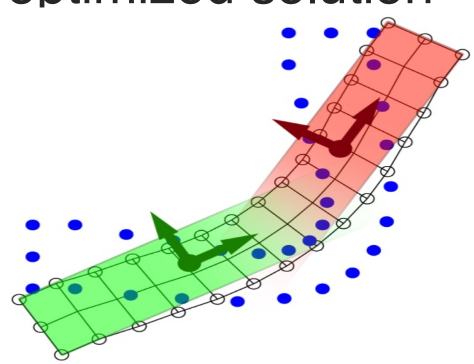
- Solution: Solve bone transformation **one-by-one** to minimize the deformation residual of remaining bones
  - ✓ Linear solver, fast, and simple
  - ✓ Near optimized solution



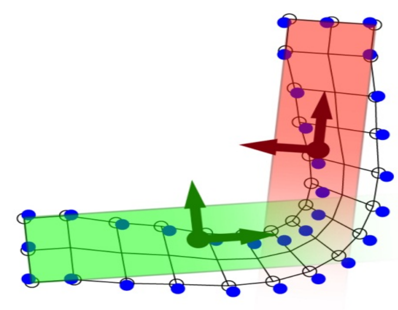


SSDR: Bone Transformations Solver

- Solution: Solve bone transformation **one-by-one** to minimize the deformation residual of remaining bones
  - ✓ Linear solver, fast, and simple
  - ✓ Near optimized solution



Before



After

SSDR: Bone Transformations Solver

- The residual q_i^t for bone $\hat{j}$

$$E_{\hat{j}}^t = \sum_{i=1}^n \Big\| \underbrace{\mathbf{v}'_{t,i} - \sum_{j=1, j\neq\hat{j}}^m w_{ij}(\mathbf{R}_{t,j}\mathbf{v}_i + \mathbf{T}_{t,j})}_{q_i^t} - \boxed{w_{i\hat{j}}(\mathbf{R}_{t,\hat{j}}\mathbf{v}_i + \mathbf{T}_{t,\hat{j}})} \Big\|^2$$

Bone $\hat{j}$ out

- Now find the rigid transformation

$$\mathbf{v}_i \xrightarrow{(\mathbf{R}_{t,\hat{j}},\ \mathbf{T}_{t,\hat{j}})} q_i^t$$

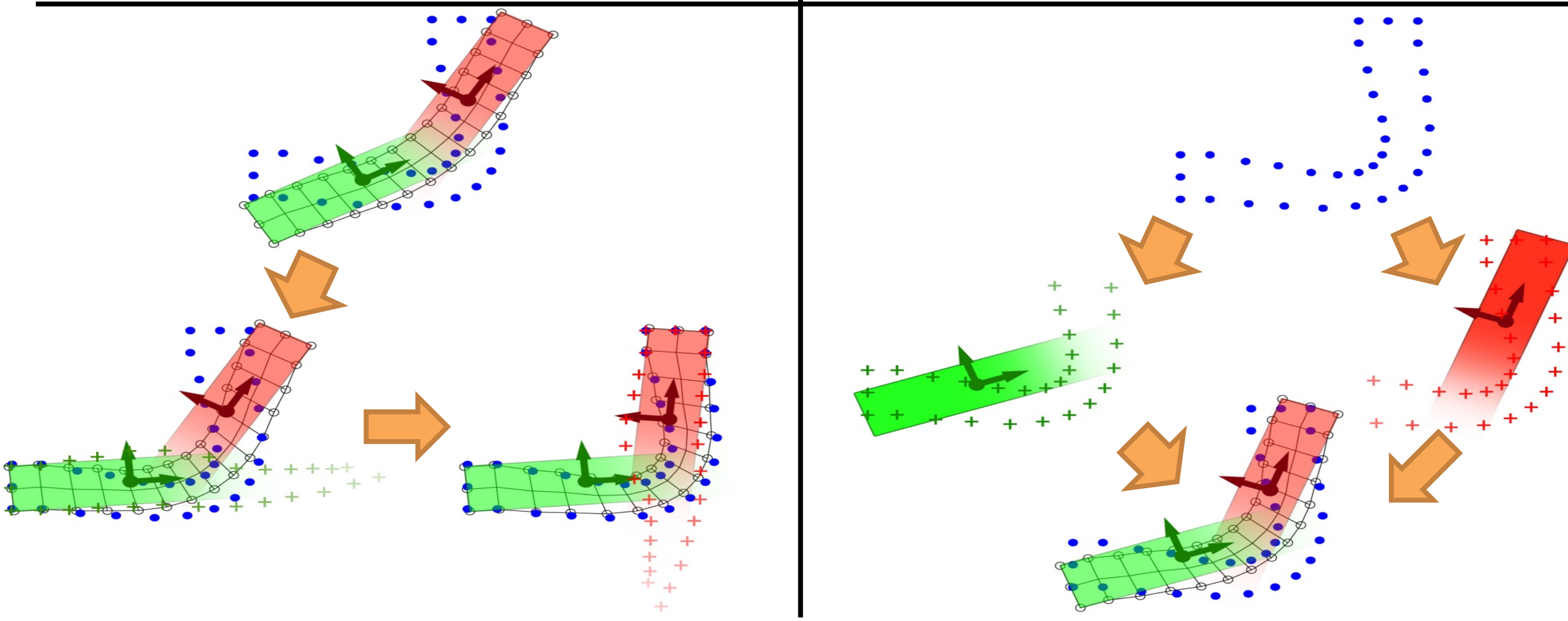
SSDR: Bone Transformations Solver

SSDR method	Weighted Absolute Orientation
$E^t = \sum_{i=1}^{n} \left\| \mathbf{v}'_{t,i} - \underbrace{\sum_{j=1}^{m} w_{ij}(\mathbf{R}_{t,j}\mathbf{v}_i + \mathbf{T}_{t,j})}_{\text{Bone blending}} \right\|^2$	$E^t = \sum_{i=1}^{n} \sum_{j=1}^{m} w_{ij} \left\| \mathbf{v}'_{t,i} - \underbrace{(\mathbf{R}_{t,j}\mathbf{v}_i + \mathbf{T}_{t,j})}_{\text{No blending}} \right\|^2$

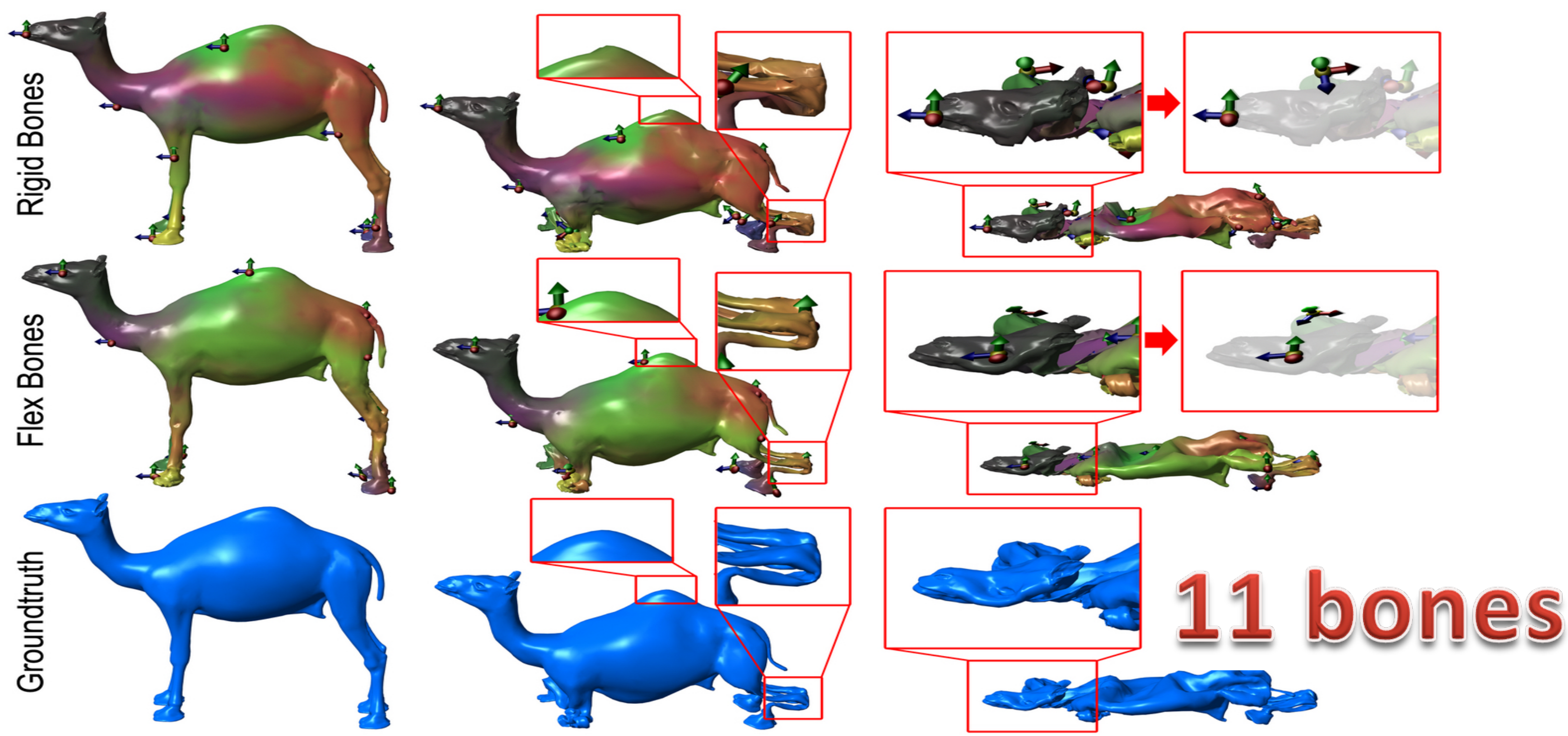
Different objective functions!

A Toy Example

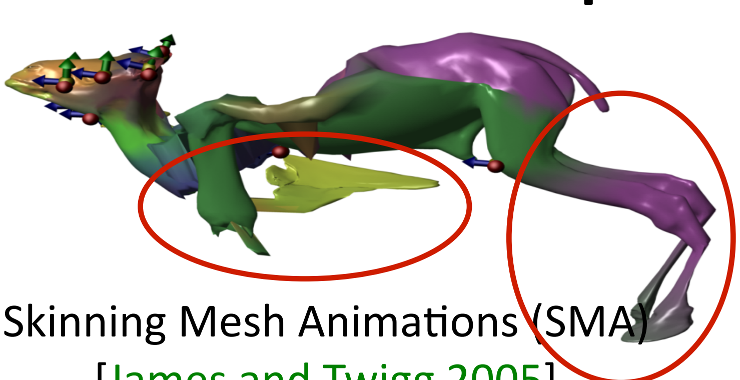
SSDR method | Weighted Absolute Orientation



