UC San Diego

Lecture 10:

3D Parametric Models

Hao Su and Ronald Yu

Feb 1, 2018

Why 3D Deep Learning

- To make money
  - Extra data
- How to relate 2D images with the 3D world that we live in?
- Can we exploit structures and patterns that exist in 3D data but not in images?
  - Geometric Properties
  - 3D data contains structures
  - 3D structures physically should make sense
  - It’s easier to describe these physical structures

Today's Lecture

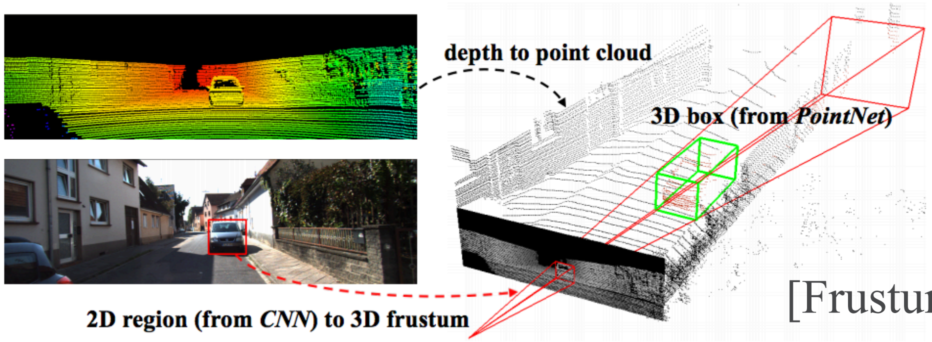
- 3D Mesh Parameterization and Manipulation
  - What makes mesh manipulation possible?
- 3D Parametric Models and Applications
  - Use several tricks
  - Consider what motivates these tricks
  - Can we automate these tricks?
- Brief overview on some of the cuter recent deep learning research using parametric models (2015-Dec. 2017)
  - Research uses these tricks
  - Can we have research that learns tricks?

Last time

- Learning on unstructured 3D data (point clouds)
- Pros:
  - Easy to capture (depth scanner, multi-view)
  - Generalizable (easy to generate point clouds regardless of data)
  - Simple data structure
  - Great for deep learning!
  - Interesting AI Problems:
    - How can AI understand the 3D world?
    - How can AI relate 2D images to the 3D world?

Last time

- Learning on unstructured 3D data (point clouds)
- Pros:
  - Easy to capture (depth scanner, multi-view)
  - Generalizable (easy to generate point clouds regardless of data)
  - Simple data structure
  - Great for deep learning!
  - Useful application$ (e.g. $elf driving car$, robotic$)



[Frustum PointNet, arxiv 2017]

Drawbacks of Point Clouds

- Noisy and unstructured

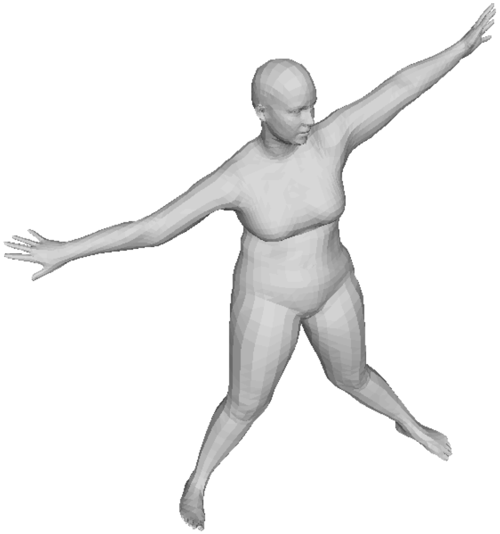
[FLAME, SIGGRAPH Asia 2017]

Drawbacks of Point Clouds

- Noisy and unstructured
- Lack of topology loses information

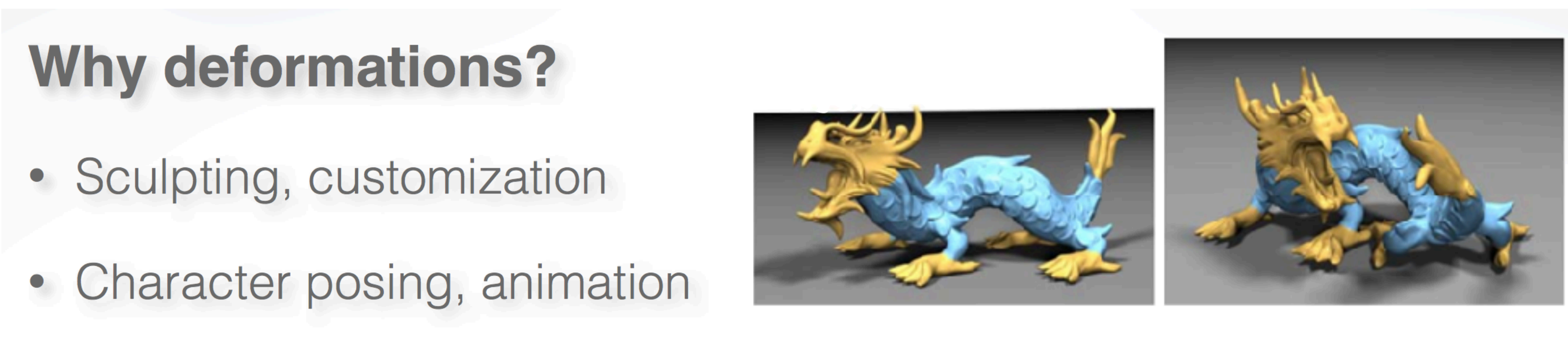
Drawbacks of Point Clouds

- Noisy and unstructured
- Lack of topology loses information



Drawbacks of Point Clouds

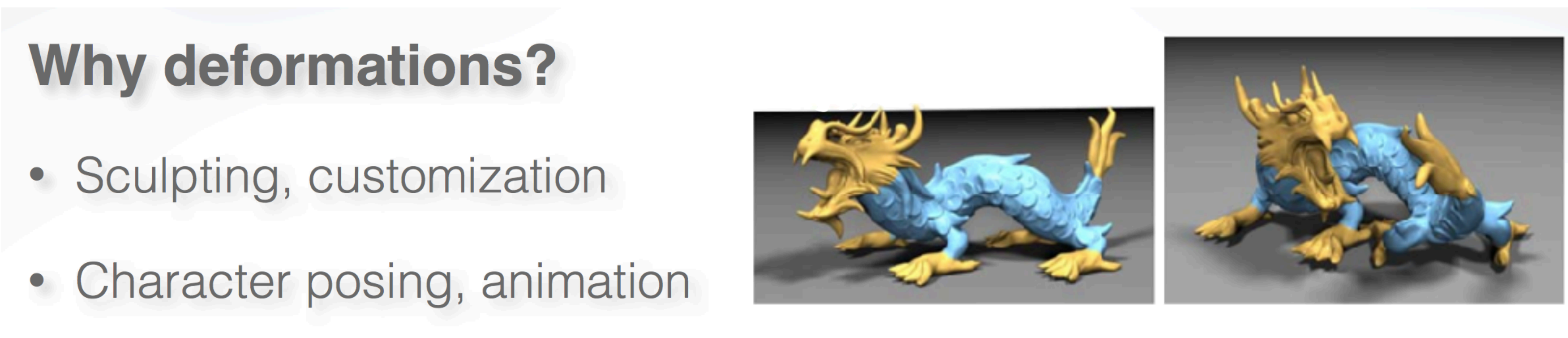
- Noisy and unstructured
- Lack of topology loses information
- Difficult to manipulate



[USC CSCI 621]

Drawbacks of Point Clouds

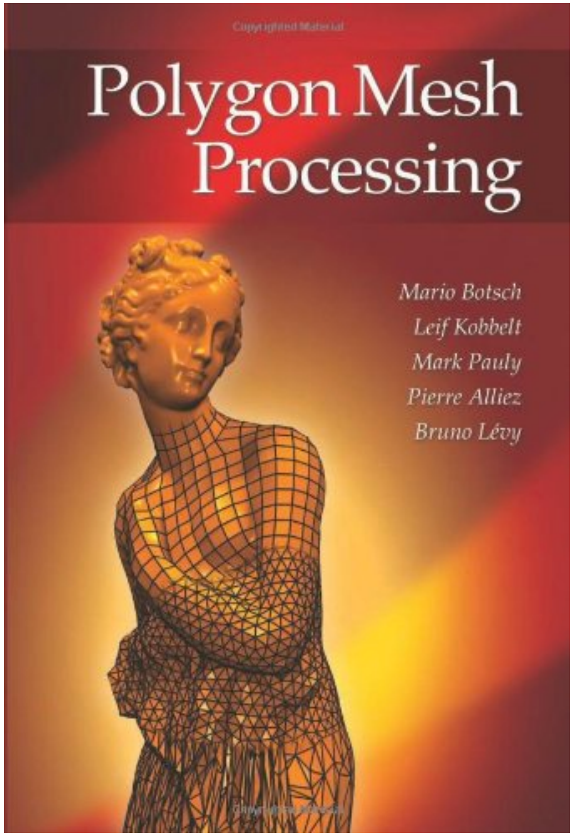
- Noisy and unstructured
- Lack of topology loses information
- Difficult to manipulate
- Hence, limited application$$$



[USC CSCI 621]

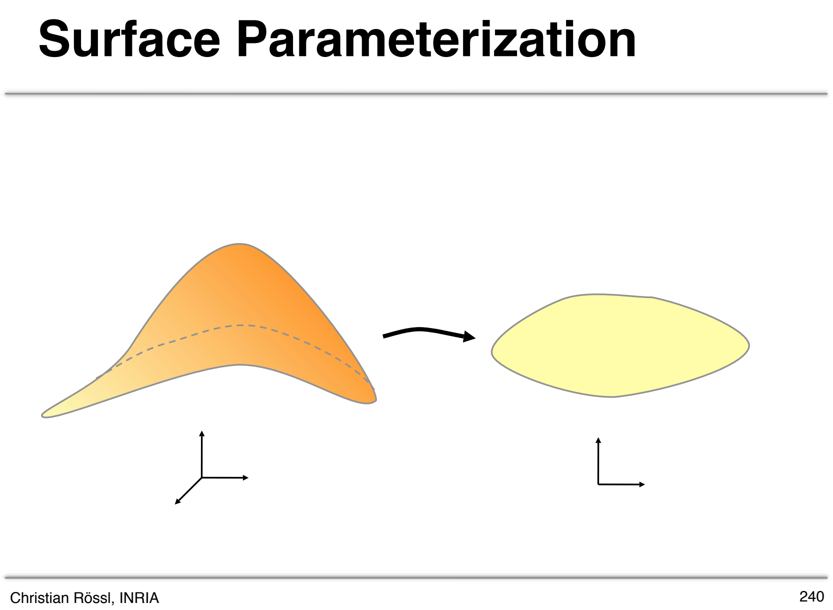
Polygon Meshes

- All mathematical equations and figures in these slides come from this book or its associated slides



Parameterized Meshes

- “a parameterization of a 3D surface is a function putting this surface in one-to-one correspondence with a 2D domain” - the Good Book
- Manipulable
- Consider Parameterization of A Family of Shapes:
  - Fixed Correspondences and Topology



Surface Parameterization

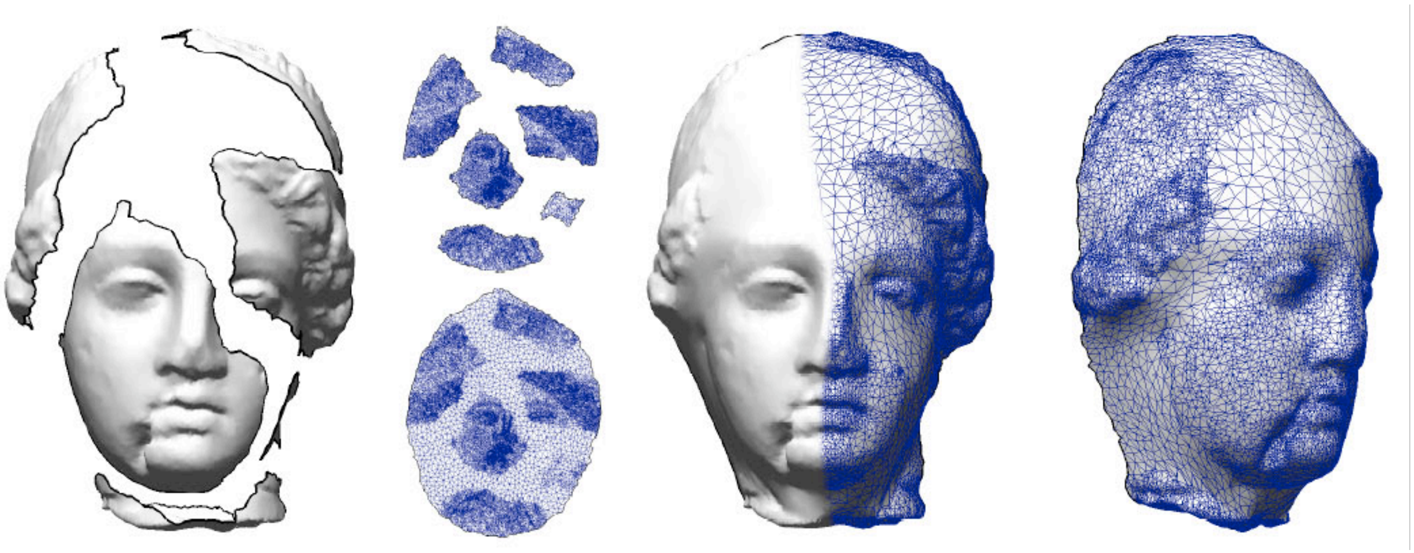
Mercator-Projektion

Mollweide-Projektion

[www.wikipedia.de]

Motivation

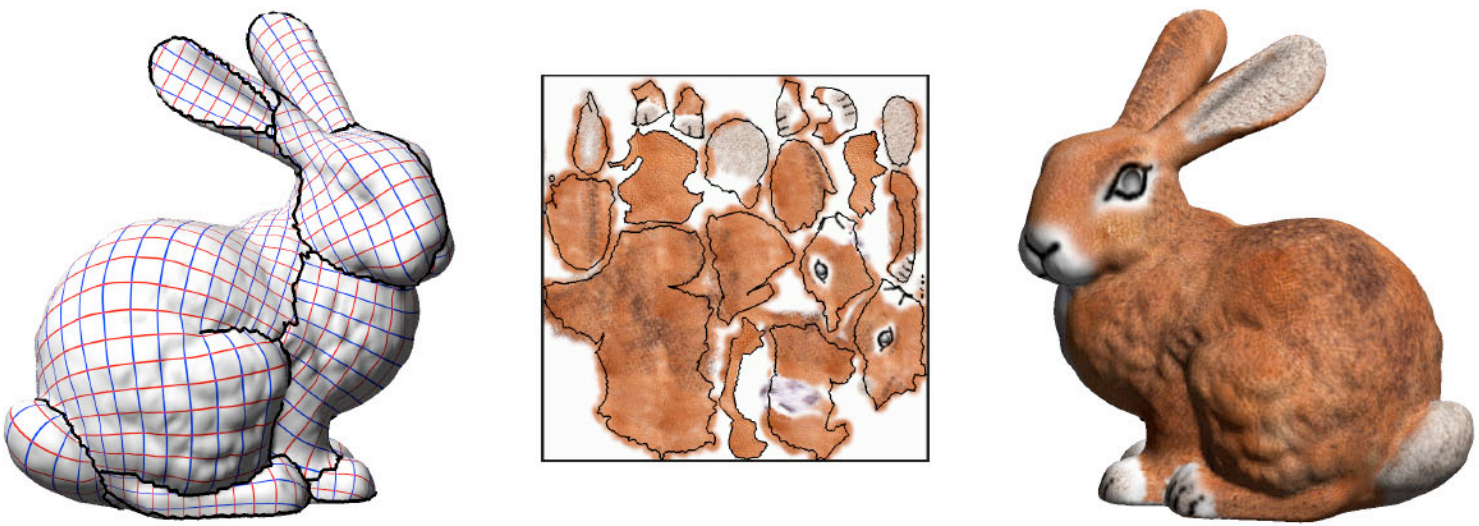
- Many operations are simpler on planar domain



Lévy: *Dual Domain Exrapolation*, SIGGRAPH 2003

Motivation

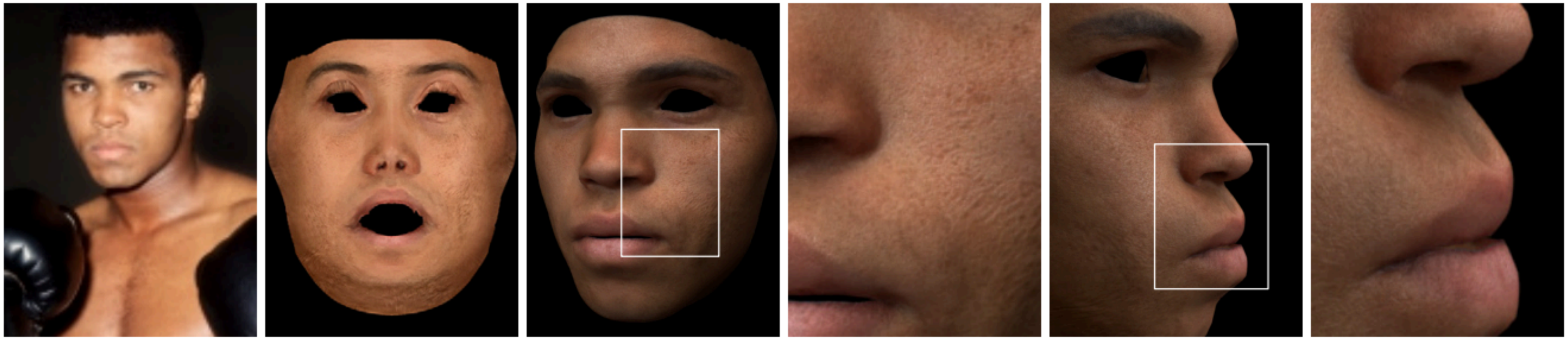
- Texture mapping



Lévy, Petitjean, Ray, and Maillot: *Least squares conformal maps for automatic texture atlas generation*, SIGGRAPH 2002

OMG Deep Learning!

- Sample application of texture space with deep learning
- Use a deep neural network to render high quality textures using low resolution images
- Style transfer techniques



input picture | output albedo map | rendering | rendering (zoom) | rendering | rendering (zoom)

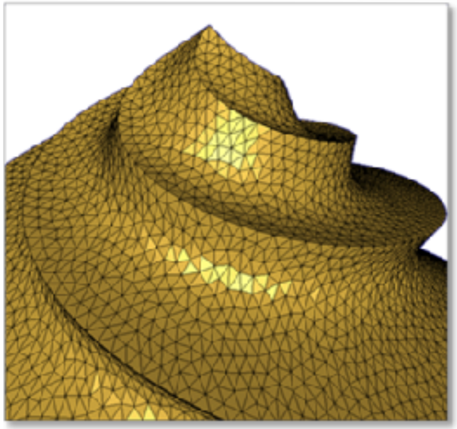
[Saito et al., CVPR 2017]

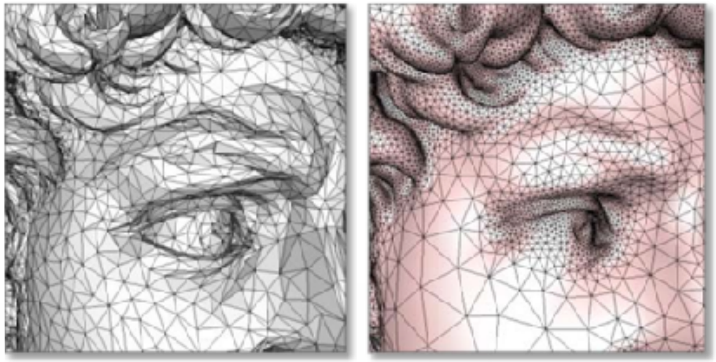
Discrete Surfaces: Point Sets, Meshes

[USC CSCI 621]

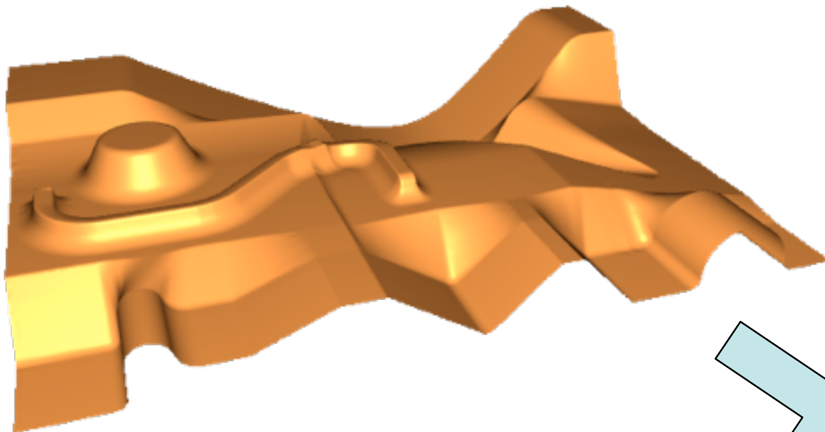
- Flexible
- Suitable for highly detailed scanned data
- No analytic surface
- No inherent "editability"

Mesh Editing

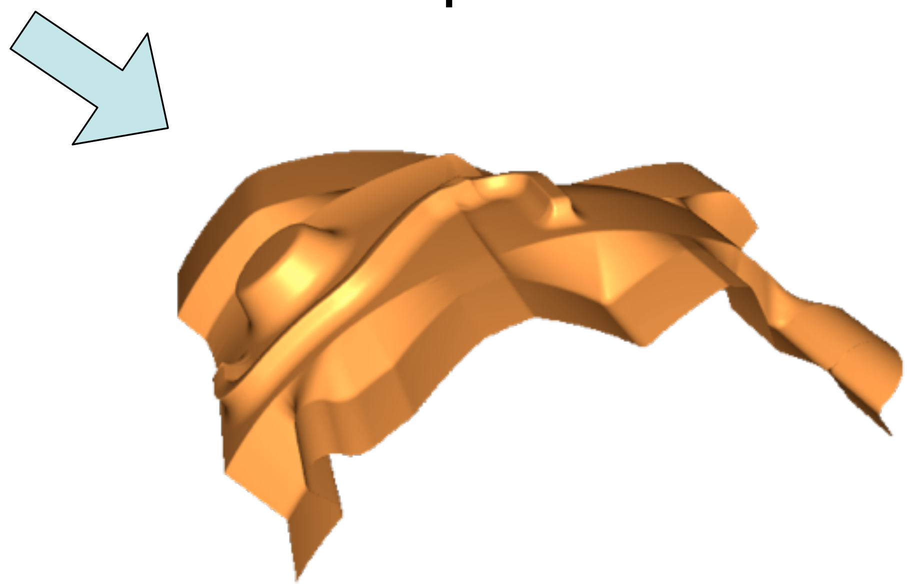




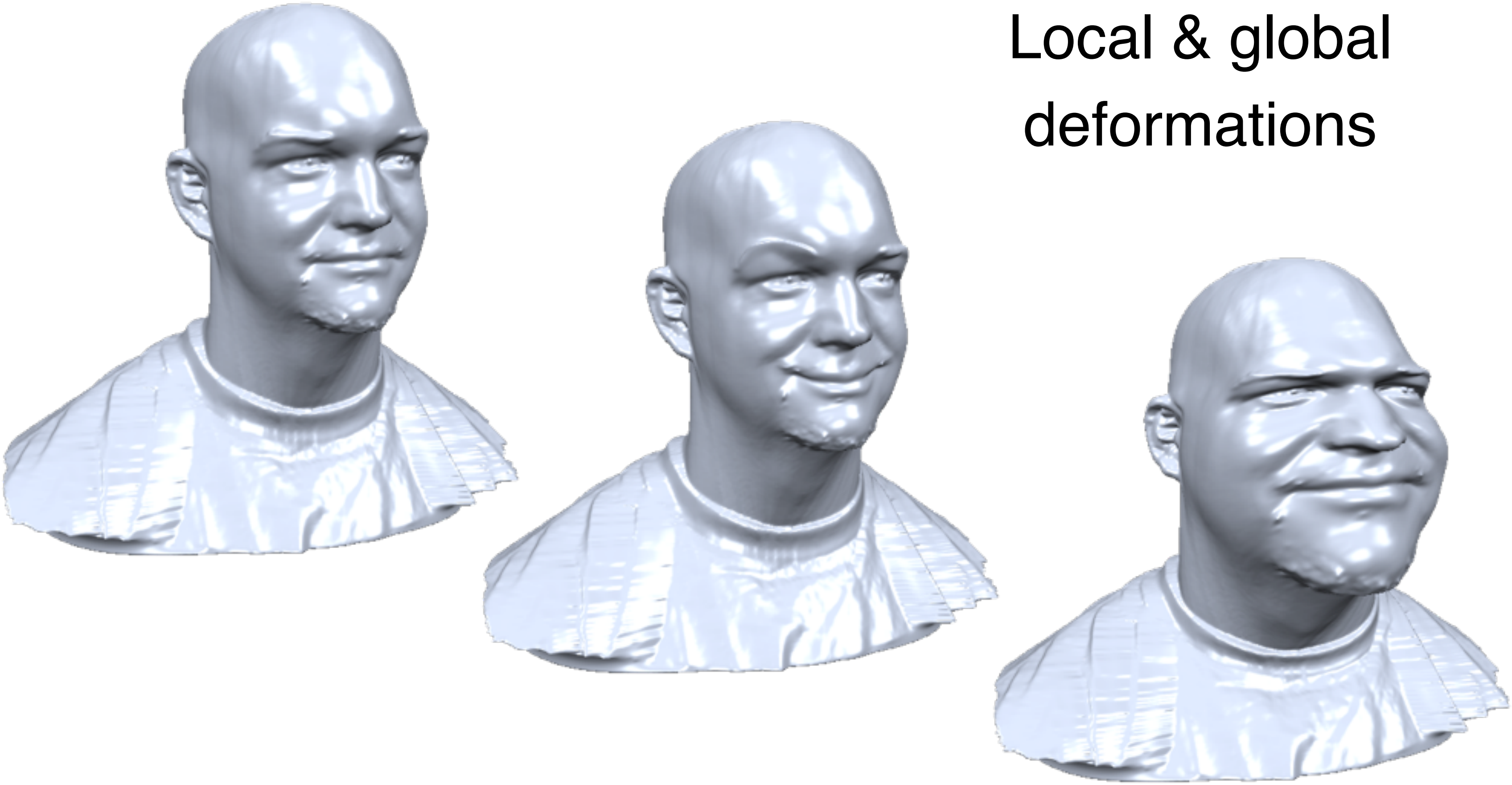
Mesh Deformation



Global deformation
with intuitive
detail preservation



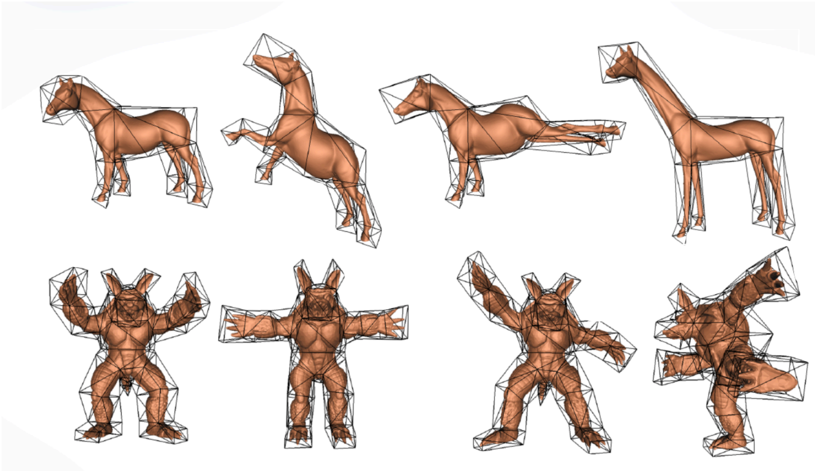
Mesh Deformation



Local & global deformations

Manipulating Parametric Models

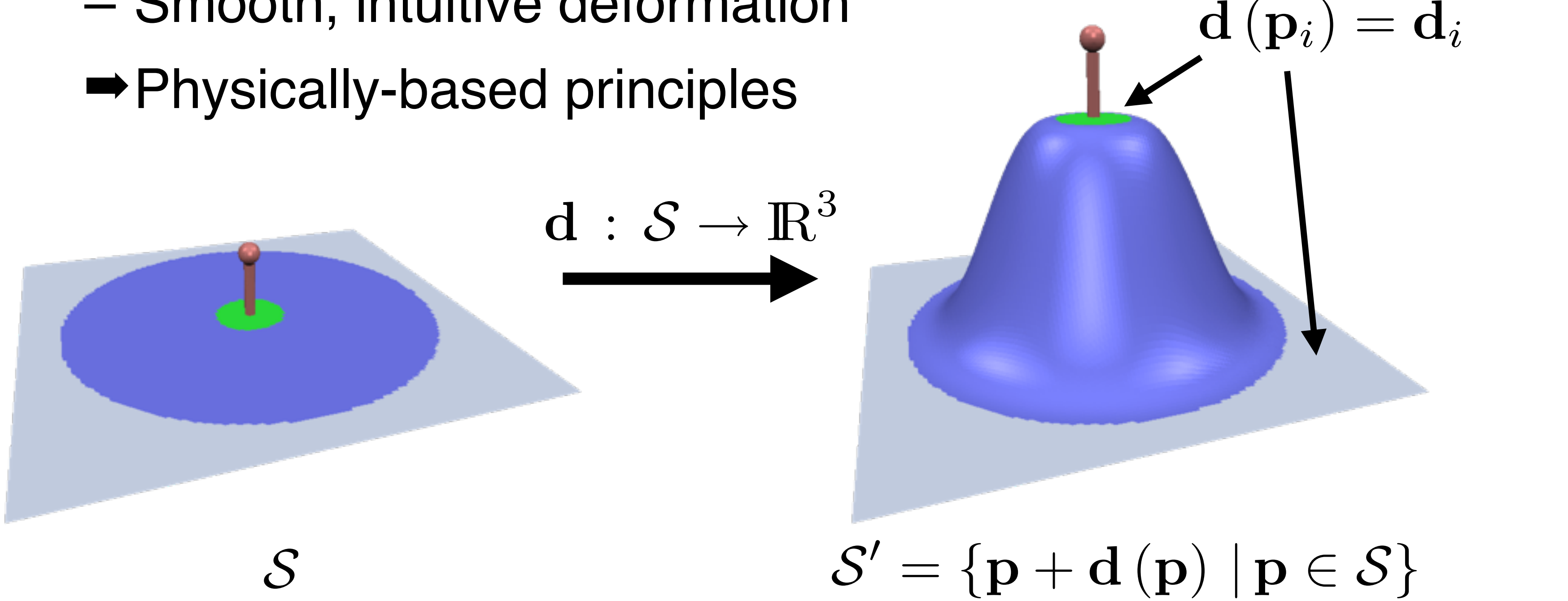
- Surface Deformations
  - Linear Method (Minimize stretching and bending)
  - Non-Linear Method (As rigid as possible)
  - Other methods exist but won’t discuss
- Space Deformations
  - Won’t Discuss
- Linear Parametric Models