SSDR: Rigid Bones versus Flex Bones

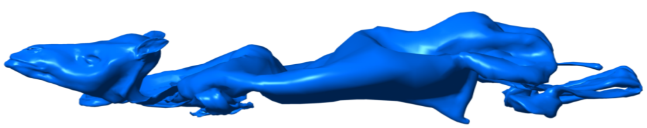


Comparisons of Different Skinning Decomposition Methods

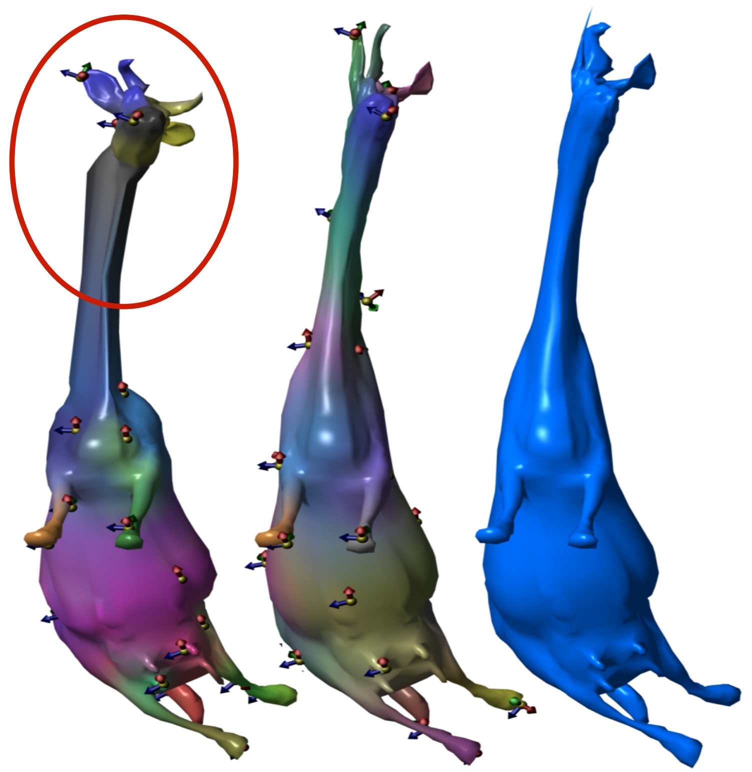


Skinning Mesh Animations (SMA) [James and Twigg 2005]

SSDR method [Le and Deng 2012]

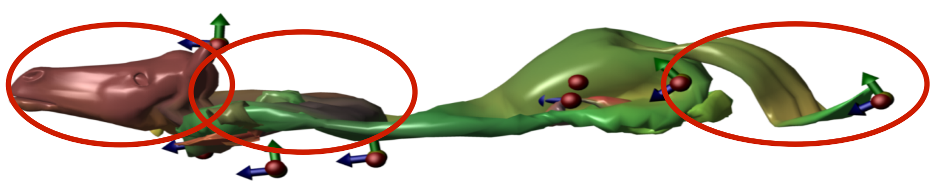


Ground truth

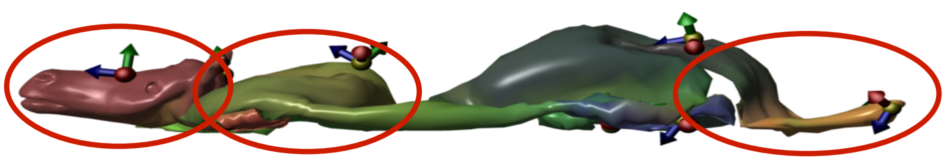


SMA SSDR Ground truth

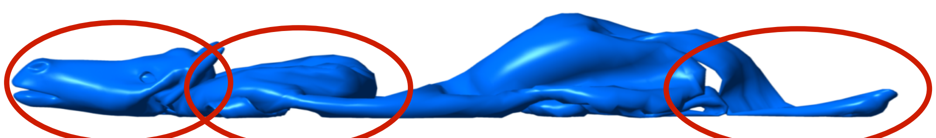
Comparisons of Different Skinning Decomposition Methods



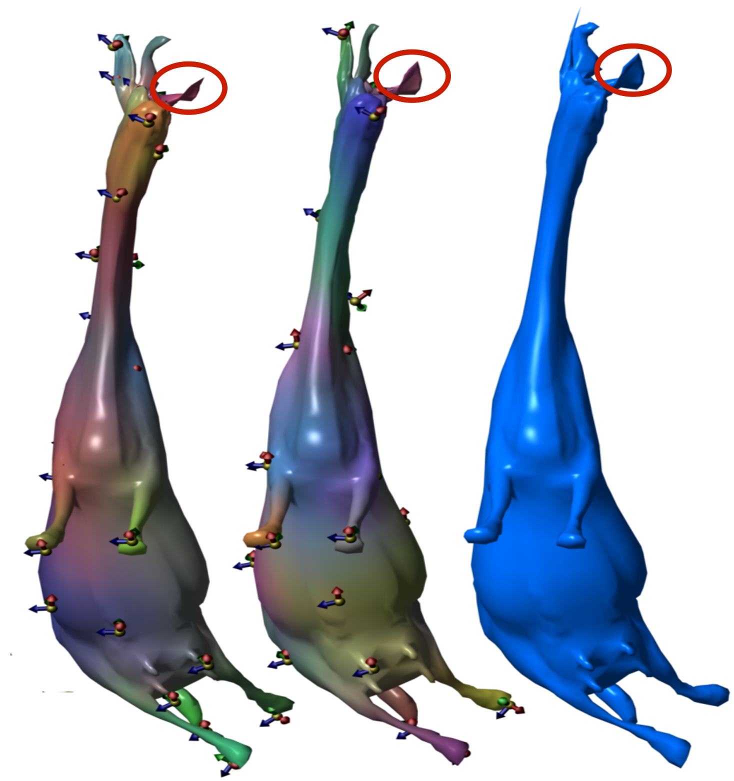
Learning Skeletons for Shape and Pose (LSSP)
[Hasler et al. 2010]



SSDR method [Le and Deng 2012]



Ground truth



LSSP SSDR Ground truth

Comparisons of Different Skinning Decomposition Methods

Dataset$_{\text{[No. of bones]}}$	Approximation error E_{RMS}			Execution time (minutes)		
	SMA	LSSP	SSDR	SMA	LSSP	SSDR
camel-collapse$_{11}$	125.3 (4)	-	5.4(1.7)	13.8	-	7.4
cat-poses$_{25}$	8.5 (3.1)	6.2(3.3)	3.4(1.4)	0.7	371.7	1.5
chickenCrossing$_{28}$	12.5 (4.2)	6.2(5.1)	8.1(5.4)	14.1	1165.4	24
horse-gallop$_{33}$	9.5 (1.5)	12.5(4.6)	2.2(1.1)	3.8	911	9.8
lion-poses$_{21}$	62.8 (5.7)	7.7(3.9)	4.4(2.2)	0.6	360.2	0.8
pcow$_{24}$	24.8 (13.2)	7.2(6.7)	5.7(4.8)	3.8	564.5	8.9
pdance$_{24}$	3.8 (1.6)	3.4(2.3)	1.3(0.8)	22	2446.8	28.3

Result in parentheses: rank-5 EigenSkin correction [Kry et al. 2002]

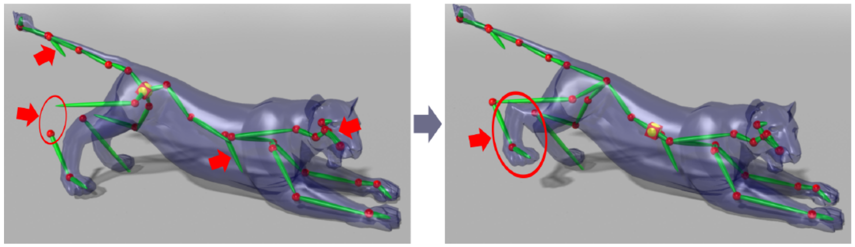
Part I:
Example-based Skinning Decomposition

Part II:
Example-based Skeleton Extraction

Part III:
Skinning Weights Reduction & Compression

Example-based Skeleton Extraction

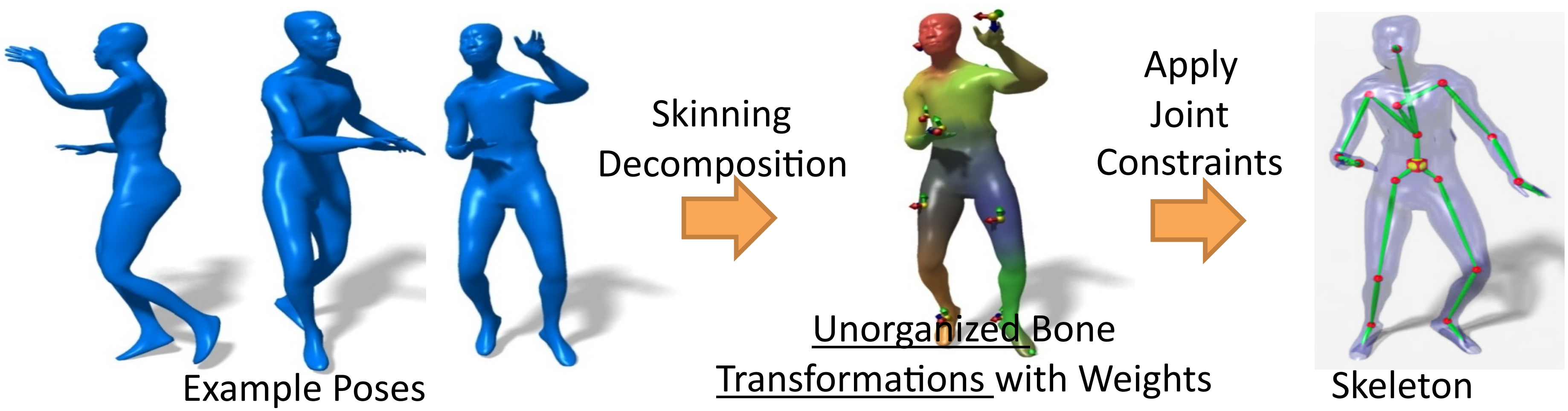
- Concept
  - Directly extract skeleton from example poses
  - Minimize the example pose reconstruction error, while handling rotational joint constraint
- Categories of method
  - Single-pass methods [Schaefer and Yuksel 2007; de Aguiar et al. 2008a; Hasler et al. 2010]
    - ✔ Fast and efficient
    - ✖ Redundant bones in skeleton
    - ✖ Accumulated errors in skeleton
  - Multi-pass methods [Le and Deng 2014]
    - ✖ Slow and more computational time
    - ✔ Accurate and robust skeleton



Results from an implementation of [Hasler et al. 2010]

Single-pass Methods

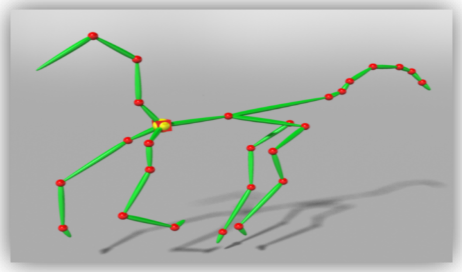
- Initialization
  - Proxy bones and skinning weights
- Skeleton construction
  - Linking bones to a tree-structure
- Joint positions calculation
  - Computing joint position



Single-pass: Initialization

- Generating bone transformations and skinning weights
  - Similar to the example-based skinning decomposition
- Initialization algorithms
  - Example-based skinning decomposition (SSDR, FSD, SMA, LSSP, etc)
  - Spectral clustering [de Aguiar et al. 2008a]
    - Seed vertices drive the segmentation of nearly rigid parts.
  - Edge collapsing [Schaefer and Yuksel 2007]
    - Bottom-up hierarchical clustering strategy

Single-pass: Skeleton Reconstruction



- A weighted graph **G**
  - Nodes correspond to bones
  - Cost of edge

$$g(j,k) = \frac{\sum_{t=1}^{S} \left\| \left([\mathbf{R}_{t,j} | \mathbf{T}_{t,j}] - [\mathbf{R}_{t,k} | \mathbf{T}_{t,k}] \right) \begin{bmatrix} \mathbf{o}_{jk} \\ 1 \end{bmatrix} \right\|_2^2}{\sum_{i=1}^{n} w_{ij} w_{ik}}$$

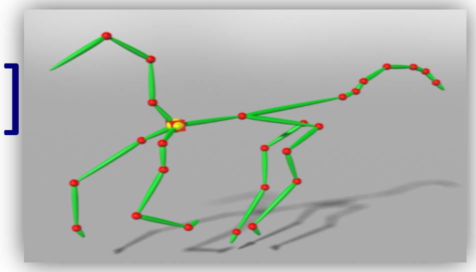
Numerator: Joint constraint value

Denominator: Weight blending of 2 bones

- Minimum spanning tree algorithm
  - [Kirk et al. 2005; Schaefer and Yuksel 2007; de Aguiar et al. 2008; Hasler et al. 2010]

Single-pass: Joint Position Solver

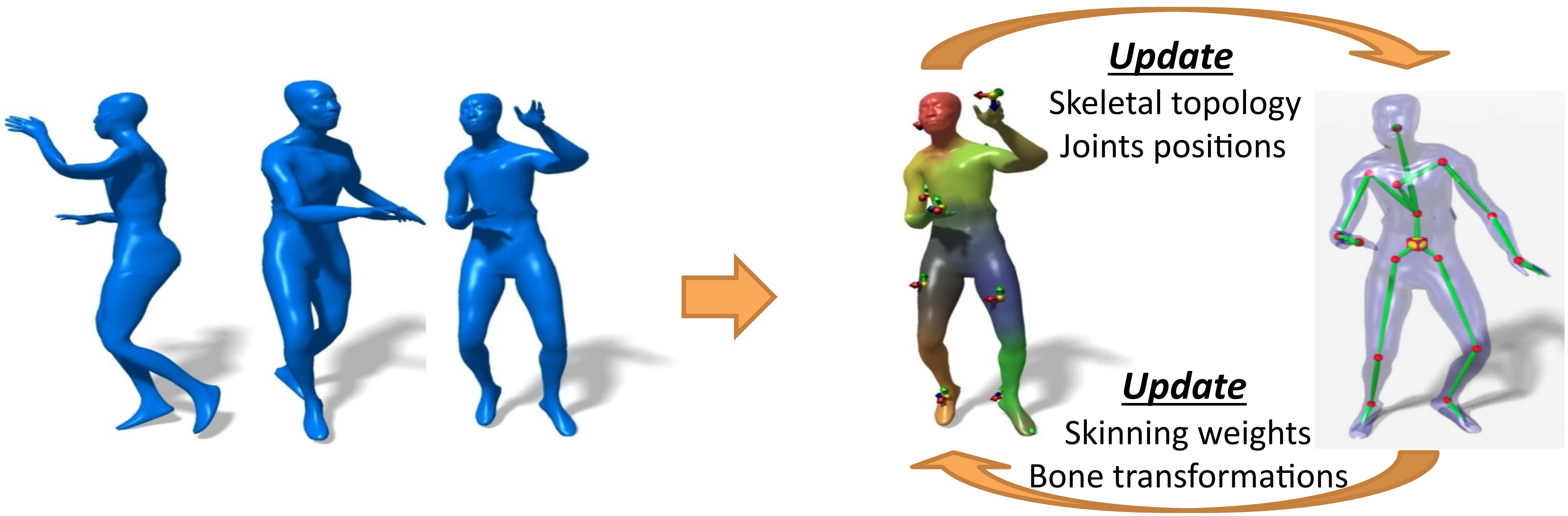
- Joint position solver [Anguelov et al. 2004]



$$\min_{\mathbf{o}_{jk}} \sum_{t=1}^{S} \underbrace{\left\| \left([\mathbf{R}_{t,j}|\mathbf{T}_{t,j}] - [\mathbf{R}_{t,k}|\mathbf{T}_{t,k}] \right) \begin{bmatrix} \mathbf{o}_{jk} \\ 1 \end{bmatrix} \right\|_2^2}_{\text{Joint constraint term}} + \mu \underbrace{\|\mathbf{g}_j + \mathbf{g}_k - 2\mathbf{o}_{jk}\|_2^2}_{\text{Distance between the joint and the centroid of two bones}}$$

Multi-pass Method

- Iterative rigging strategy [Le and Deng 2014]



Multi-pass Method

Multi-pass: Objective Function

$$\min_{\mathbb{S},\mathbf{o}_{jk},w_{ij},[\mathbf{R}_{t,j}|\mathbf{T}_{t,j}]} E = E_D + \omega E_S + \lambda E_J$$

Where:

$$E_D = \frac{1}{nS}\sum_{i=1}^{n}\sum_{t=1}^{S}\left\|\sum_{j=1}^{m} w_{ij}[\mathbf{R}_{t,j}|\mathbf{T}_{t,j}]\begin{bmatrix}\mathbf{v}_i\\1\end{bmatrix} - \mathbf{v}'_{t,i}\right\|_2^2$$

Data Term

$$E_S = \sum_{j=1}^{m} \mathbf{w}_j{}^{\top}\mathbf{L}\mathbf{w}_j$$

Weights Smoothness

$$E_J = \frac{1}{S}\sum_{(j,k)\in\mathbb{S}}\sum_{t=1}^{S}\left\|\left([\mathbf{R}_{t,j}|\mathbf{T}_{t,j}] - [\mathbf{R}_{t,k}|\mathbf{T}_{t,k}]\right)\begin{bmatrix}\mathbf{o}_{jk}\\1\end{bmatrix}\right\|_2^2$$

Joint Constraints

Subject to:

$$w_{ij} \geq 0,\ \sum_{j=1}^{m} w_{ij} = 1,\ \|w_i\|_0 \leq K(=4),\ \forall i,j$$

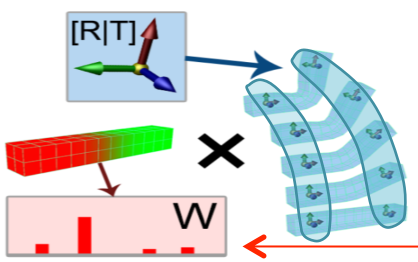
$$\mathbf{R}_{t,j}{}^{\top}\mathbf{R}_{t,j} = \mathbf{I},\ \det\mathbf{R}_{t,j} = 1,\ \forall j,t$$

$$\min E = \boxed{E_D} + \omega E_S + \lambda E_J$$

Multi-pass: Data Term

- Data Term
  - Minimizing reconstruction error w.r.t. Weights & Bone transformations [Le and Deng 2012]
  - No skeletal structure

$$E_D = \frac{1}{nS} \sum_{i=1}^{n} \sum_{t=1}^{S} \left\| \sum_{j=1}^{m} w_{ij} [\mathbf{R}_{t,j} | \mathbf{T}_{t,j}] \begin{bmatrix} \mathbf{v}_i \\ 1 \end{bmatrix} - \mathbf{v}'_{t,i} \right\|_2^2$$

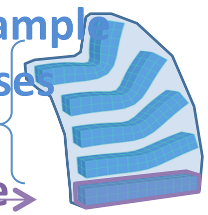


Bone transformations

Skinning weights

Input example poses

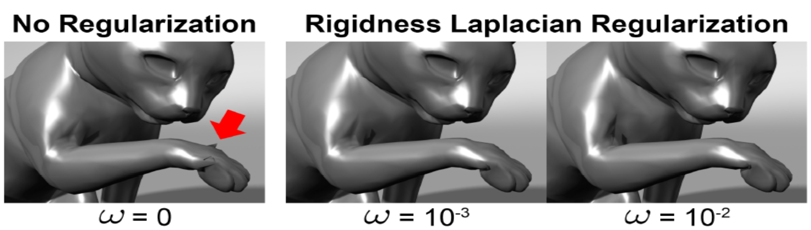
Rest pose



$$\min E = E_D + \omega \boxed{E_S} + \lambda E_J$$

Multi-pass: Weight Regularization Term

- Skinning Weight Regularization Term
  - No regularization: Fracture due to weights sparseness constraint or noisy input data
  - Our rigidness Laplacian regularization: Smooth, deformation sensitive



$$E_S = \sum_{j=1}^{m} \mathbf{w}_j{}^{\mathsf{T}} \mathbf{L} \mathbf{w}_j$$

$$L_{ik} = \begin{cases} 1 & \text{if } k = i \\ -\frac{d_{ik}}{\sum_{h \in \mathcal{N}(i)} d_{ih}} & \text{if } k \in \mathcal{N}(i) \\ 0 & \text{otherwise.} \end{cases}$$

Where: $\mathcal{N}(i)$ denotes all the 1-ring neighbors of vertex i

Evaluated on all example poses ($f = 1..F$) →

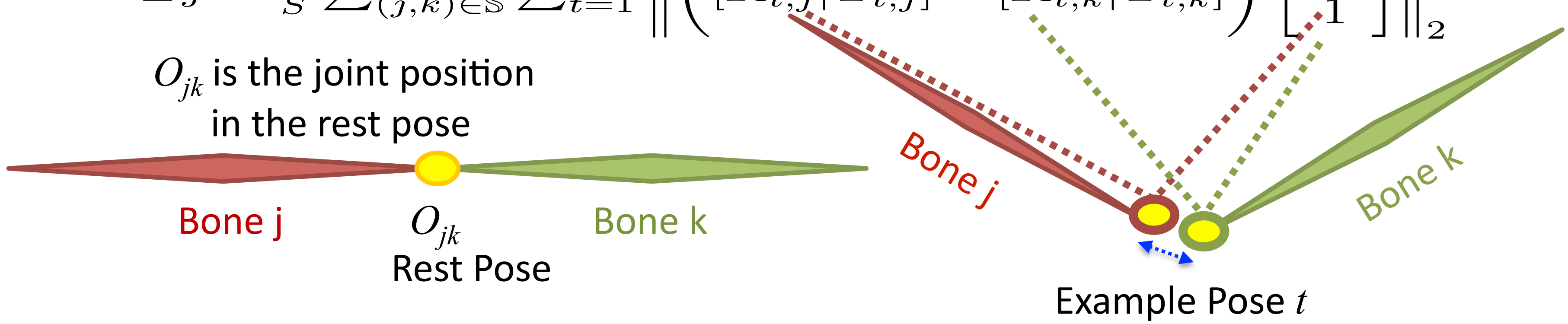
$$d_{ik} = \frac{1}{\sqrt{\frac{1}{S} \sum_{t=1}^{S} \left(\left\| \mathbf{v}'_{t,i} - \mathbf{v}'_{t,k} \right\|_2 - \| \mathbf{w}_i - \mathbf{w}_k \|_2 \right)^2}}$$

$$\min E = E_D + \omega E_S + \lambda \boxed{E_J}$$

Multi-pass: Joint Constraint Term

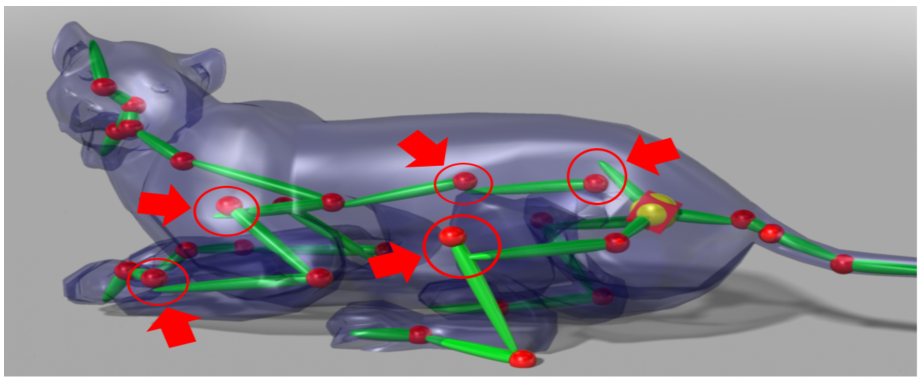
- Joint Constraint Term
  - Minimizing deviations of the joint locations after applying bone transformations

$$E_J = \frac{1}{S} \sum_{(j,k)\in\mathbb{S}} \sum_{t=1}^{S} \left\| \left([\mathbf{R}_{t,j}|\mathbf{T}_{t,j}] - [\mathbf{R}_{t,k}|\mathbf{T}_{t,k}] \right) \begin{bmatrix} \mathbf{o}_{jk} \\ 1 \end{bmatrix} \right\|_2^2$$