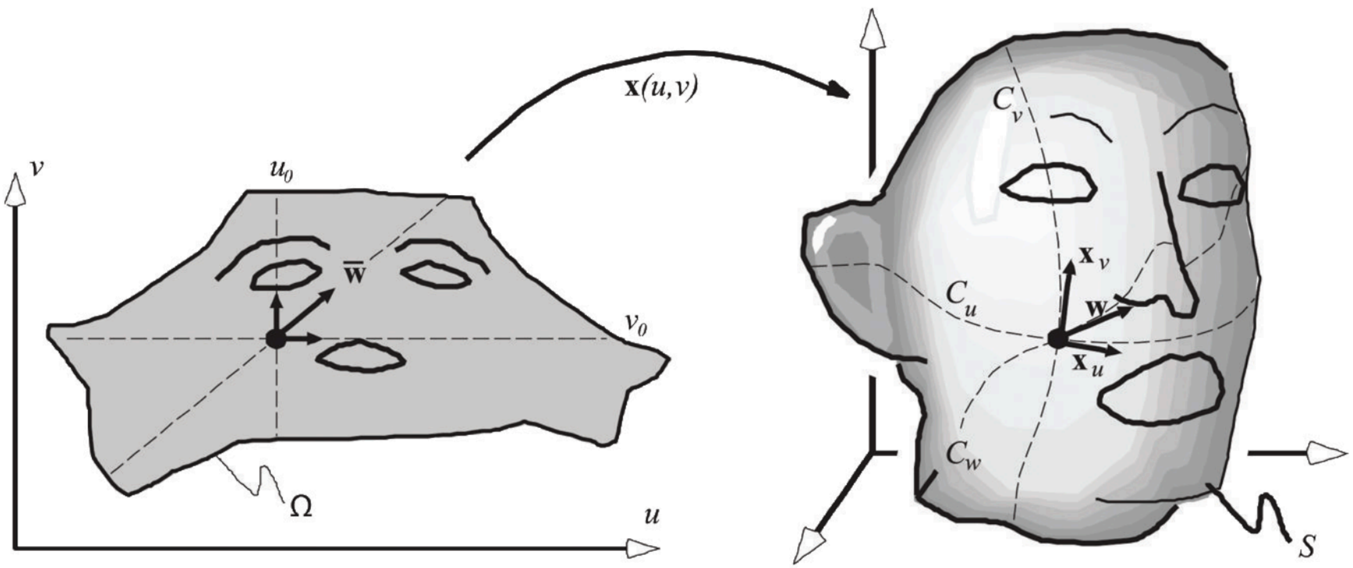
[USC CSCI 621]

Modeling Notation

- Mesh deformation by displacement function **d**
  - – Interpolate prescribed constraints
  - – Smooth, intuitive deformation
  - ➡ Physically-based principles

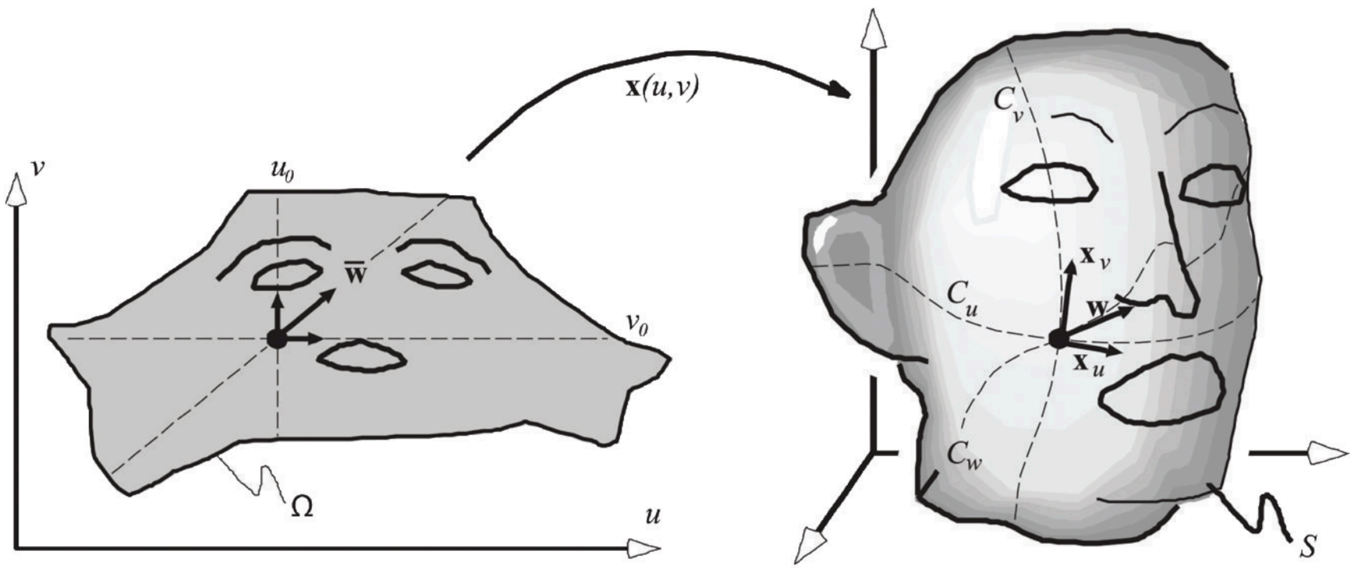


Differential Geometry



$$\mathbf{x}(u,v) = \begin{pmatrix} x(u,v) \\ y(u,v) \\ z(u,v) \end{pmatrix}, \quad (u,v) \in \Omega \subset \mathbb{R}^2,$$

Differential Geometry

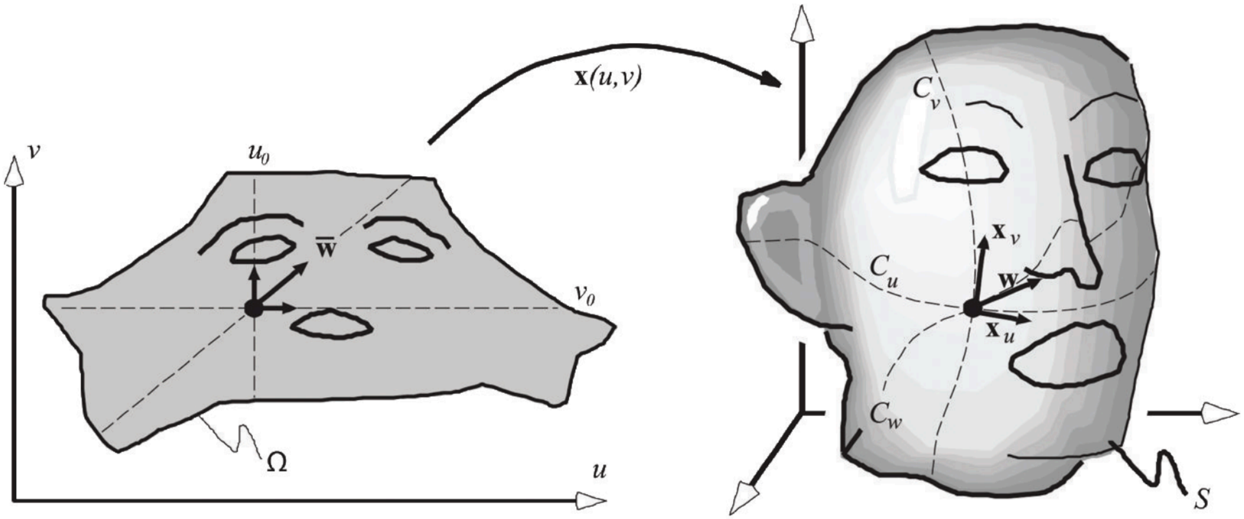


$$\mathbf{x}_u(u_0, v_0) := \frac{\partial \mathbf{x}}{\partial u}(u_0, v_0) \quad \text{and} \quad \mathbf{x}_v(u_0, v_0) := \frac{\partial \mathbf{x}}{\partial v}(u_0, v_0)$$

are, respectively, the tangent vectors of the two *iso-parameter curves*

$$\mathbf{C_u}(t) = \mathbf{x}(u_0 + t, v_0) \quad \text{and} \quad \mathbf{C_v}(t) = \mathbf{x}(u_0, v_0 + t)$$

Differential Geometry



$$\bar{\mathbf{w}} = (u_w, v_w)^T \qquad \mathbf{C}_{\mathbf{w}}(t) = \mathbf{x}(u_0 + tu_w, v_0 + tv_w)$$

$$\mathbf{w} = \partial \mathbf{C}_{\mathbf{w}}(t)/\partial t$$

$$\mathbf{w} = \mathbf{J}\bar{\mathbf{w}} \qquad \mathbf{J} = \begin{bmatrix} \frac{\partial x}{\partial u} & \frac{\partial x}{\partial v} \\ \frac{\partial y}{\partial u} & \frac{\partial y}{\partial v} \\ \frac{\partial z}{\partial u} & \frac{\partial z}{\partial v} \end{bmatrix} = [\mathbf{x}_u \, , \, \mathbf{x}_v]$$

First Fundamental Form

- The first fundamental form acts as a dot product into parameter space
- Gives us a notion of length and angle

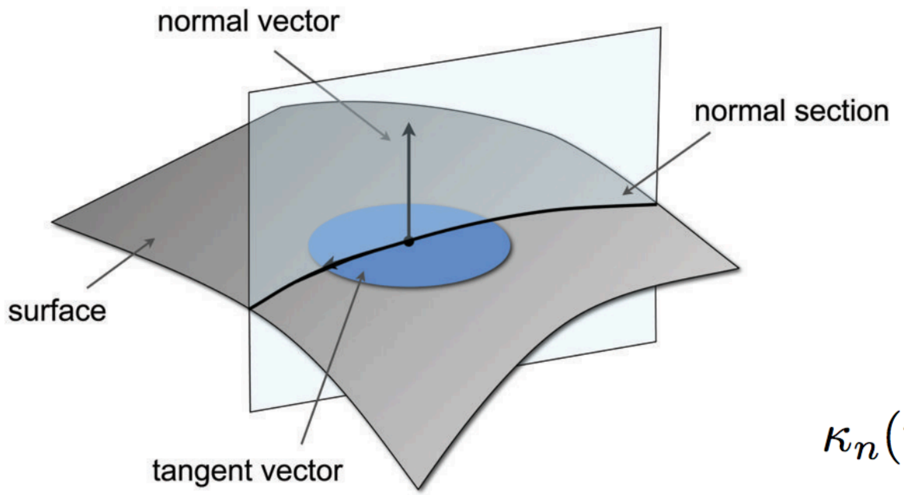
$$\mathbf{w}_1^T \mathbf{w}_2 = (\mathbf{J}\bar{\mathbf{w}}_1)^T (\mathbf{J}\bar{\mathbf{w}}_2) = \bar{\mathbf{w}}_1^T \left(\mathbf{J}^T\mathbf{J}\right) \bar{\mathbf{w}}_2$$

$$\mathbf{I} = \mathbf{J}^T\mathbf{J} = \begin{bmatrix} E & F \\ F & G \end{bmatrix} := \begin{bmatrix} \mathbf{x}_u^T \mathbf{x}_u & \mathbf{x}_u^T \mathbf{x}_v \\ \mathbf{x}_u^T \mathbf{x}_v & \mathbf{x}_v^T \mathbf{x}_v \end{bmatrix}$$

$$\begin{aligned} l(a,b) &= \int_a^b \sqrt{(u_t, v_t)\mathbf{I}(u_t, v_t)^T \mathrm{d}t} \\ &= \int_a^b \sqrt{E u_t^2 + 2F u_t v_t + G v_t^2 \mathrm{d}t}. \end{aligned}$$

Second Fundamental Form

- Second Fundamental Form gives a notion of curvature



$$\mathbf{n} = \frac{\mathbf{x}_u \times \mathbf{x}_v}{\|\mathbf{x}_u \times \mathbf{x}_v\|}$$

$$\kappa_n(\bar{\mathbf{t}}) = \kappa_1 \cos^2 \psi + \kappa_2 \sin^2 \psi$$

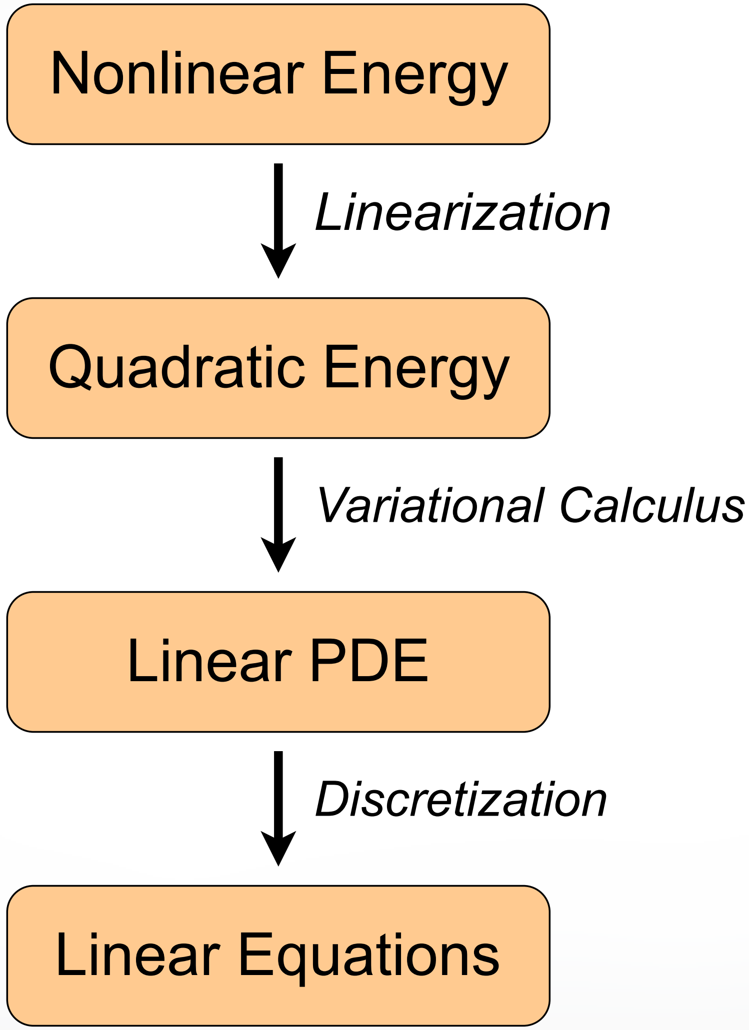
$$\kappa_n(\bar{\mathbf{t}}) = \frac{\bar{\mathbf{t}}^T \mathbf{I\!I}\, \bar{\mathbf{t}}}{\bar{\mathbf{t}}^T \mathbf{I} \bar{\mathbf{t}}} = \frac{e u_t^2 + 2 f u_t v_t + g v_t^2}{E u_t^2 + 2 F u_t v_t + G v_t^2}$$

where $\mathbf{I\!I}$ denotes the *second fundamental form* defined as

$$\mathbf{I\!I} = \begin{bmatrix} e & f \\ f & g \end{bmatrix} := \begin{bmatrix} \mathbf{x}_{uu}^T \mathbf{n} & \mathbf{x}_{uv}^T \mathbf{n} \\ \mathbf{x}_{uv}^T \mathbf{n} & \mathbf{x}_{vv}^T \mathbf{n} \end{bmatrix}.$$