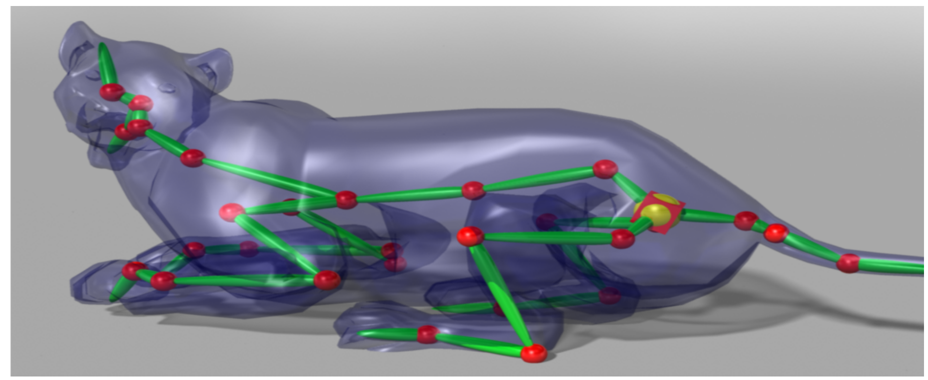


Recovering articulated object models from 3D range data [Anguelov et al. 2004]

Multi-pass: Joint Constraint



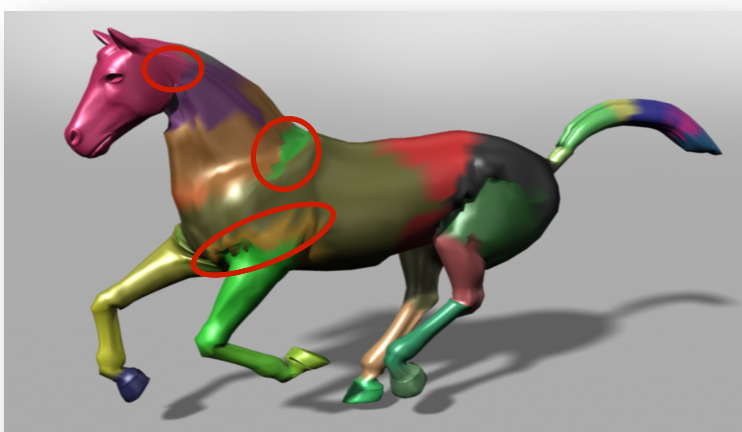
No Joint Constraint



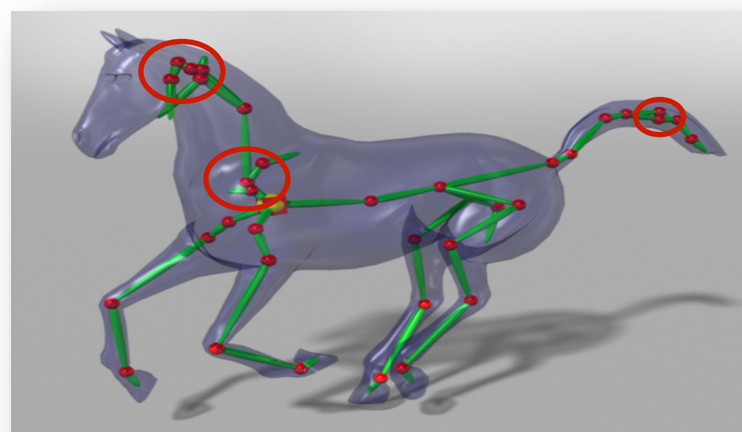
With Joint Constraints

Multi-pass: Skeleton Pruning

- Over-completed clustering (for bones) at the initialization step
  - Difficult to avoid (true # of bones is unknown)
  - Lead to unnecessary redundant bones in the skeleton



Initialization



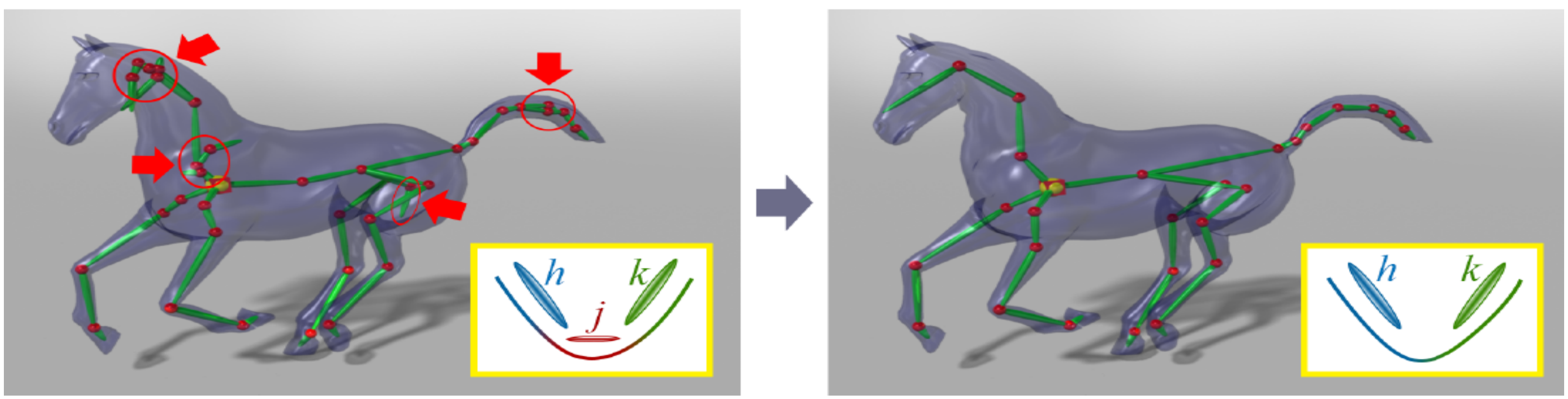
Skeleton

Multi-pass: Skeleton Pruning

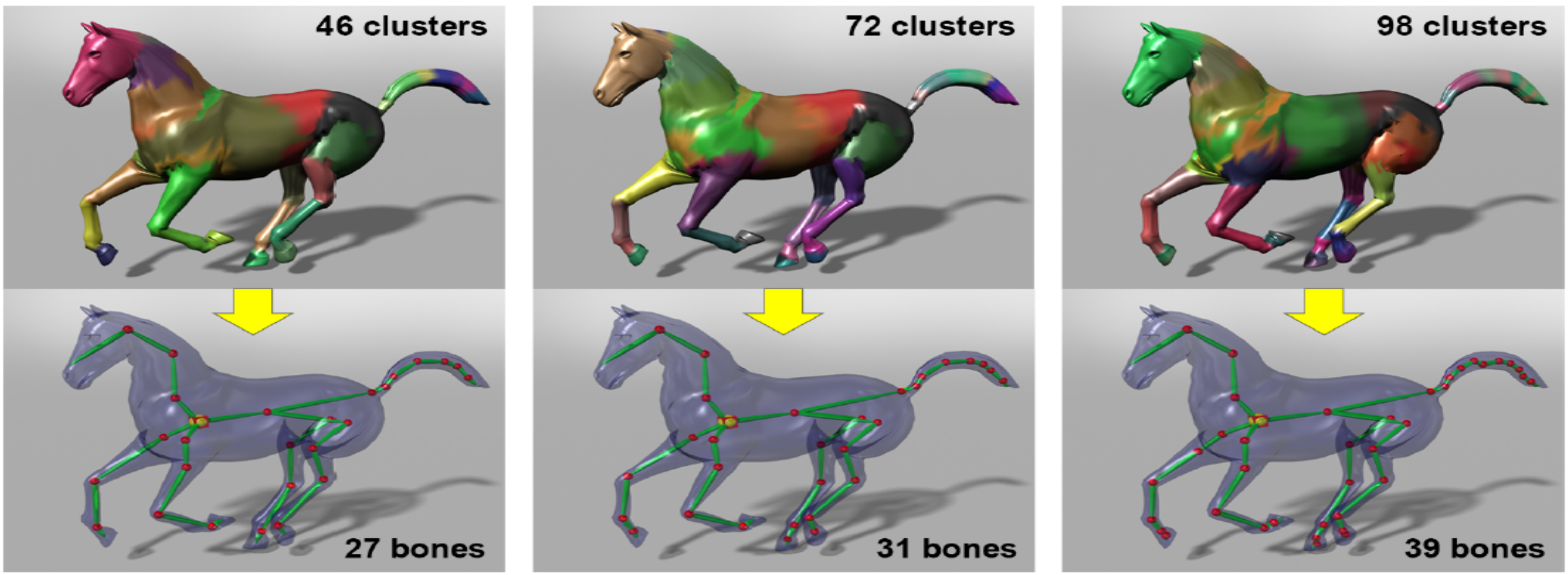
With weights smoothness term

$$\text{Remove bone } j \text{ if } \sum_{i=1}^{n} {w_{ij}}^2 < 10^{-2} M$$

$$\text{where: } M = \max_k \left\{ \sum_{i=1}^{n} {w_{ik}}^2 \right\}$$

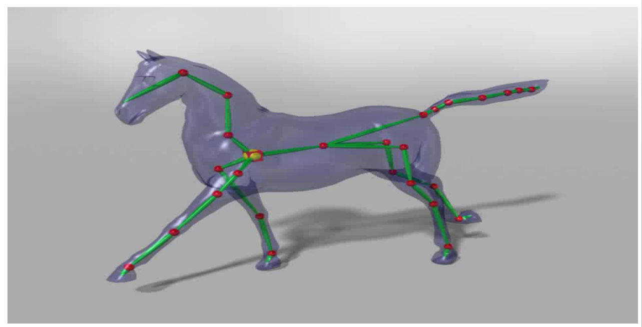


Multi-pass: Skeleton Pruning

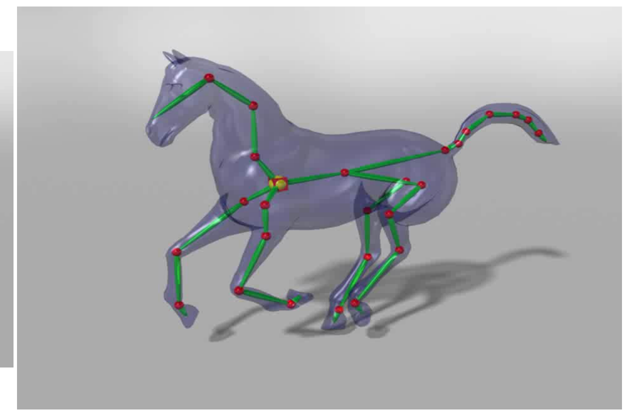


Multi-pass: Results

Input (48 poses)



Skeleton (27 bones)