Derivation Steps

[USC CSCI 621]



25

Physically-Based Deformation

- Non-linear stretching & bending energies

$$\int_{\Omega} k_s \underbrace{\boxed{\left\| \mathbf{I} - \mathbf{I}' \right\|^2}}_{\text{stretching}} + k_b \underbrace{\boxed{\left\| \mathbf{II} - \mathbf{II}' \right\|^2}}_{\text{bending}} \, \mathrm{d}u\mathrm{d}v$$

- Linearize energies

$$\int_{\Omega} k_s \underbrace{\boxed{\left(\|\mathbf{d}_u\|^2 + \|\mathbf{d}_v\|^2 \right)}}_{\text{stretching}} + k_b \underbrace{\boxed{\left(\|\mathbf{d}_{uu}\|^2 + 2\,\|\mathbf{d}_{uv}\|^2 + \|\mathbf{d}_{vv}\|^2 \right)}}_{\text{bending}} \mathrm{d}u\mathrm{d}v$$

Physically-Based Deformation

- Minimize linearized bending energy

$$E(\mathbf{d}) \;=\; \int_{\mathcal{S}} \|\mathbf{d}_{uu}\|^2 + 2\,\|\mathbf{d}_{uv}\|^2 + \|\mathbf{d}_{vv}\|^2 \,\mathrm{d}\mathcal{S} \qquad \boxed{f(x) \rightarrow \min}$$

- Variational calculus, Euler-Lagrange PDE

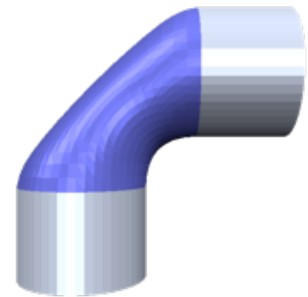
$$\Delta^2 \mathbf{d} \;:=\; \mathbf{d}_{uuuu} + 2\mathbf{d}_{uuvv} + \mathbf{d}_{vvvv} \;=\; 0 \qquad \boxed{f'(x) = 0}$$

➡ “Best” deformation that satisfies constraints

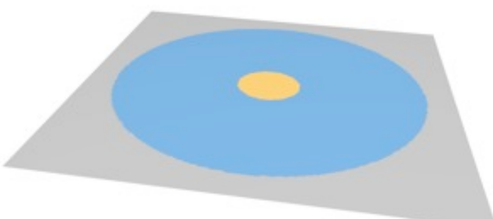
Deformation Energies

$$-k_s \Delta \mathbf{d} + k_b \Delta^2 \mathbf{d} \;=\; \mathbf{0}$$

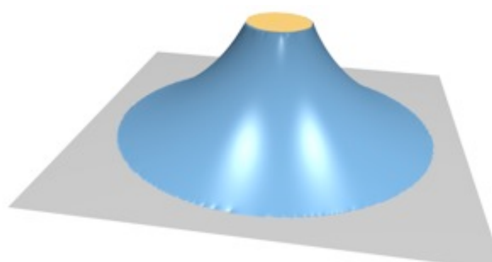
$$\Delta \mathbf{p} = 0$$



$$\Delta^2 \mathbf{p} = 0$$

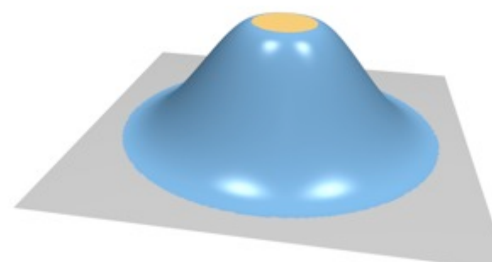


Initial state



$$\Delta \mathbf{d} = 0$$

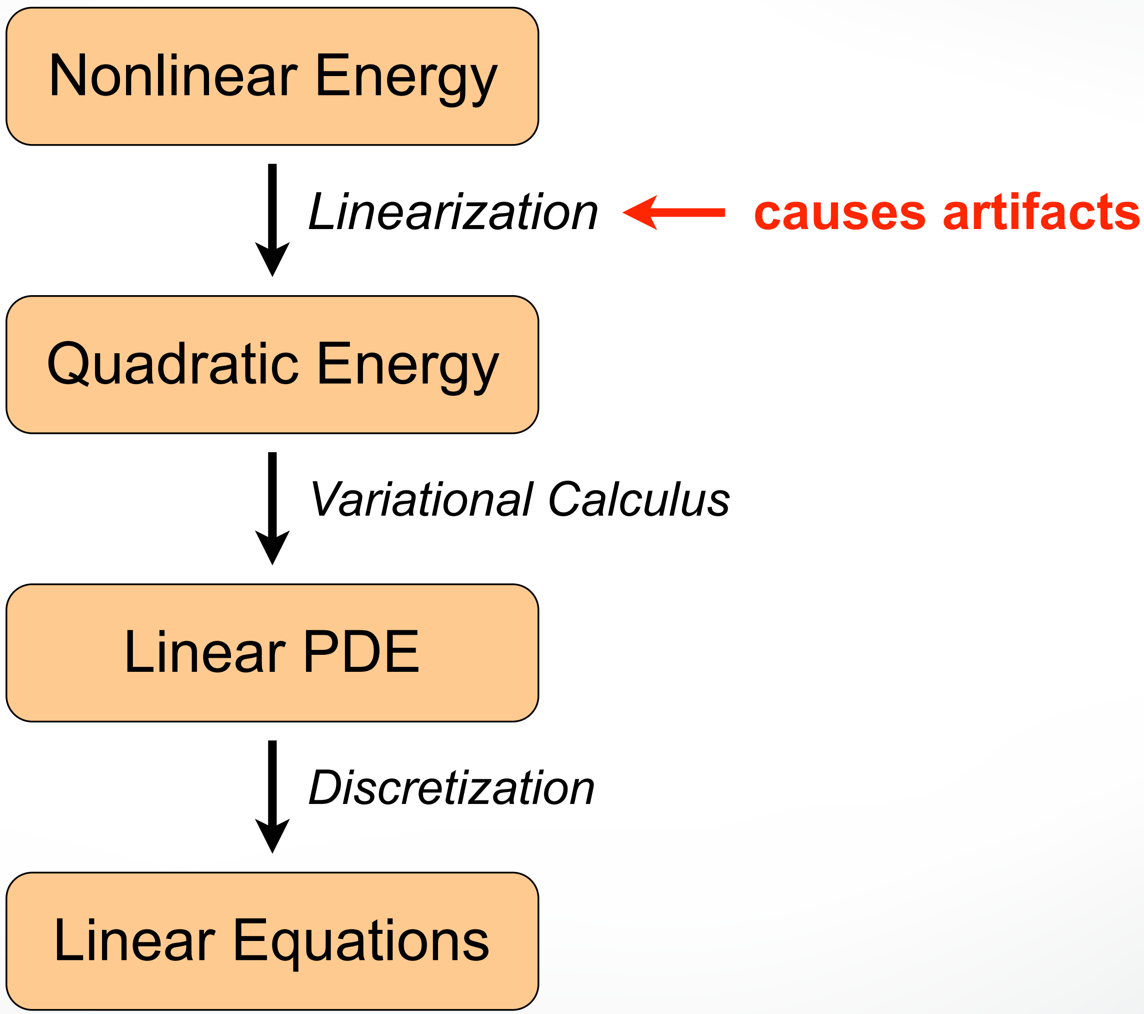
(Membrane)



$$\Delta^2 \mathbf{d} = 0$$

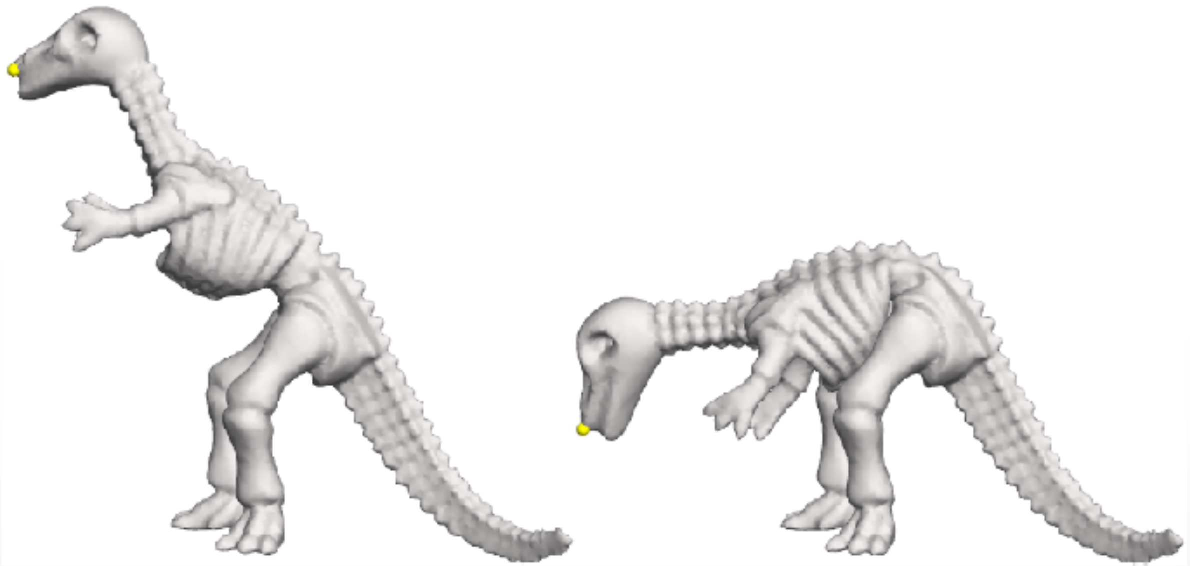
(Thin plate)

Linear Approaches



Surface Deformation

- Smooth large scale deformation
- Local as-rigid-as-possible behavior
  - Preserves small-scale details

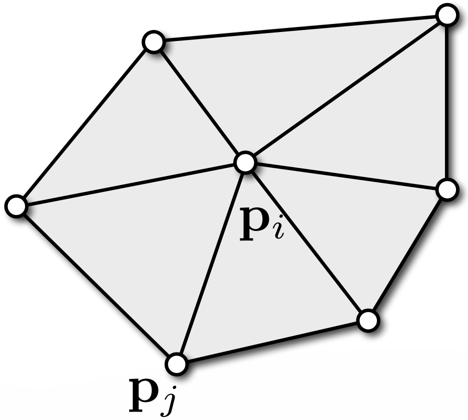


Cell Deformation Energy

[USC CSCI 621]

- Vertex neighborhoods should deform rigidly

$$\sum_{j \in N(i)} \left\| \left(\mathbf{p}'_j - \mathbf{p}'_i\right) - \mathbf{R}_i \left(\mathbf{p}_j - \mathbf{p}_i\right) \right\|^2 \rightarrow \min$$

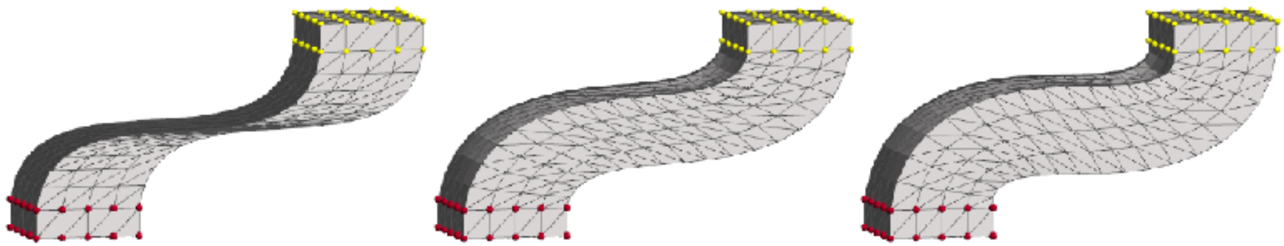


34

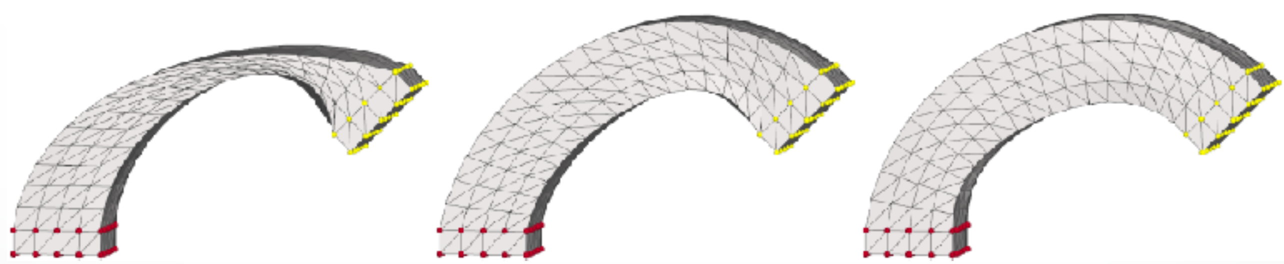
As-Rigid-As-Possible Modeling

[USC CSCI 621]

- Start from naïve Laplacian editing as initial guess

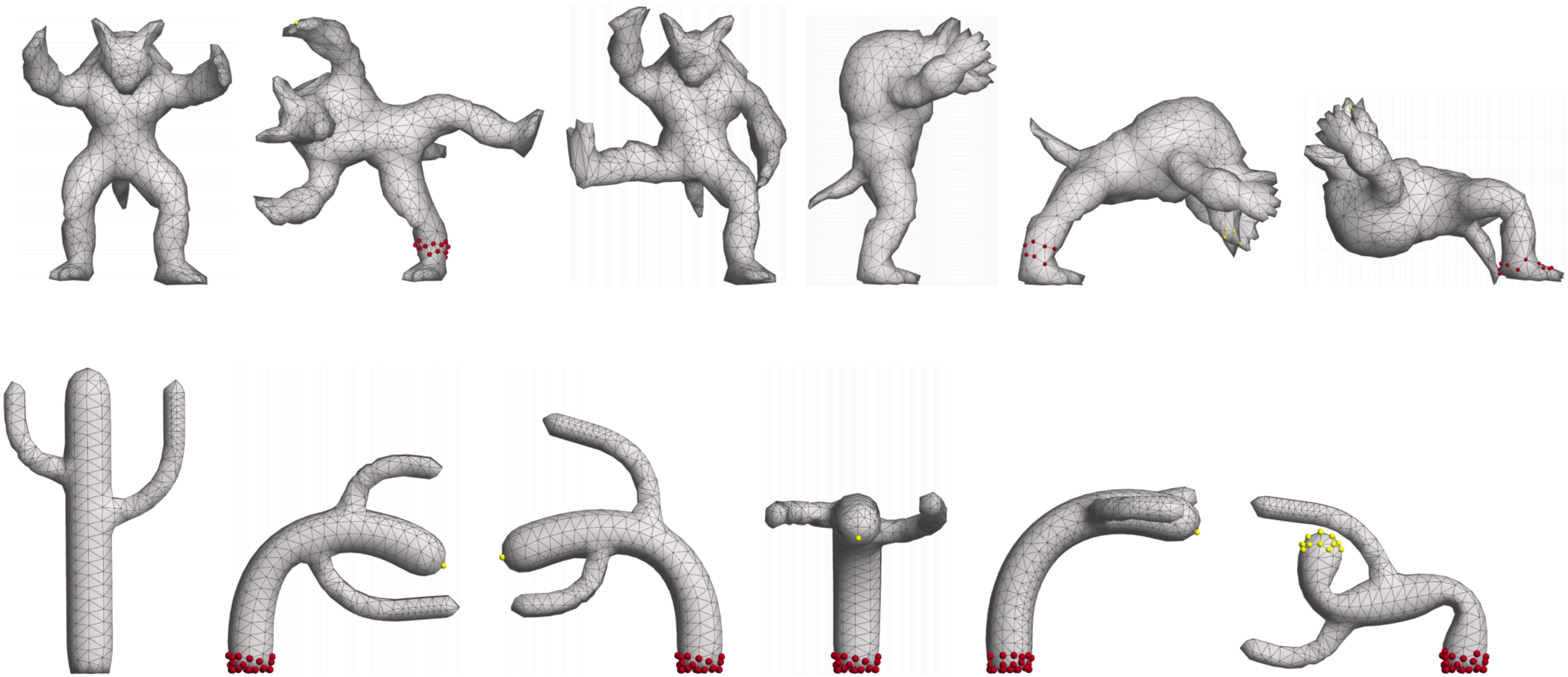


initial guess | 1 iteration | 2 iterations



initial guess | 1 iterations | 4 iterations

As Rigid As Possible Results



[As Rigid As Possible Surface Modeling, SGP 2007]