Skinning Weights

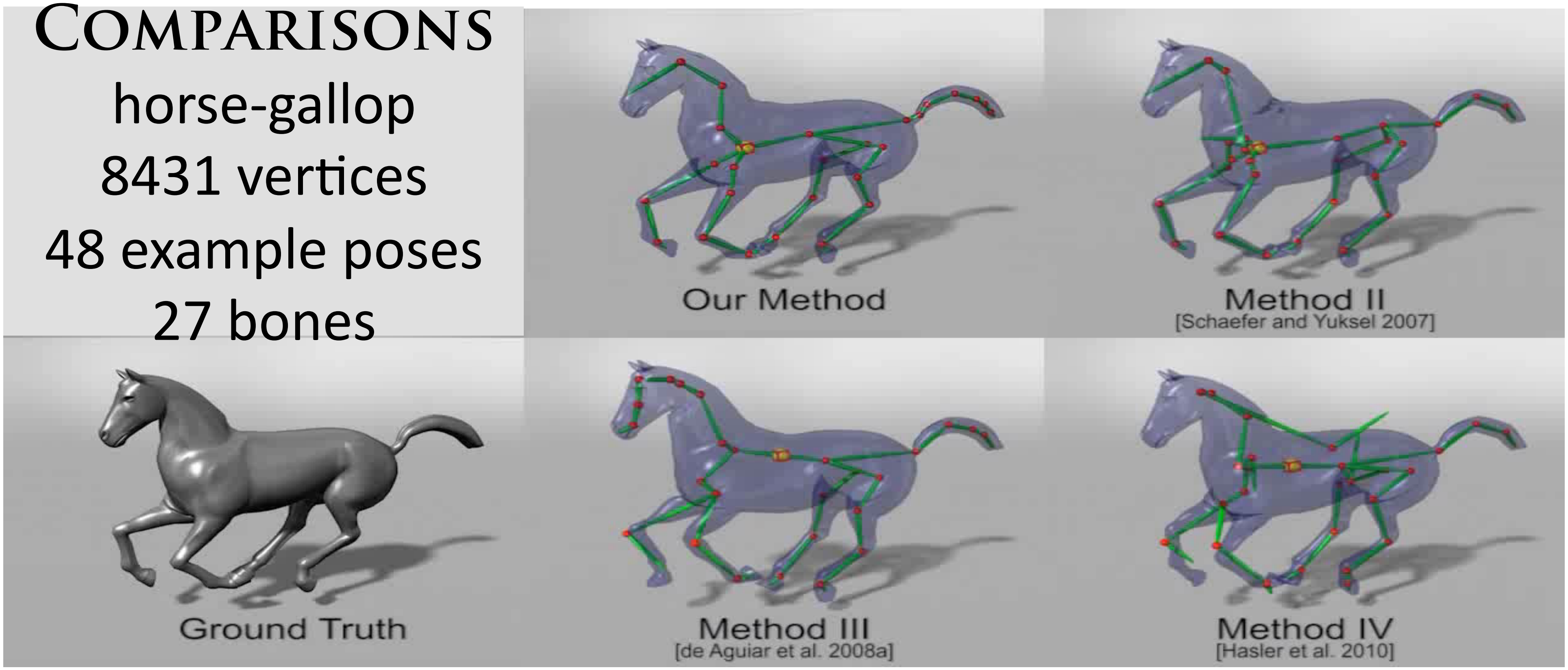
Comparisons

horse-gallop

8431 vertices

48 example poses

27 bones

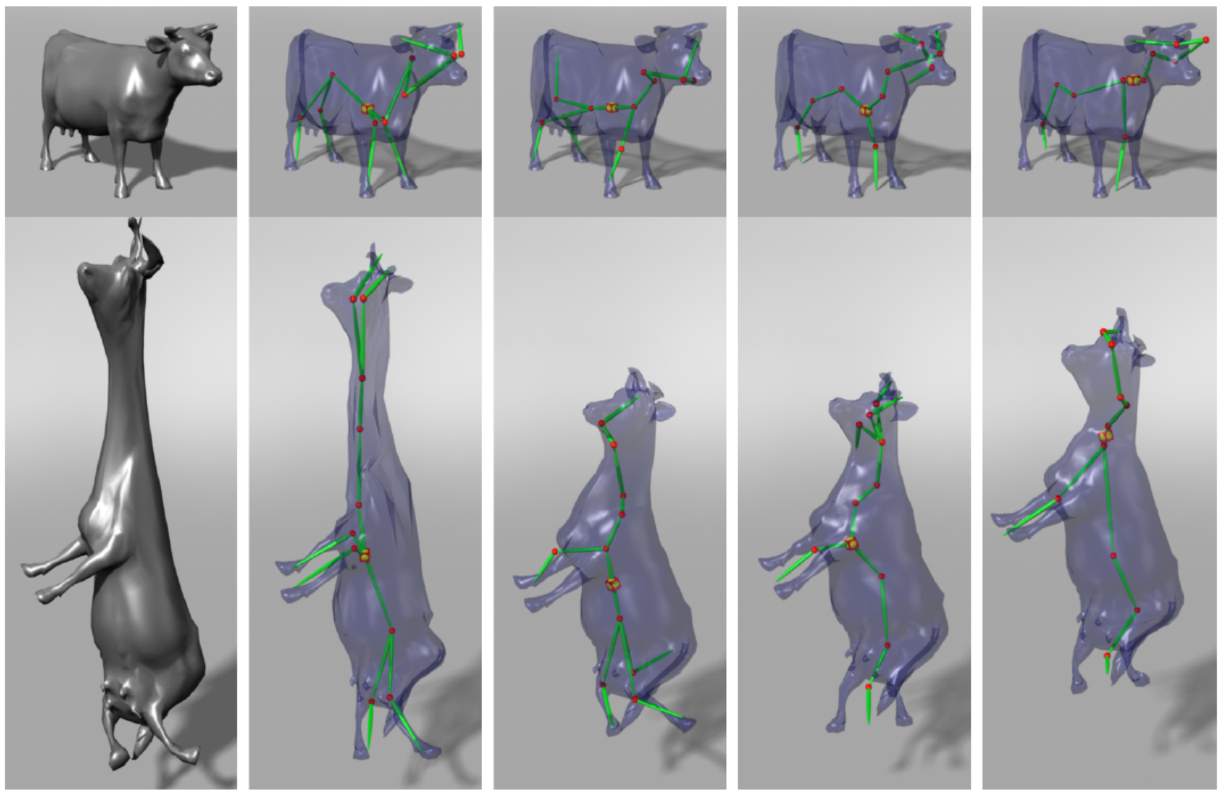


Comparisons

Dataset	N	F	B	Method I		Method II		Method III		Method IV	
				Time	RMSE	Time	RMSE	Time	RMSE	Time	RMSE
cat-poses	7207	9	28	5.8	**0.25**	0.1	0.68	6.9	1.04	17.2	0.63
horse-poses	8431	10	27	7.7	**0.21**	0.2	0.54	6.2	1.24	20.0	0.75
lion-poses	5000	9	30	4.1	**0.27**	0.1	0.83	4.0	1.62	11.7	1.14
horse-gallop	8431	48	27	41.9	**0.22**	0.8	0.44	33.3	1.10	80.3	0.88
hand	7997	43	18	65.1	**0.18**	0.6	0.23	20.0	0.42	41.9	**0.18**
dance	7061	201	16	148.7	**0.22**	2.5	0.76	61.8	0.78	168.0	0.53
scape	12500	70	23	252.1	**0.42**	1.7	1.03	60.7	1.18	410.4	1.24
samba	9971	175	22	348.2	**0.56**	3.3	1.29	95.1	1.57	296.0	1.79
cow	2904	204	11	72.3	**1.52**	1.0	5.41	16.0	5.61	47.9	5.58

Lowest RMSE

Comparisons



Ground Truth

Method I
[Le and Deng 2014]

Method II
[Schaefer and Yuksel 2007]

Method III
[de Aguiar et al. 2008a]

Method IV
[Hasler et al. 2010]

Part I:
Example-based Skinning Decomposition

Part II:
Example-based Skeleton Extraction

Part III:
Skinning Weights Reduction & Compression

Skinning Weights Reduction

- Weights in linear surface deformation
  - Non-negativity and/or affinity constraints
  - Locality
    - A small number of bones (or control points) per vertex
    - Weights decrease w.r.t. distance between vertex and the bone (or control point).

$$\mathbf{v}'_i = \sum_j w_{ij}\mathbf{c}_j$$

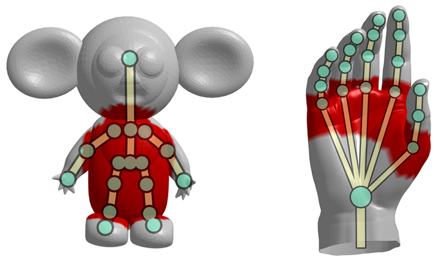
(cage-based deformation)

$$\mathbf{v}'_i = \sum_{j=1}^{m} w_{ij}[\mathbf{R}_j|\mathbf{T}_j]\mathbf{v}_i$$

(linear blend skinning)

Skinning Weights Reduction

- Discrete optimization: $|\{w_{ij}|w_{ij} \neq 0\}| \leq |K|, \forall i$
  - Difficult to find optimum solution
  - High pay-off for non-optimum solution
    - Fracture
    - Significant increase of computing cost: nK non-zero → $n(K+1)$ non-zero
- Pros and Cons
  - Speed up skinning efficiency significantly (in particular, GPUs)
  - Loss of skinning visual quality (in particular, exceptional vertices)



exceptional vertices

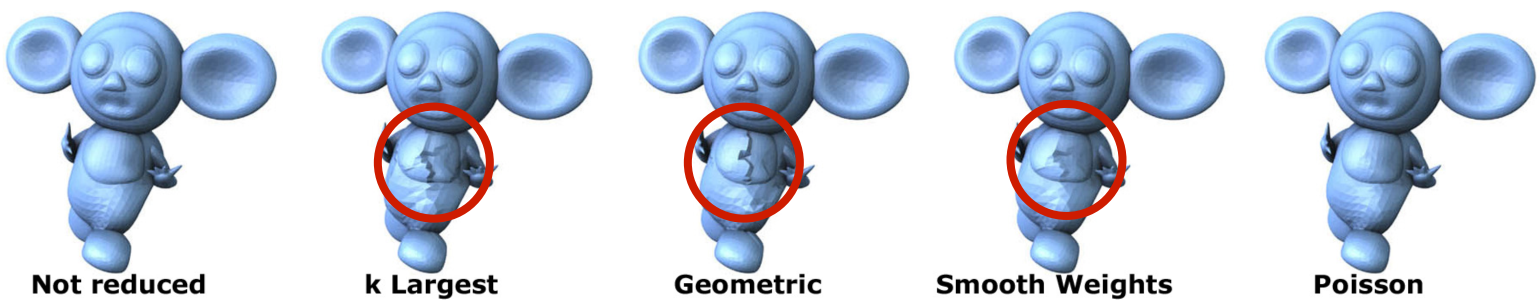
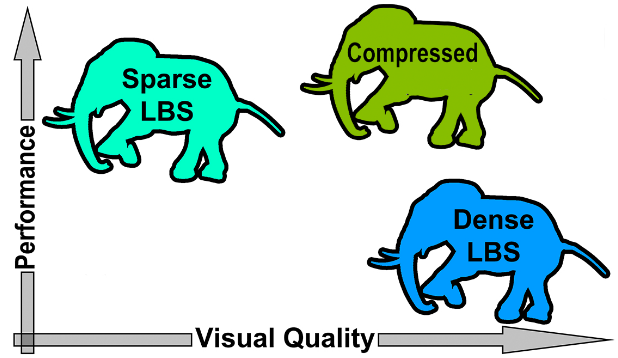


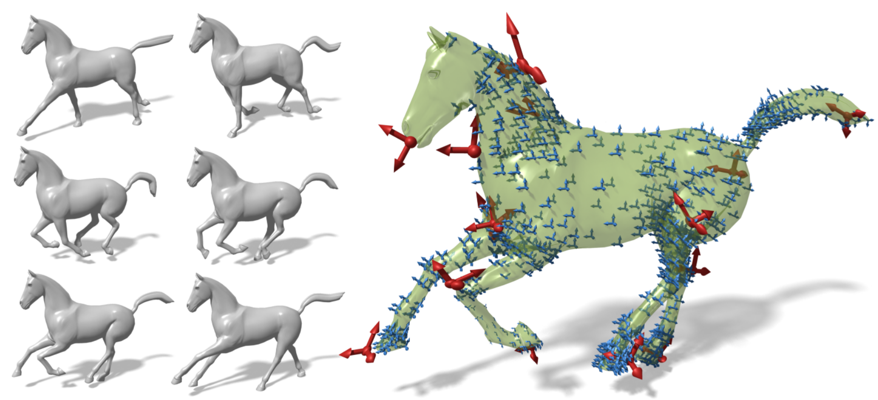
Image courtesy of [Landreneau and Schaefer 2010], @EG 2010

Two-layer Sparse Compression of Dense Skinning Weights

- Positioning
  - Speed up skinning efficiency
  - Maximally maintain visual quality

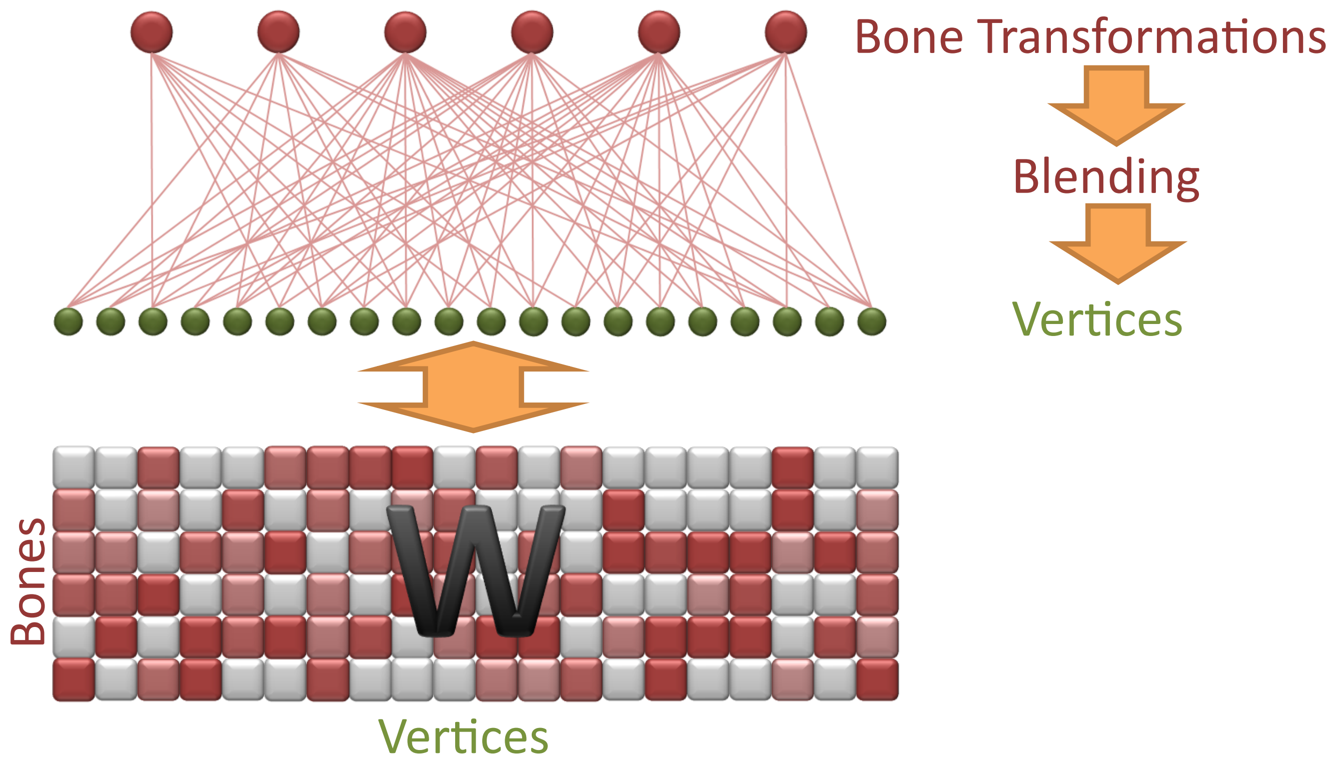


- Idea
  - Two-layer blending
  - Caching similar blending operations



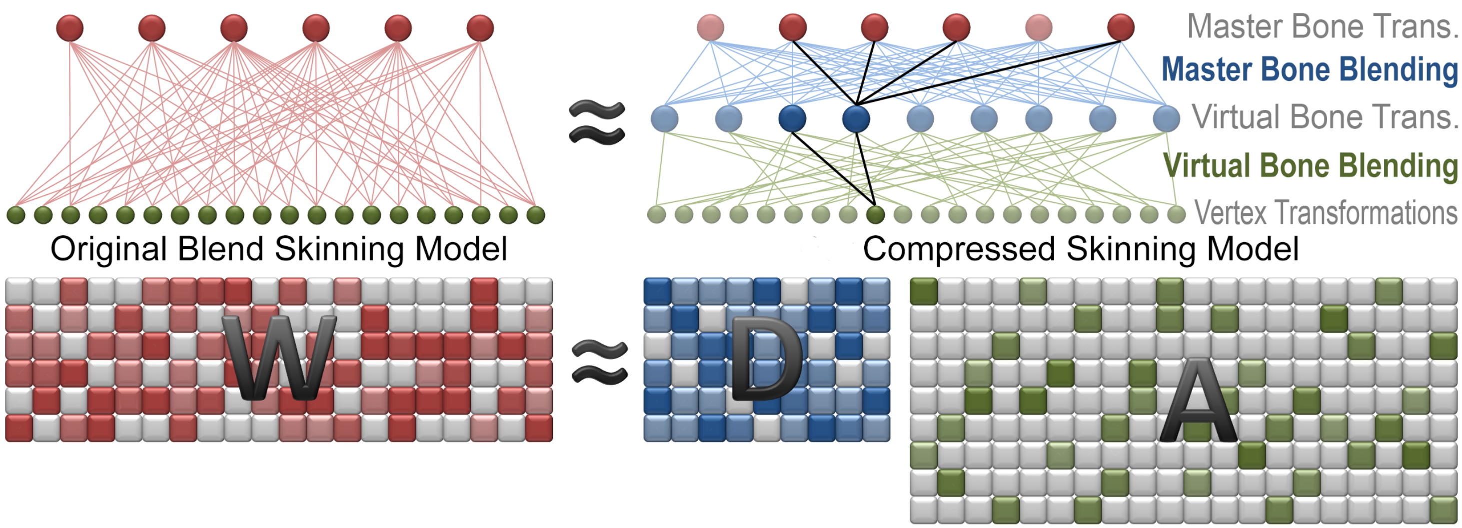
Two-layer Sparse Compression

- **<u>Input:</u>** Skinning weights → dense matrix

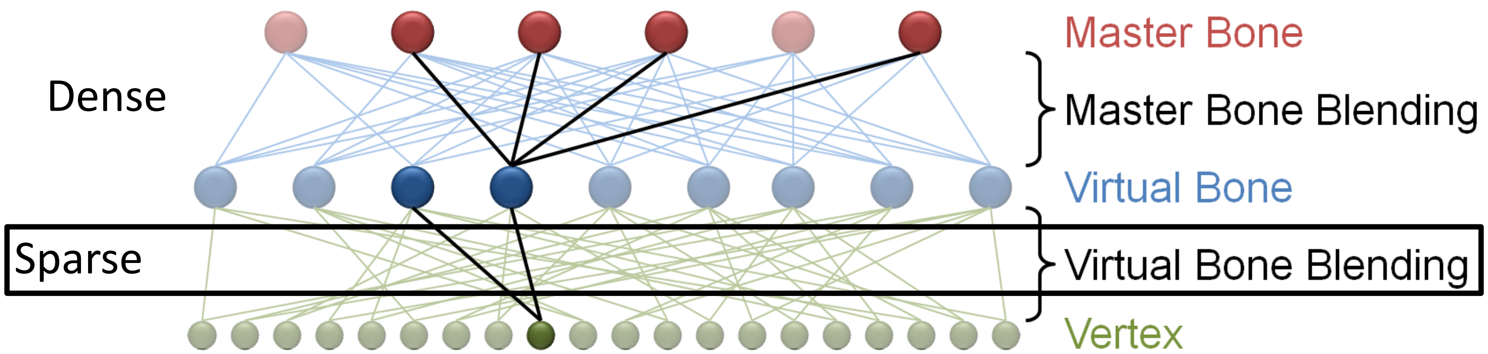


Two-layer Sparse Compression

- **Compression:** $\mathbf{W} \approx \mathbf{DA}$
  - Dictionary $\mathbf{D}$, and sparse coefficients $\mathbf{A}$
  - $\text{card}(\mathbf{D}) + \text{card}(\mathbf{A}) << \text{card}(\mathbf{W})$

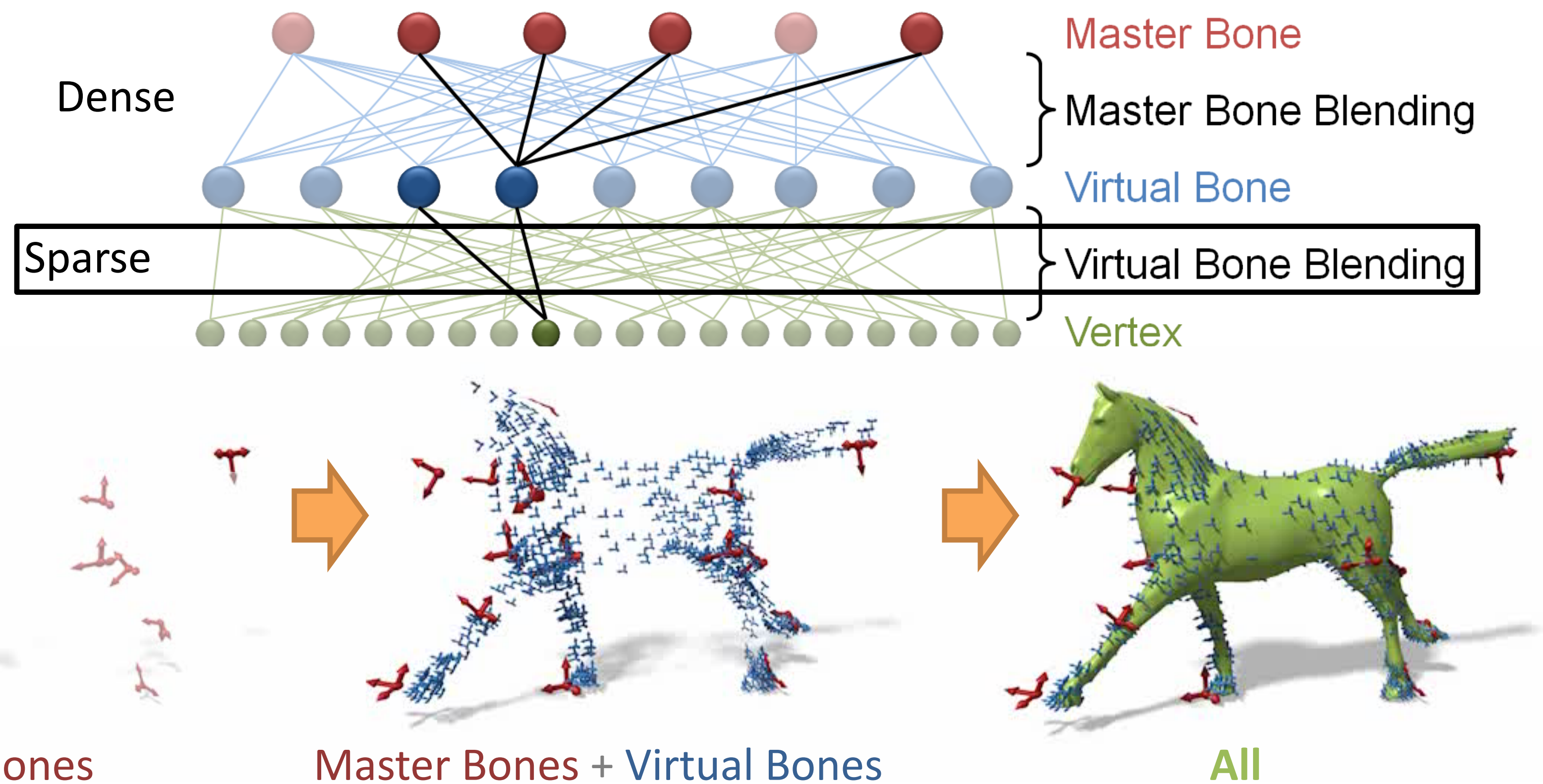


Two-layer Sparse Compression

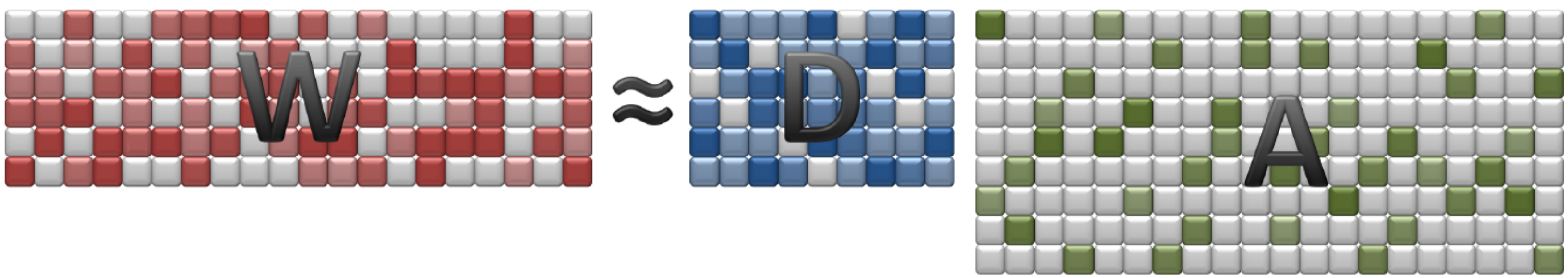


- **Virtual Bones**
  - Cache similar transformation
  - Use as basis
- **Sparse Virtual Bone Blending**
  - Keep the number of bones/vertex as small as possible (**2 bones/vertex**)