In Summary

- Exploit Geometry
- Make use of physical structure
- Minimize Stretching and Bending
- Maximize Rigidity

Low-Dimensional Parametric Models

- What if we already know what object we are dealing with?
- For 3D Reconstruction, remove the difficulty in guessing the correct shape from a 2D image
- We can introduce a template shape and a parameterization that deforms the template appropriately
  - Create a shape space
- Instead estimating the whole model we estimate the low-dimensional parameterization
- We can parameterize deformations into fast linear operations on a low-dimensional space

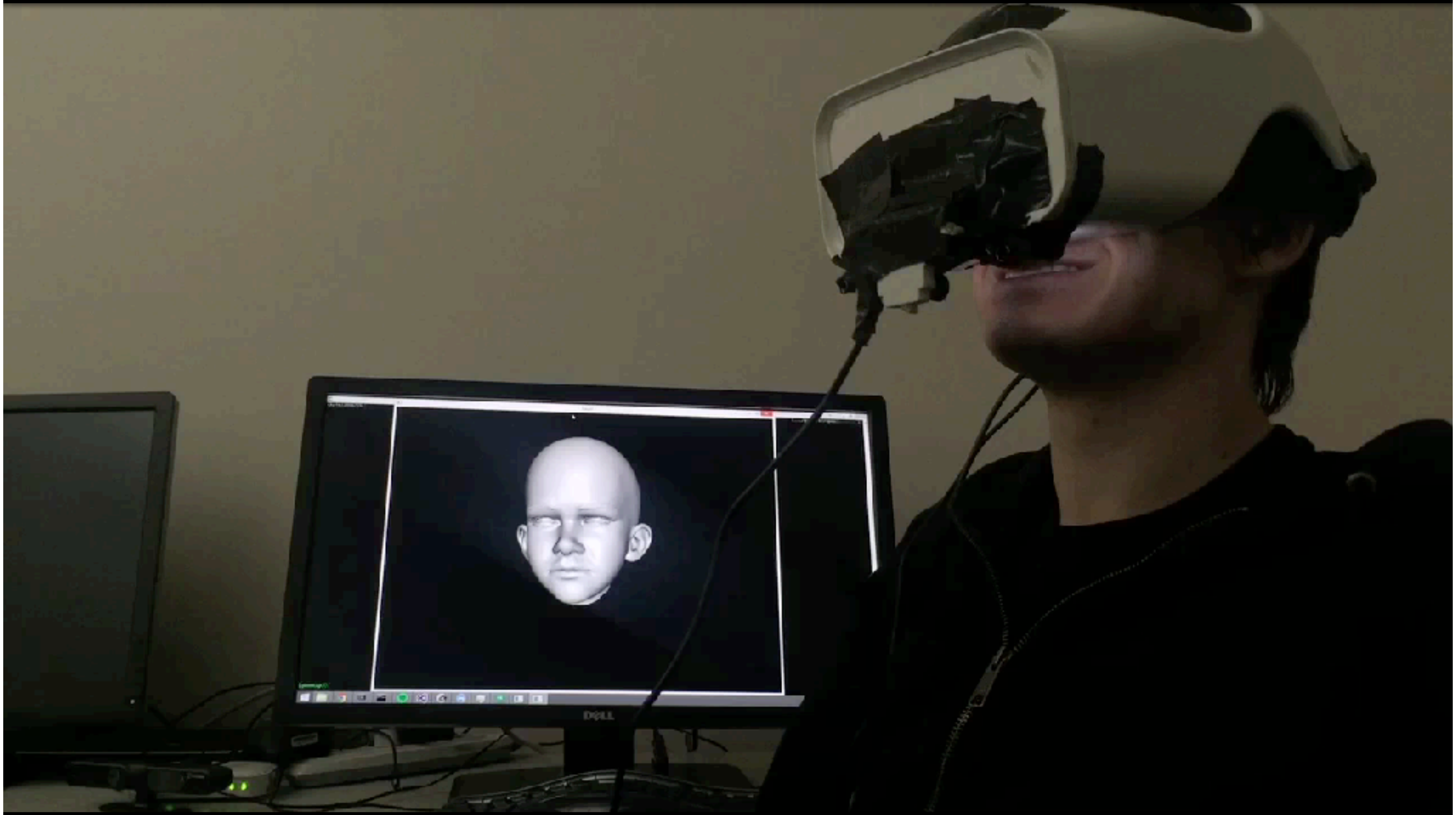
Low-Dimensional Parametric Models

- Pros:
  - Leverage our prior knowledge of the task/our world
  - Low-Dimensional
  - Accurate
  - Won't fall off the shape manifold. Always generate plausible shapes
  - Easy to deform

Some Parametric Models In Industry

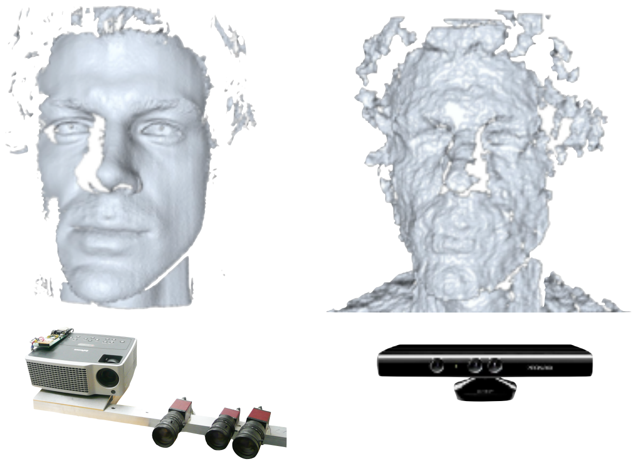
- Some common parametric models include:
  - **Faces**
  - Bodies
  - Hands

Performance to Facial Animation [USC CSCI 621]



Requirements for a Practical System

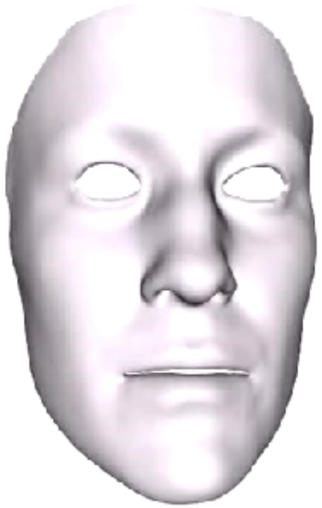
[USC CSCI 621]



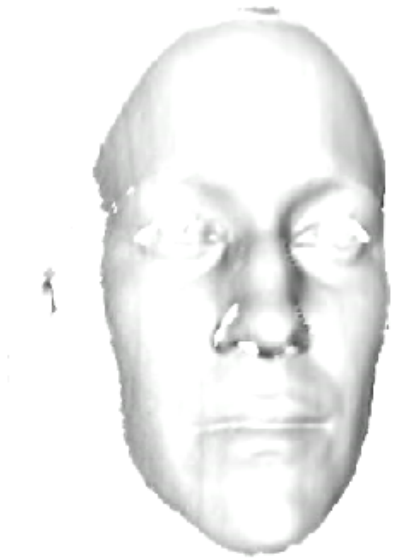
1. Real-time performance
2. Robustness to noise
3. High-level semantics

Building Expression Space

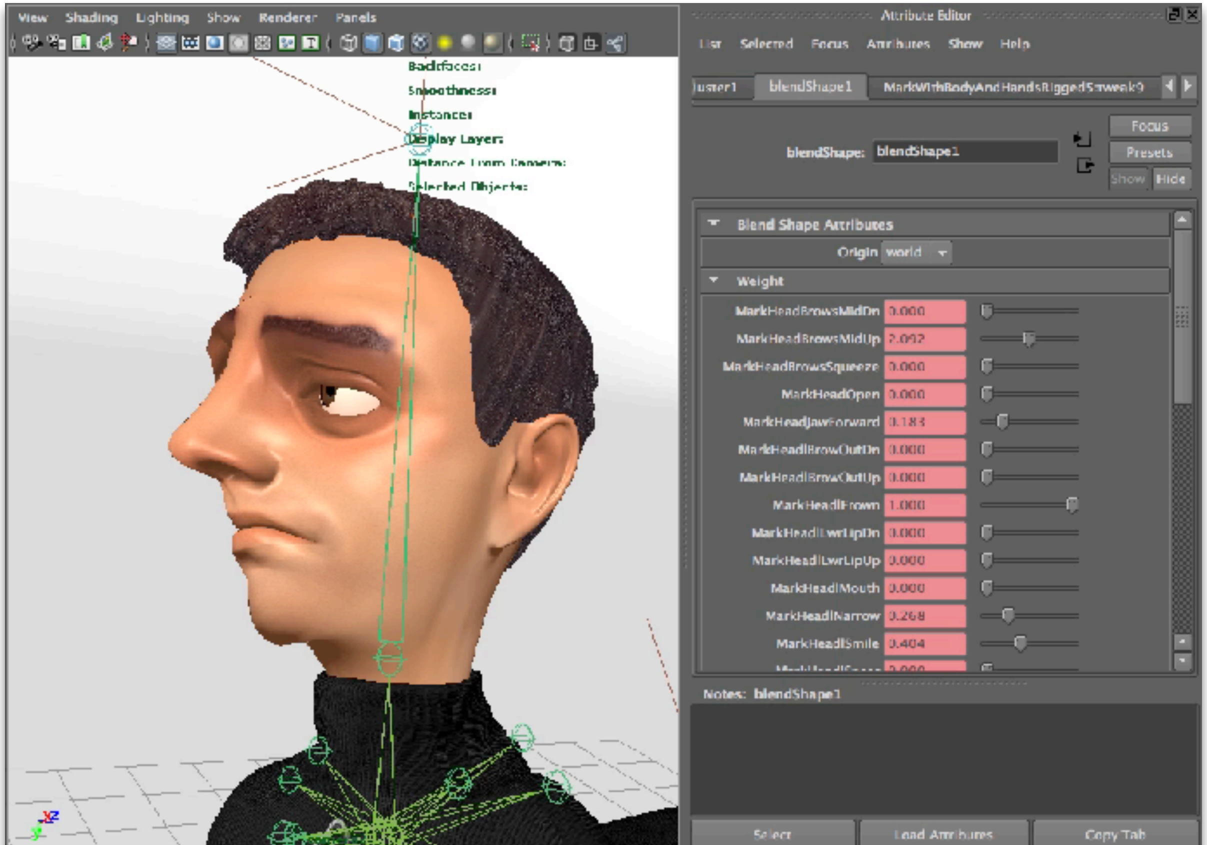
[USC CSCI 621]



tracked template



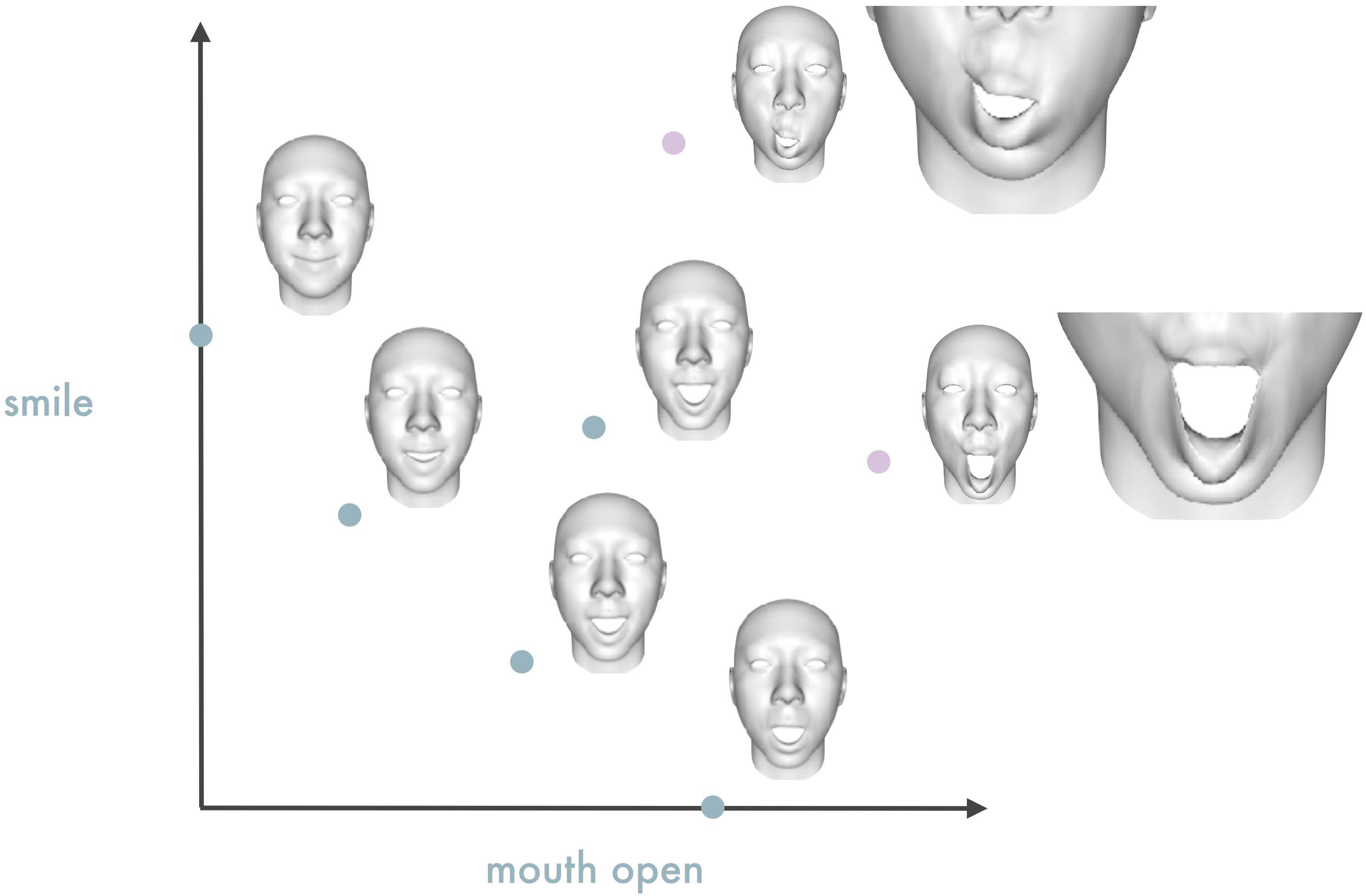
input scan


View Shading Lighting Show Renderer Panels
Backfaces:
Smoothness:
Instances:
Display Layer:
Selected Objects:
Attribute Editor
List Selected Focus Attributes Show Help
blendShape1
Focus
Presets
Show Hide
blendShape: blendShape1
Blend Shape Attributes
Origin world
Weight
MarkHeadBrowsMidDn
MarkHeadBrowsMidUp
MarkHeadBrowsSqueeze
MarkHeadOpen
MarkHeadJawForward
MarkHeadlBrowOutDn
MarkHeadlBrowOutUp
MarkHeadlFrown
MarkHeadlLwrLipDn
MarkHeadlLwrLipUp
MarkHeadlMouth
MarkHeadlNarrow
MarkHeadlSmile
Notes: blendShape1
Select Load Attributes Copy Tab

13

N-Dim Expression Space

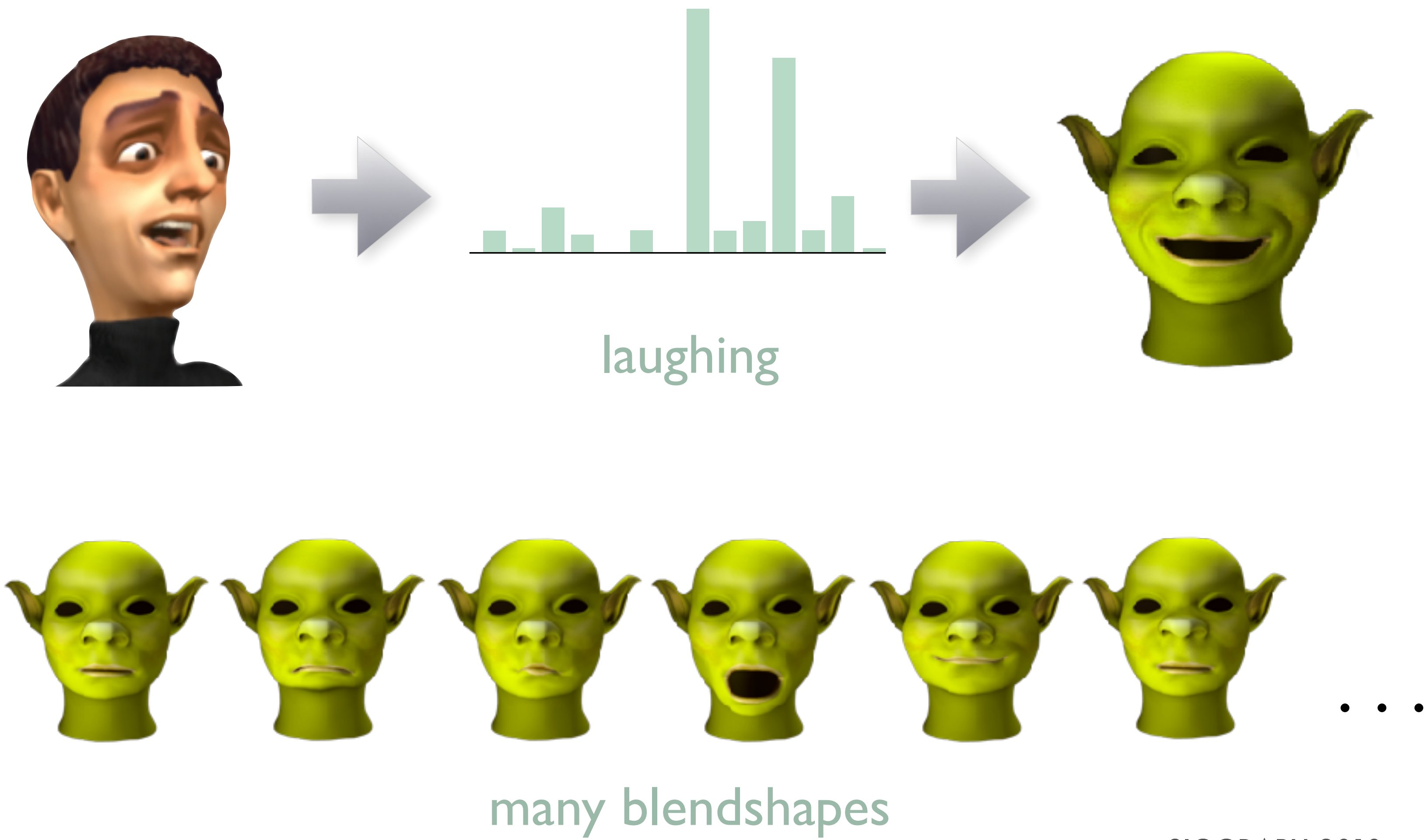
[USC CSCI 621]



58

Blendshape Retargeting

[USC CSCI 621]

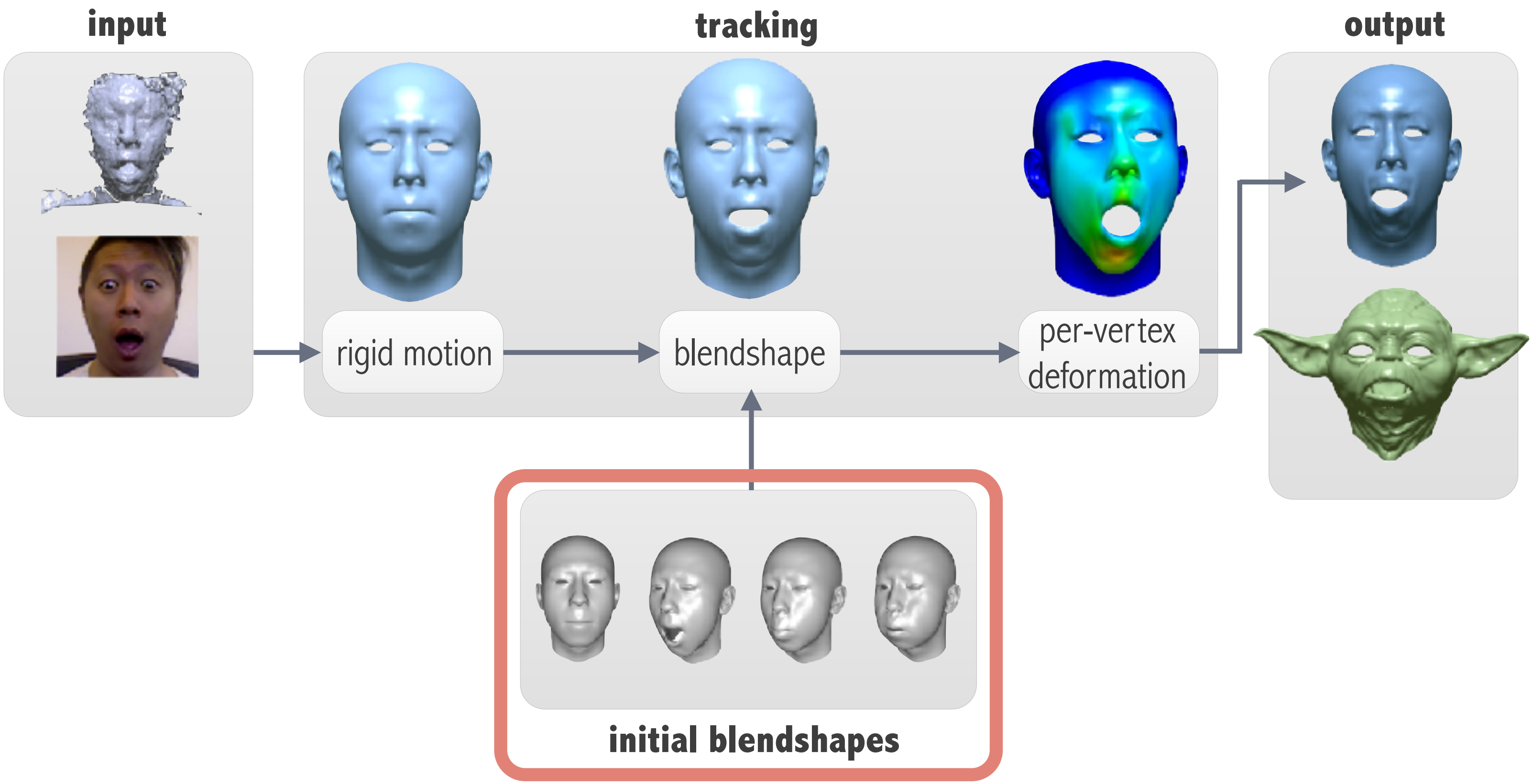


SIGGRAPH 2010

43

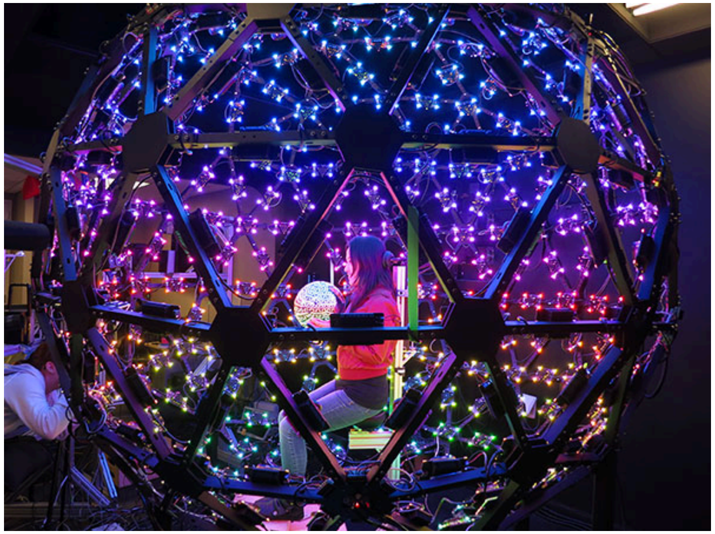
Pipeline Overview

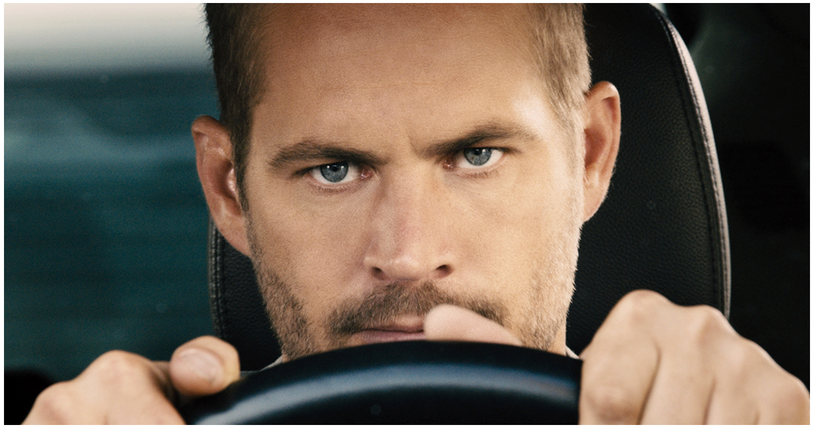
[USC CSCI 621]



76

Hollywood Face Models

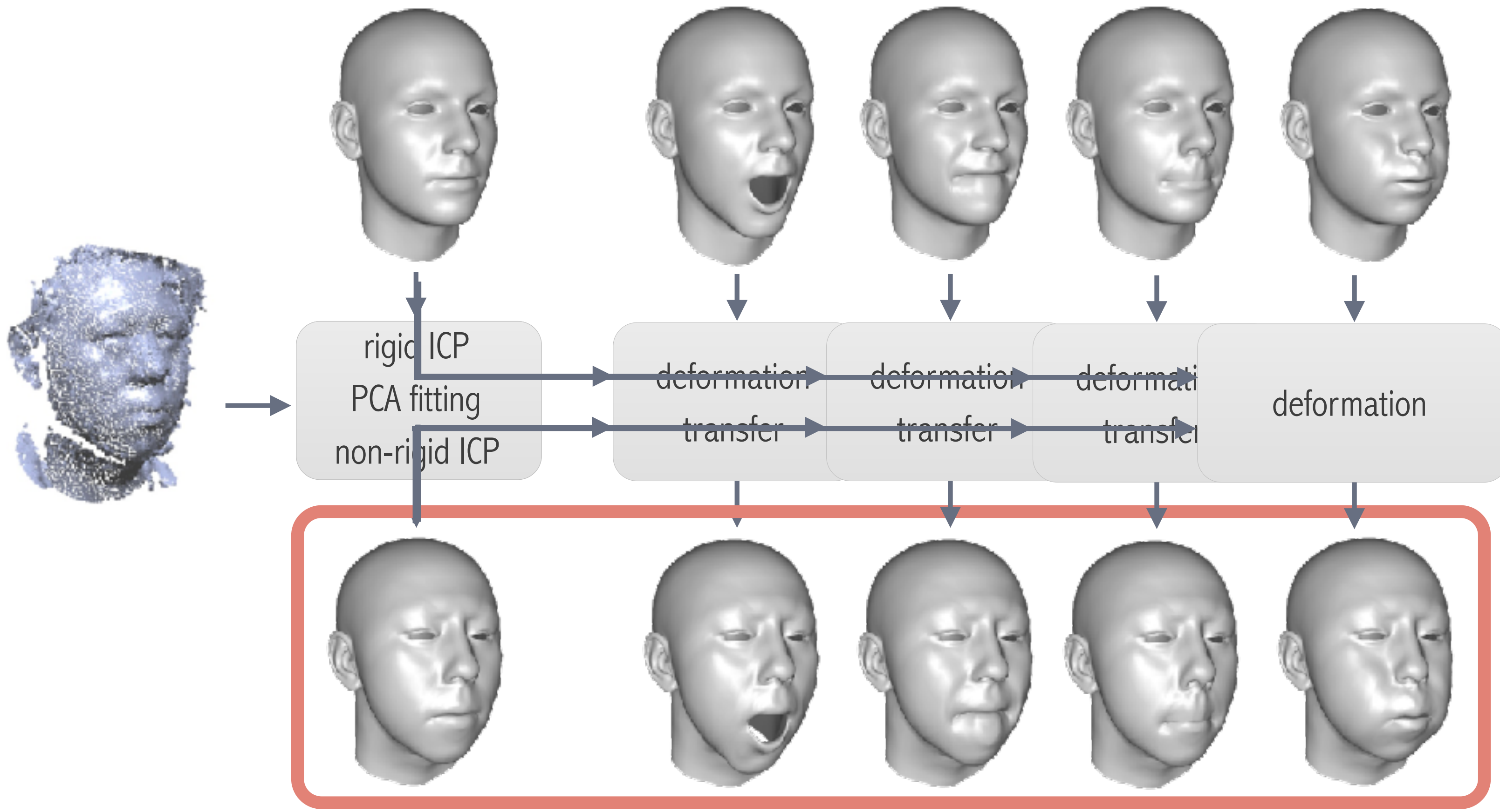




]

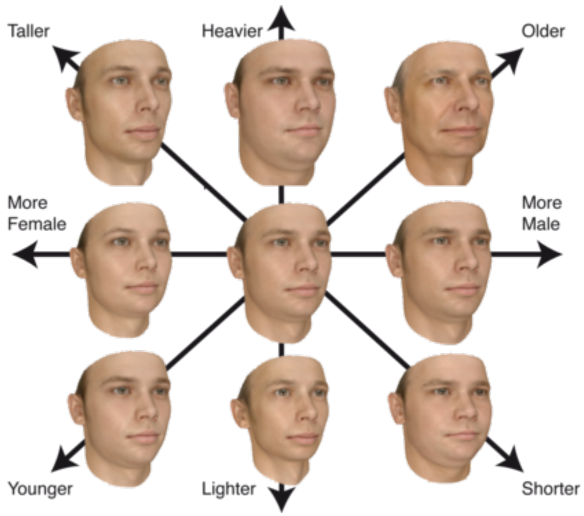
Building Initial Blendshape Model

[USC CSCI 621]