Two-layer Sparse Compression



Two-layer Sparse Compression



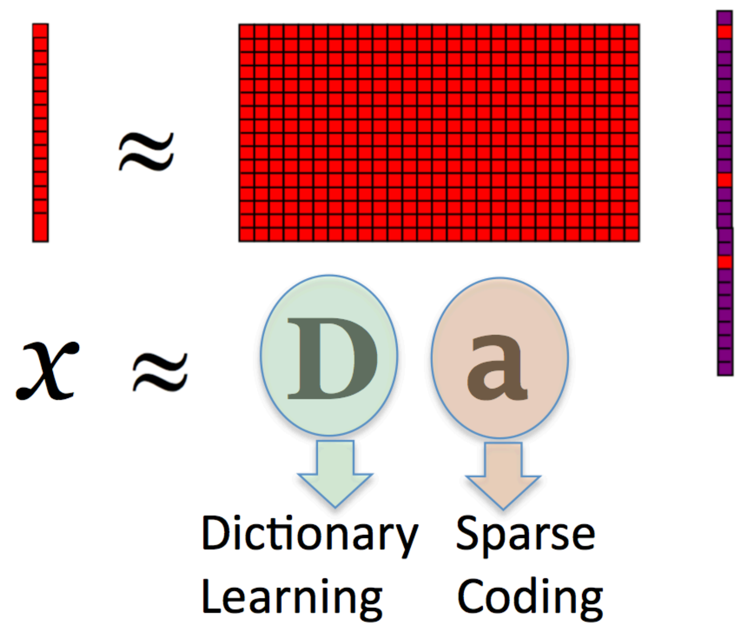
$$\min_{D,A} {\Delta_W}^2 = \min_{D,A} \frac{1}{kn} \|DA - W\|_F^2$$

Subject to: $\mathbf{card}(d_i) \le c, \forall i$ $\quad c = \max\{\mathbf{card}(w_i)\}+1$

$\mathbf{card}(\alpha_i) \le 2, \forall i$ $\quad \begin{cases} n \text{ is very large} \\ \mathbf{card}(\mathbf{A}) = 2n \rightarrow \min \end{cases}$

Preliminaries: Sparse Decomposition

- Definition
  - Given multi-dimensional observed data and a dictionary matrix, estimate a sparse vector that satisfies a linear system of equations,
- Applications
  - Image processing, audio processing, data compression, denoising, etc.



Preliminaries: Sparse Coding

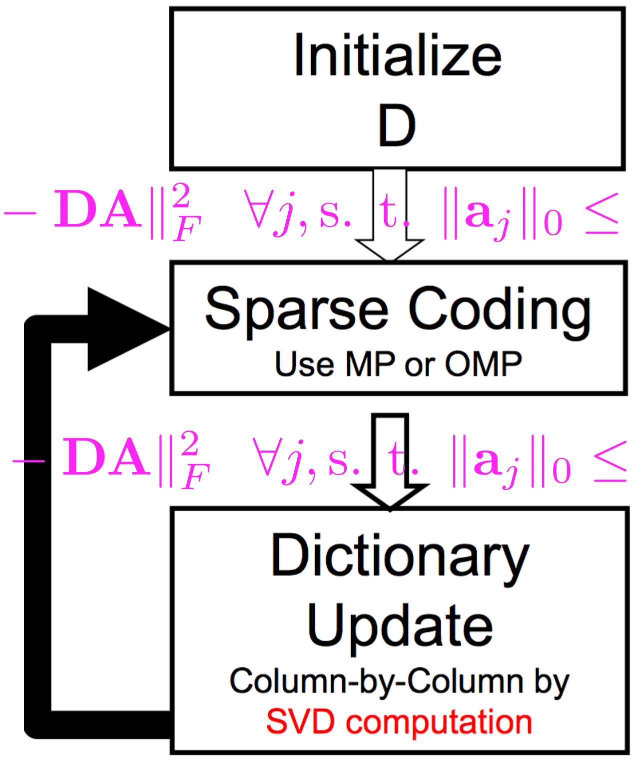
- Definition
  - Solving the optimal sparse vector ***a*** given the dictionary **D**
  - L_0 norm and L_1 norm regularization
- Algorithms
  - Matching pursuit algorithm [Mallat and Zhang 1993]
  - Orthogonal matching pursuit algorithm [Tropp and Gilbert 2007, Cai and Wang 2011]

$$\min_{\mathbf{a}\in\mathbb{R}^m} \frac{1}{2}\|\mathbf{x}-\mathbf{D}\mathbf{a}\|_2^2 \qquad \text{subject to} \ \ \|\mathbf{a}\|_0 \leq L$$

Sparse coding with L_0 norm regularization

Preliminaries: Dictionary Learning

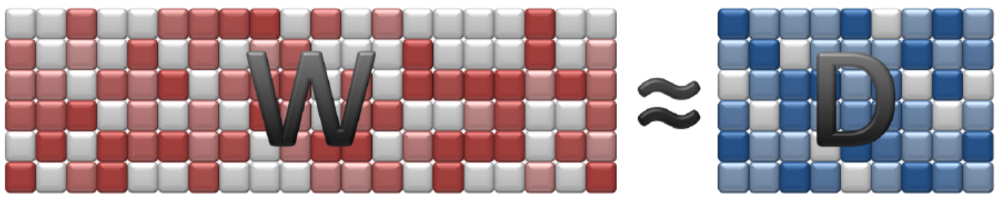
- Definition
  - Solving the optimal Dictionary **D** given the sparse vector ***a***
- Algorithms
  - K-SVD for L_0 norm regularization [Aharon et al. 2006]
  - Online dictionary learning for L_1 norm regularization [Mairal et al. 2010]

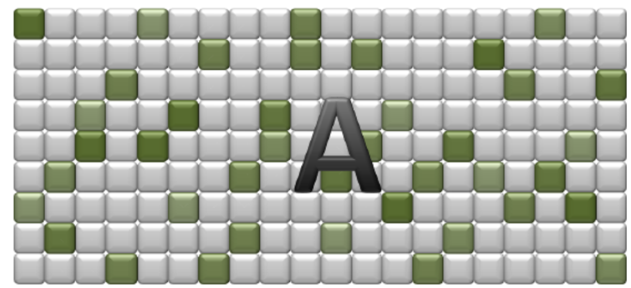


$$\min_{\mathbf{A}} \|\mathbf{X} - \mathbf{DA}\|_F^2 \quad \forall j, \text{s. t. } \|\mathbf{a}_j\|_0 \leq L$$

$$\min_{\mathbf{D}} \|\mathbf{X} - \mathbf{DA}\|_F^2 \quad \forall j, \text{s. t. } \|\mathbf{a}_j\|_0 \leq L$$

Two-layer: Sparse Matrix Decomposition Solver





- Alternative update $\mathbf{D}$ and $\mathbf{A}$ (Block coordinate descent)

Two-layer: Update Dictionary **D**

- Online learning with warm restart [Mairal et al. 2010]

$$\Phi = \sum_{i=1}^{n} \mathbf{a}_i \mathbf{a}_i^{\mathsf{T}} = [\phi_1, \ldots, \phi_m] \in \mathbb{R}^{q \times q}$$

$$\Gamma = \sum_{i=1}^{n} \mathbf{w}_i \mathbf{a}_i^{\mathsf{T}} = [\gamma_1, \ldots, \gamma_m] \in \mathbb{R}^{m \times q}$$

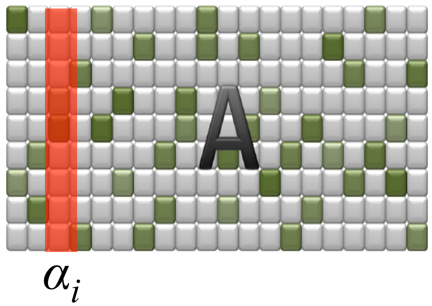
$$\boxed{\mathbf{d}_j \leftarrow \frac{1}{a_{jj}} (\gamma_j - \mathbf{D}\phi_j) + \mathbf{d}_j}$$