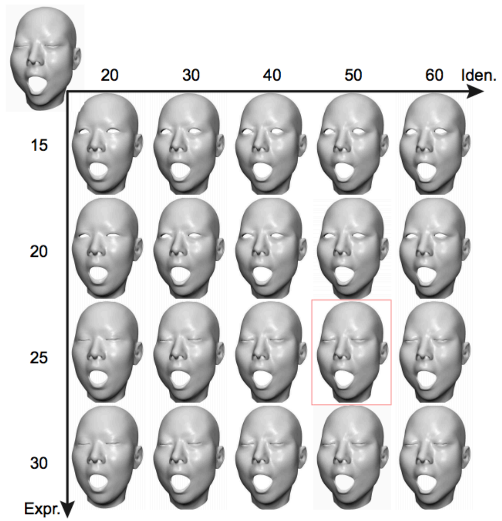


77

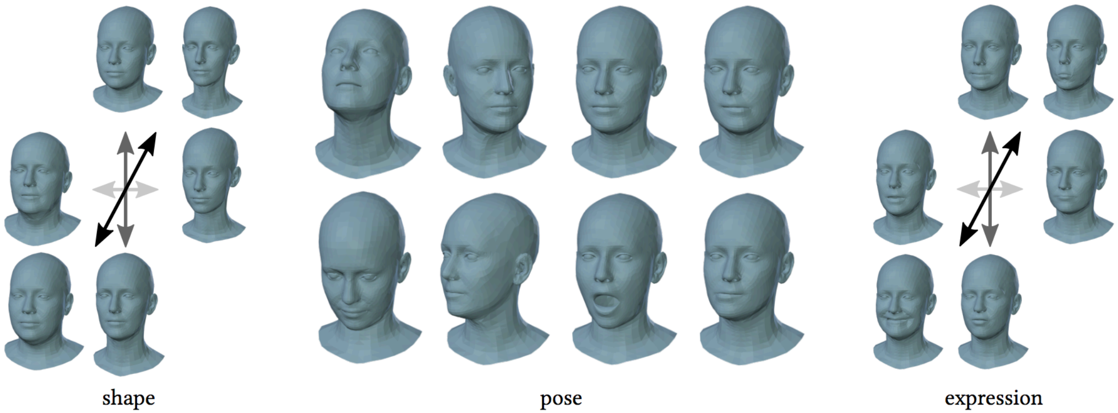
Public Face Models



[Basel Face Model, 1997, 2009, 2017]



[FaceWarehouse, TOG 2014]



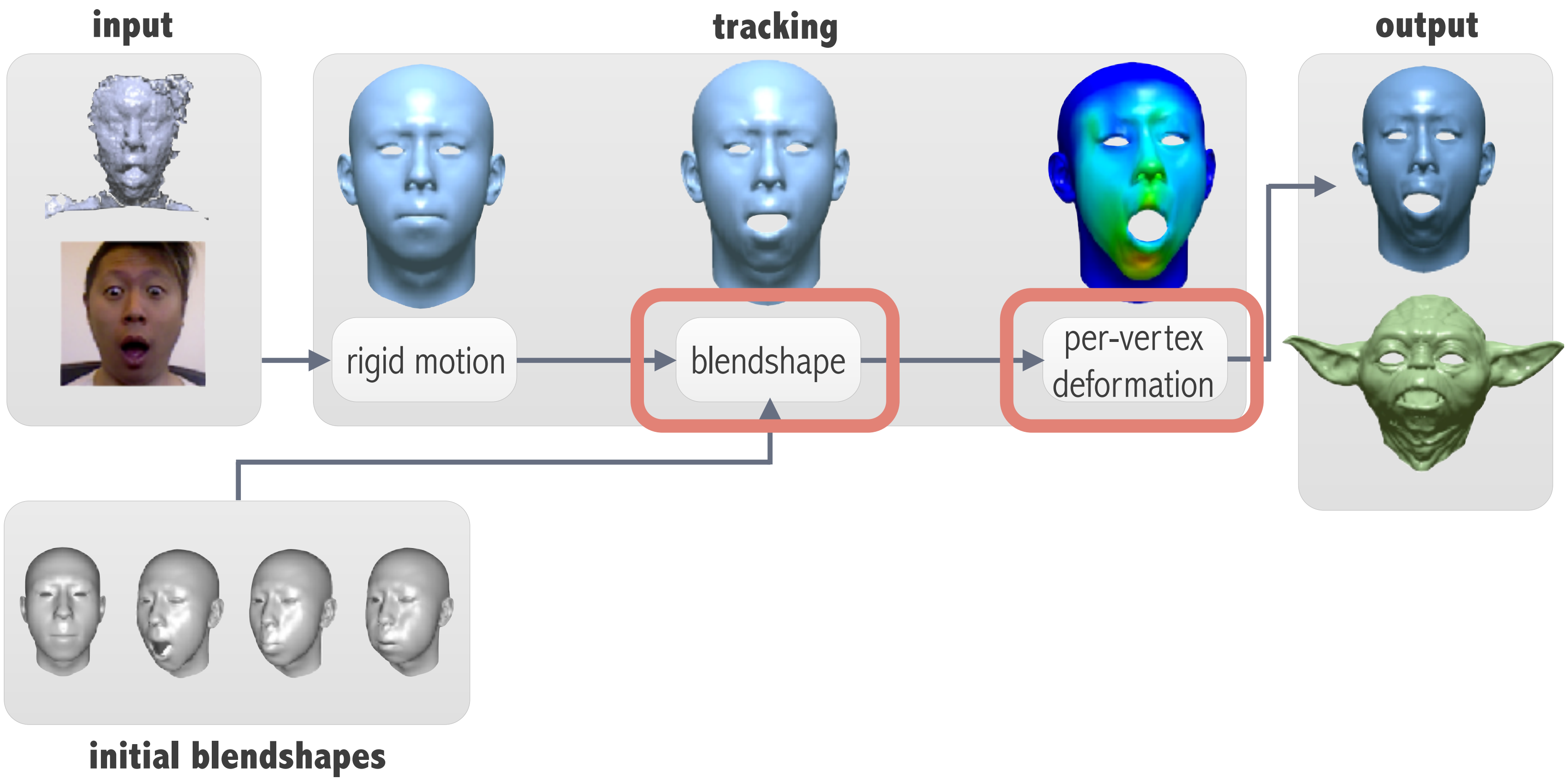
[FLAME, SIGGRAPH Asia 2017]

Building Initial Blendshape Models

- Exploit commonalities among face data
- Create a low-dimensional shape space that you can interpolate inside of
- We have used data to largely automate the process
- Difficult future work:
  - Are there even more basic commonalities among nature that we can exploit?
  - Can we learn to exploit these basic commonalities automatically?

Pipeline Overview

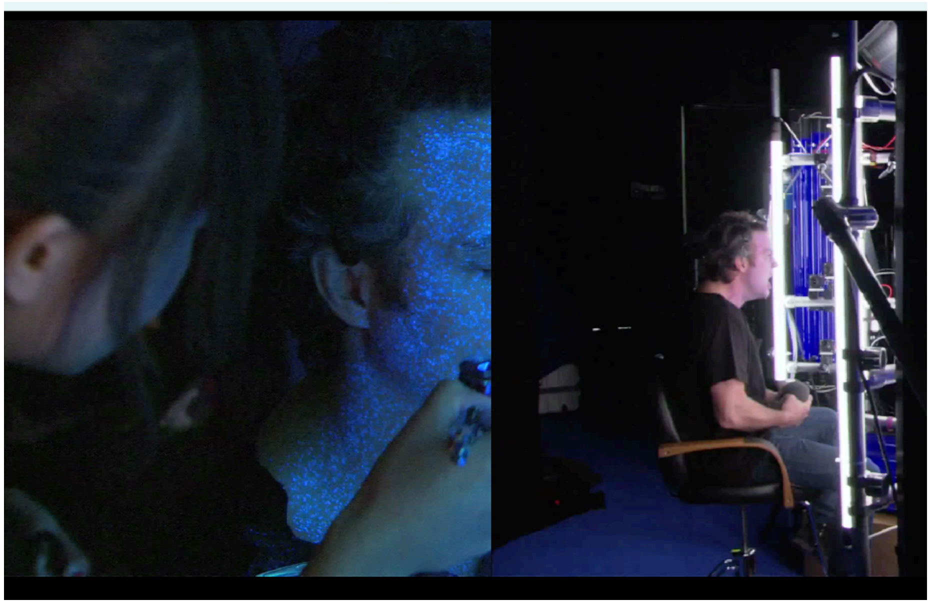
[USC CSCI 621]

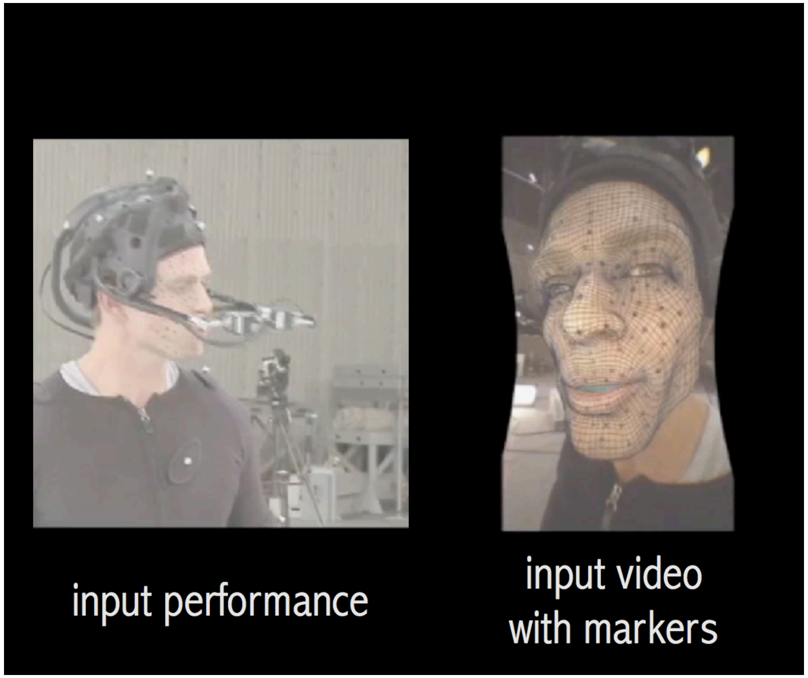


82

Solving for Pose and Blend-shapes

- First extract facial landmarks





Solving for Pose and Blend-shapes

- First extract facial landmarks
- Markerless capture
- Democratizing human body digitization!

[Face++]

Solving for Pose and Blend-shapes

[Face++]

Given a set of (p^i, q^i) correspondences where p^i denotes a pixel in the input image and q^i denotes its corresponding vertex in the 3DMM, we minimize the following energy function defined over the parameters $\mathcal{X} = (f, \mathbf{R}, \mathbf{t}, \alpha_{id}, \alpha_{exp})$:

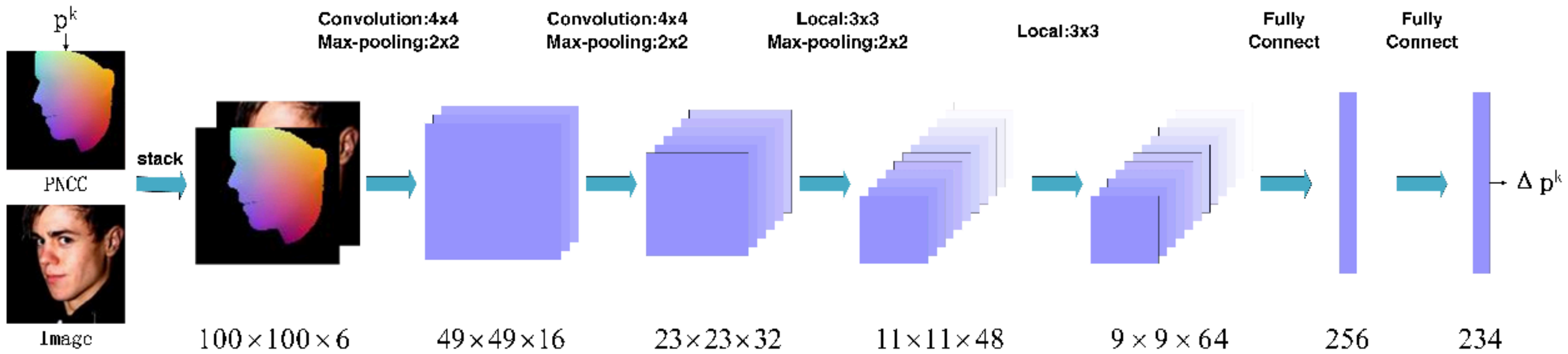
$$
\begin{aligned}
E(\mathcal{X}) &= E_{data}(\mathcal{X}) + E_{reg}(\mathcal{X}), \qquad (3)\\
E_{data}(\mathcal{X}) &= \sum_i w_i \|p^i - \Pi_f(\mathbf{R}S_{q^i} + \mathbf{t})\|^2,\\
E_{reg}(\mathcal{X}) &= w_{id} \sum_{id,i} \left(\frac{\alpha_{id,i}}{\sigma_{id,i}}\right)^2 + w_{exp} \sum_{exp,i} \left(\frac{\alpha_{exp,i}}{\sigma_{exp,i}}\right)^2,
\end{aligned}
$$

$$S_{q^i} = (\bar{S} + A_{id}\alpha_{id} + A_{exp}\alpha_{exp})_{q^i}$$

[Yu et al., ICCV 2017]

Solving for Pose and Blend-shapes…

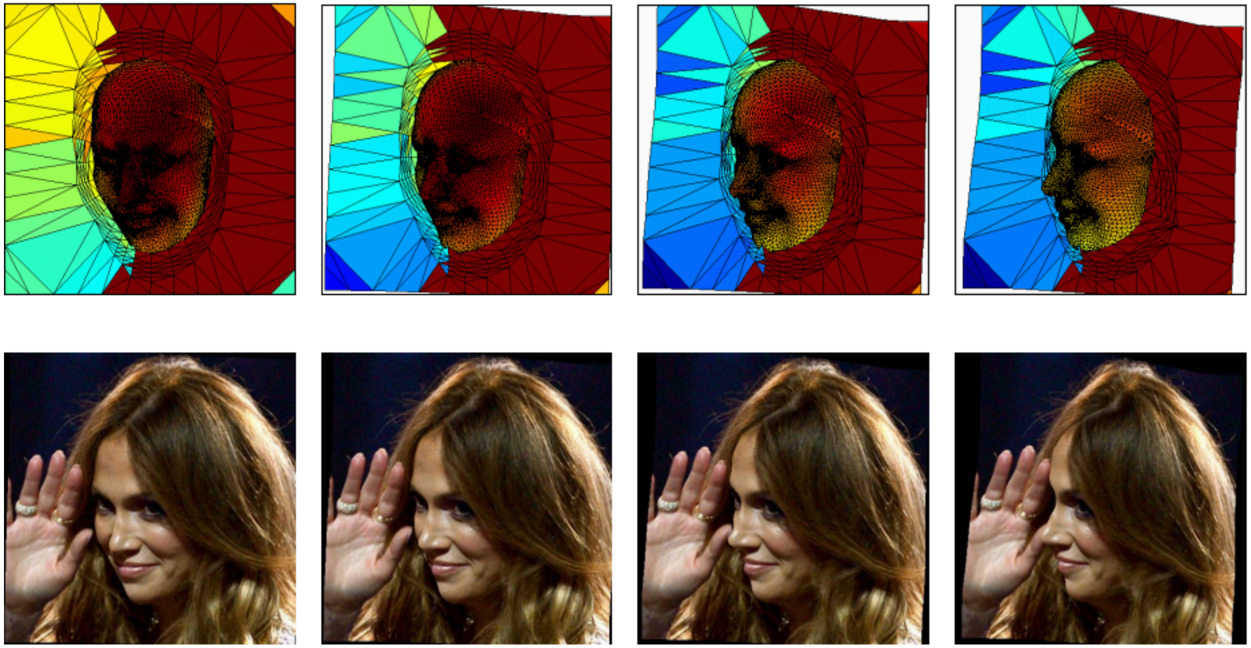
- …with deep learning!
- 3DFFA directly learns pose and shape parameters