– Result d_j is normalized to have the unit length

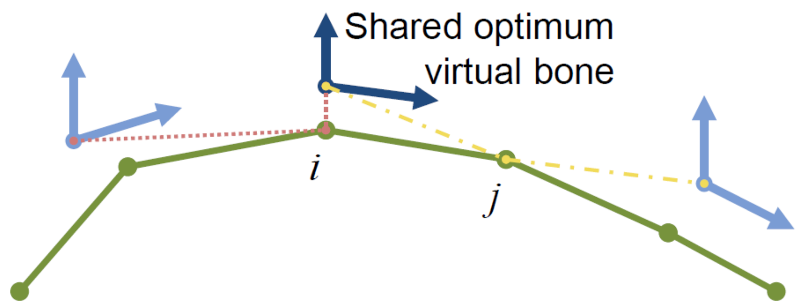
Two-layer: Update Coefficients **A**

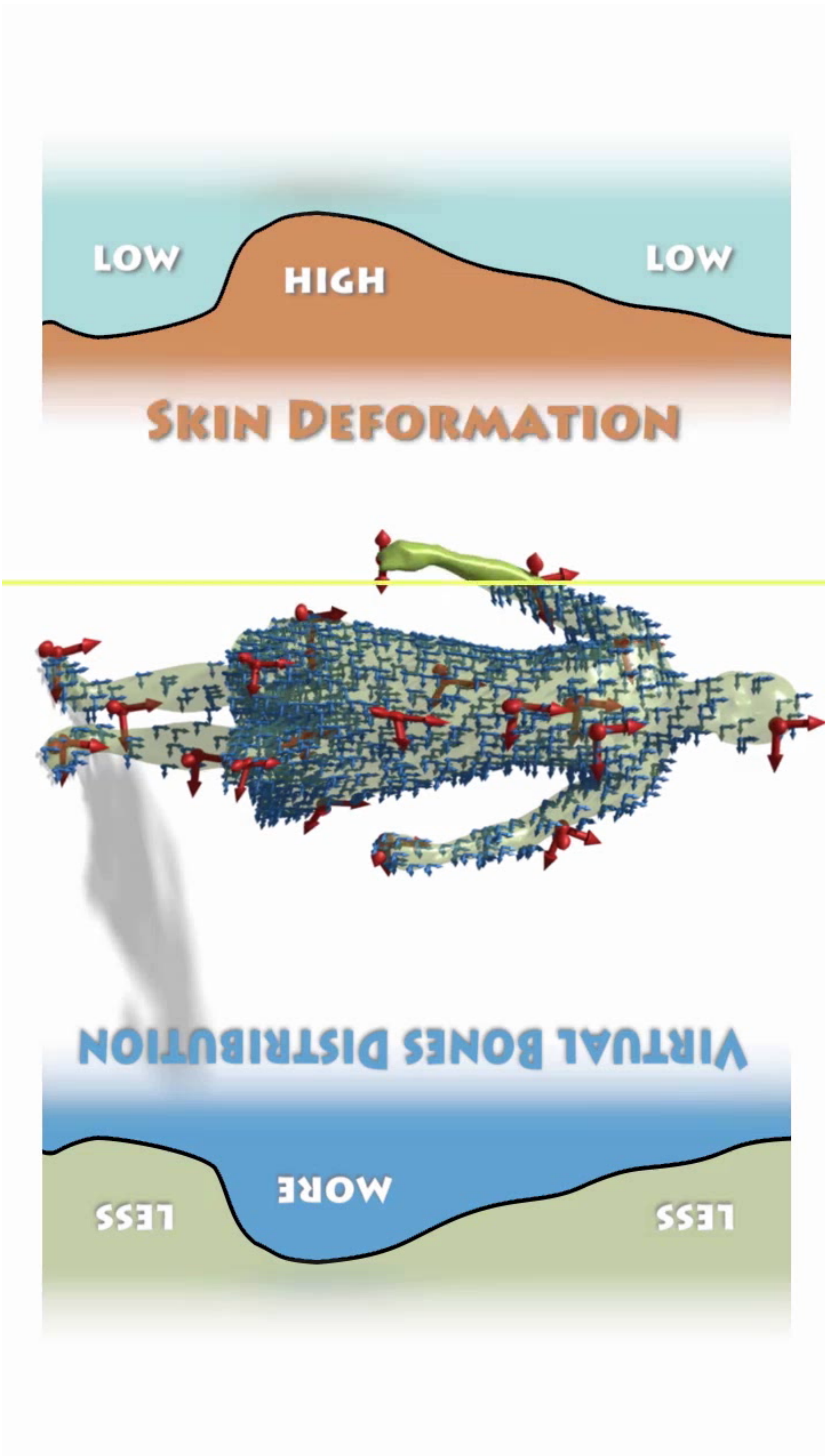
- (per vertex) Linear least squares with 2 unknowns

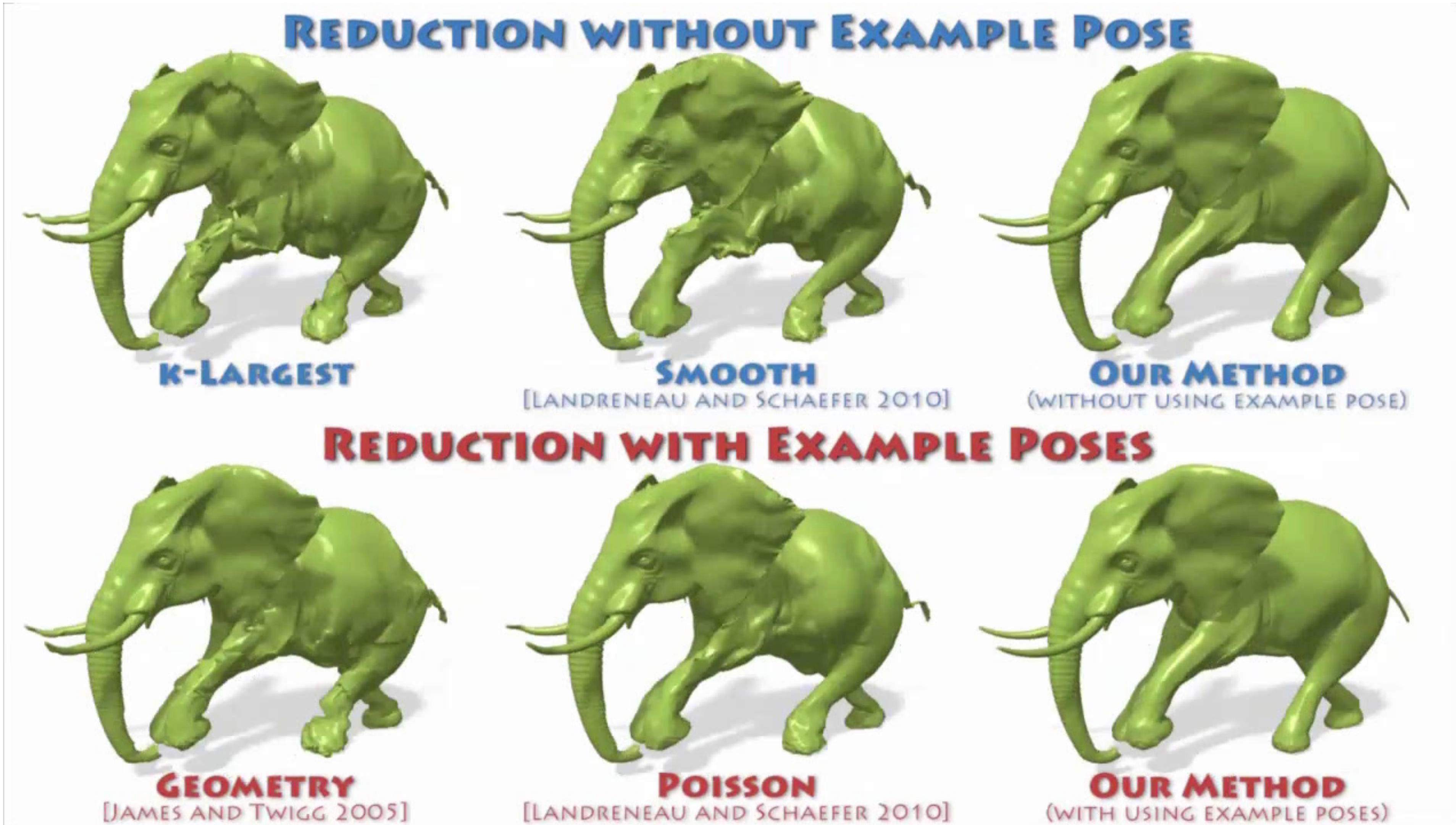
$$\min_{\substack{(\alpha_i)_r\\(\alpha_i)_s}} \|d_r(\alpha_i)_r + d_s(\alpha_i)_s - w_i\|_2^2 \text{ s.t. } (\alpha_i)_r + (\alpha_i)_s = 1$$

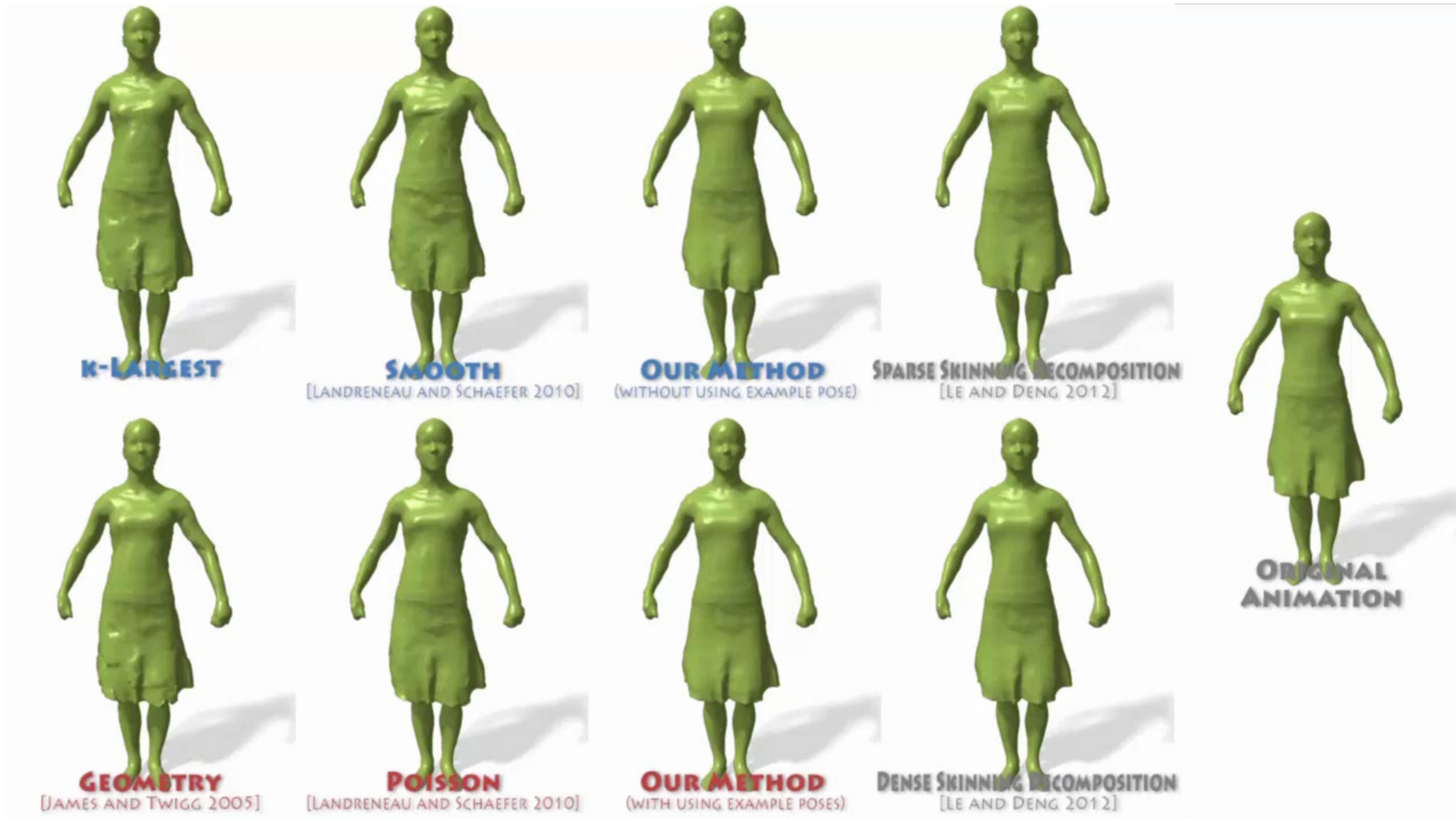


- Use mesh smoothness assumption to quickly find the non-zero candidates (virtual bones)




LOW
HIGH
LOW
SKIN DEFORMATION
VIRTUAL BONES DISTRIBUTION
LESS
MORE
LESS


Reduction without Example Pose
k-Largest
Smooth
[Landreneau and Schaefer 2010]
Our Method
(without using example pose)
Reduction with Example Poses
Geometry
[James and Twigg 2005]
Poisson
[Landreneau and Schaefer 2010]
Our Method
(with using example poses)


k-Largest
Smooth
[Landreneau and Schaefer 2010]
Our Method
(without using example pose)
Sparse Skinning Decomposition
[Le and Deng 2012]
Original
Animation
Geometry
[James and Twigg 2005]
Poisson
[Landreneau and Schaefer 2010]
Our Method
(with using example poses)
Dense Skinning Decomposition
[Le and Deng 2012]

Summary

- Example-based skinning decomposition
  - Inverse problem of LBS, extracting the LBS model from a small set of example poses
  - Handling joint constraint for skeleton extraction
- Skinning weight reduction and compression
  - Speed up skinning efficiency esp. on GPUs
  - Balance the trade-off between efficiency and quality

Thank you for attention!

- Executable code/files of
  - Smooth skinning decomposition with rigid bones [Le and Deng 2012],
  - Skeletal rigging from mesh sequences [Le and Deng 2014],
  - two-layer sparse compression of dense skinning weights [Le and Deng 2013]

  are available at the link below.

http://graphics.cs.uh.edu/ble/progs/skinning/