[3DDFA, CVPR 2016]

Solving for Pose and Blend-shapes…

- …with deep learning!
- 3DFFA directly learns pose and shape parameters
- Allows face-tracking for side-faces and difficult occlusions because deep learning



[3DDFA, CVPR 2016]

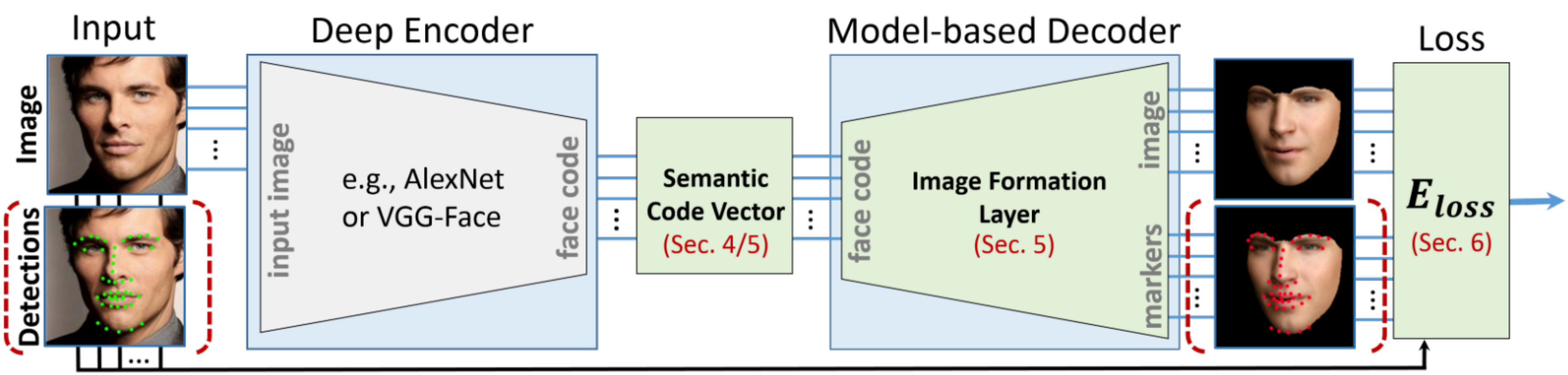
Solving for Pose and Blend-shapes…

- …with deep learning!
- 3DFFA directly learns pose and shape parameters
- Allows face-tracking for side-faces and difficult occlusions because deep learning

[3DDFA, CVPR 2016]

More face models and deep learning

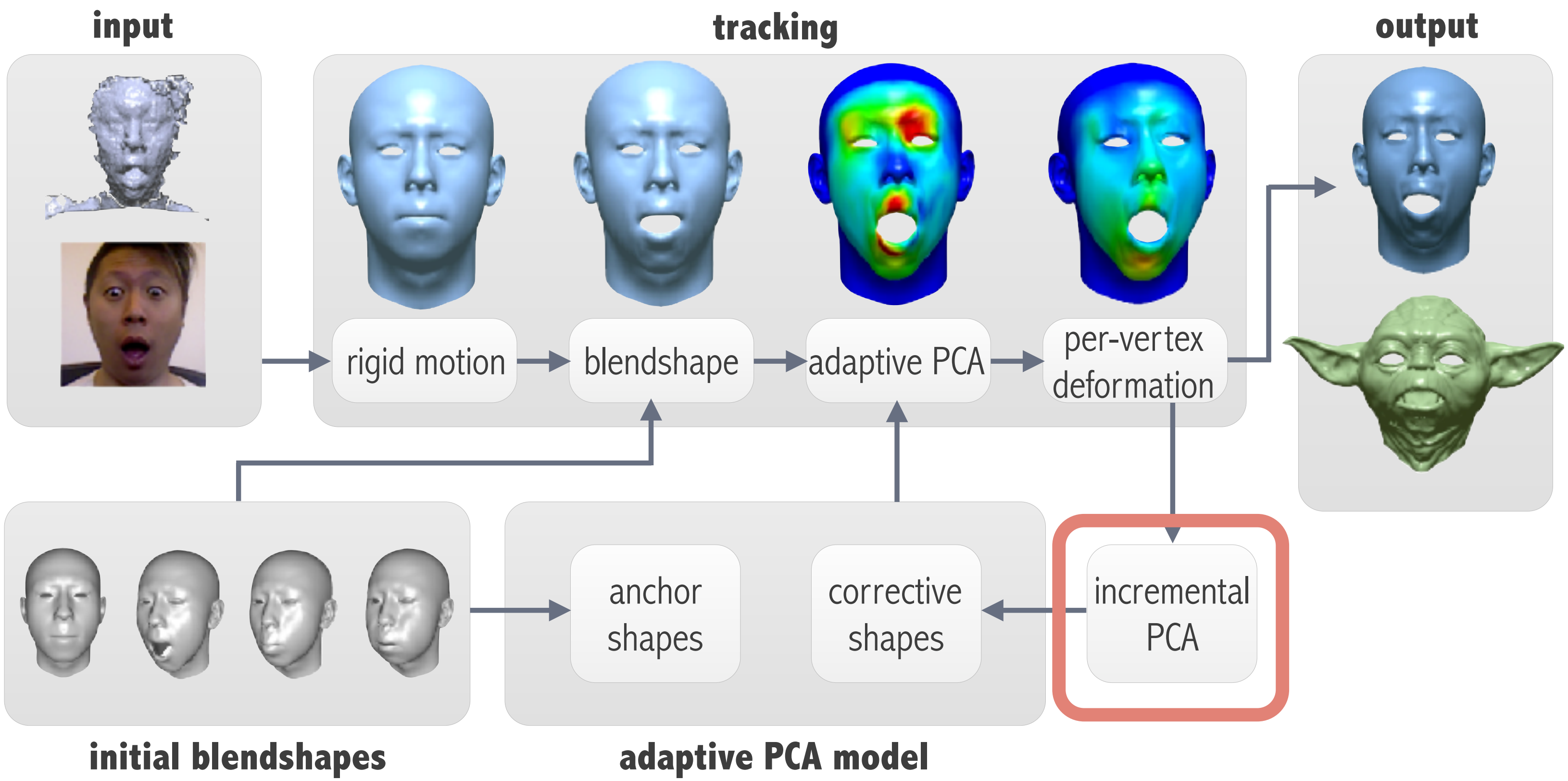
- Unsupervised deep learning also possible using parametric models
- Passes knowledge of the model into the encoder



[MOFA, ICCV 2017]

Pipeline Overview

[USC CSCI 621]



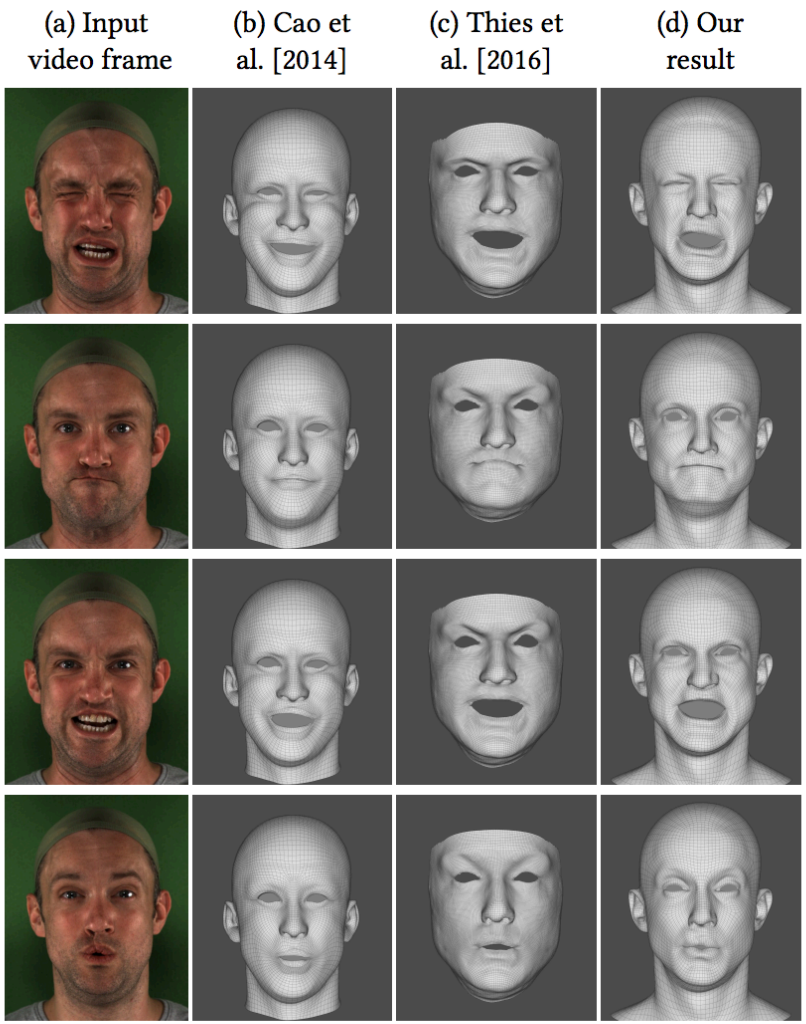
89

Doing it all…

- …with deep learning!

Name	Description
input	Input $1 \times 240 \times 320$ image
conv1a	Conv 3×3, $1 \longrightarrow 64$, stride 2×2, ReLU
conv1b	Conv 3×3, $64 \longrightarrow 64$, stride 1×1, ReLU
conv2a	Conv 3×3, $64 \longrightarrow 96$, stride 2×2, ReLU
conv2b	Conv 3×3, $96 \longrightarrow 96$, stride 1×1, ReLU
conv3a	Conv 3×3, $96 \longrightarrow 144$, stride 2×2, ReLU
conv3b	Conv 3×3, $144 \longrightarrow 144$, stride 1×1, ReLU
conv4a	Conv 3×3, $144 \longrightarrow 216$, stride 2×2, ReLU
conv4b	Conv 3×3, $216 \longrightarrow 216$, stride 1×1, ReLU
conv5a	Conv 3×3, $216 \longrightarrow 324$, stride 2×2, ReLU
conv5b	Conv 3×3, $324 \longrightarrow 324$, stride 1×1, ReLU
conv6a	Conv 3×3, $324 \longrightarrow 486$, stride 2×2, ReLU
conv6b	Conv 3×3, $486 \longrightarrow 486$, stride 1×1, ReLU
drop	Dropout, $p = 0.2$
fc	Fully connected $9720 \longrightarrow 160$, linear activation
output	Fully connected $160 \longrightarrow N_{\text{out}}$, linear activation

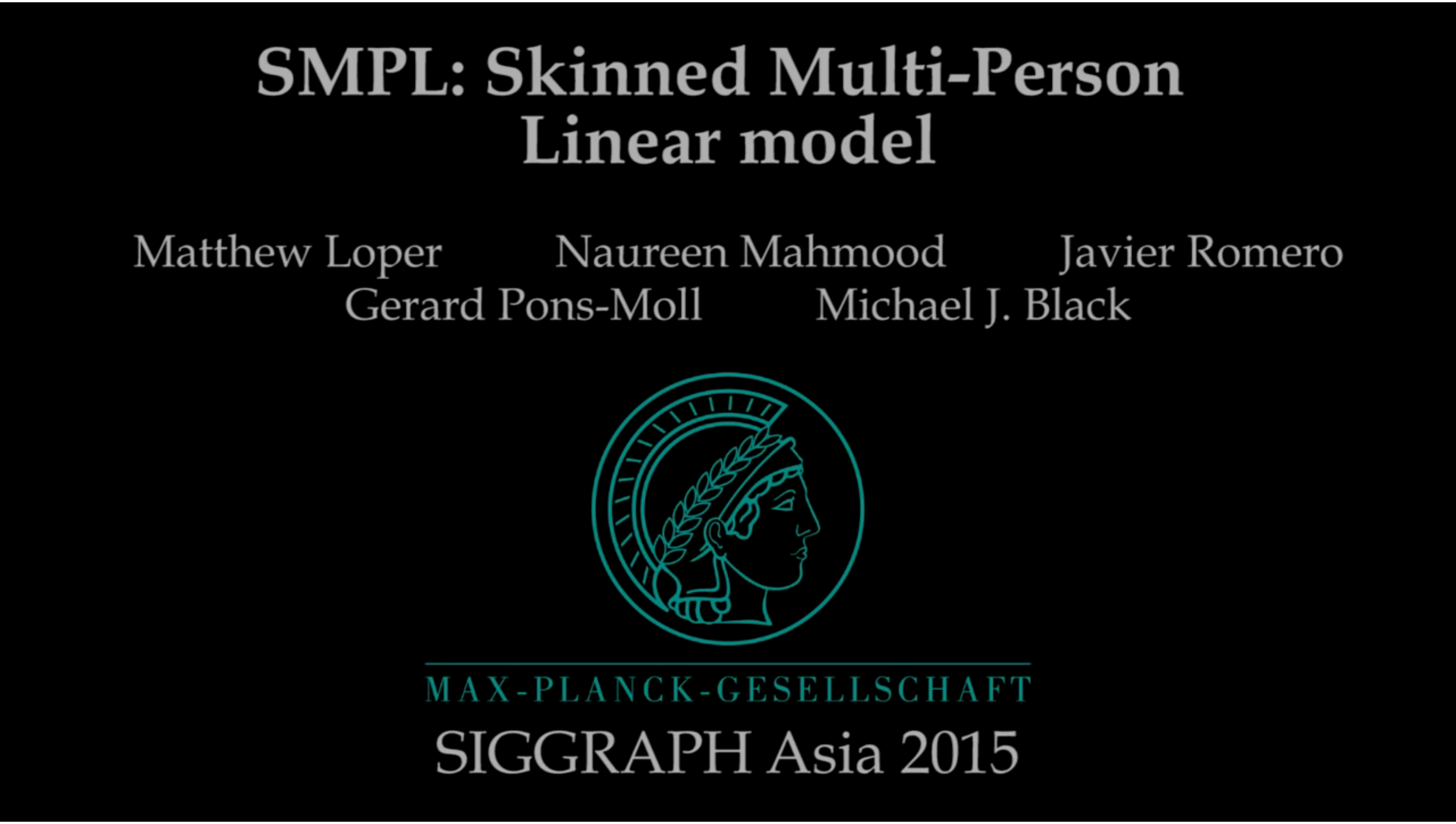
[Laine et al. , SCA 2017]



Some Parametric Models

- Some common parametric models include:
  - Faces
  - **Bodies**
  - Hands

SMPL (SIGGRAPH Asia 2015)

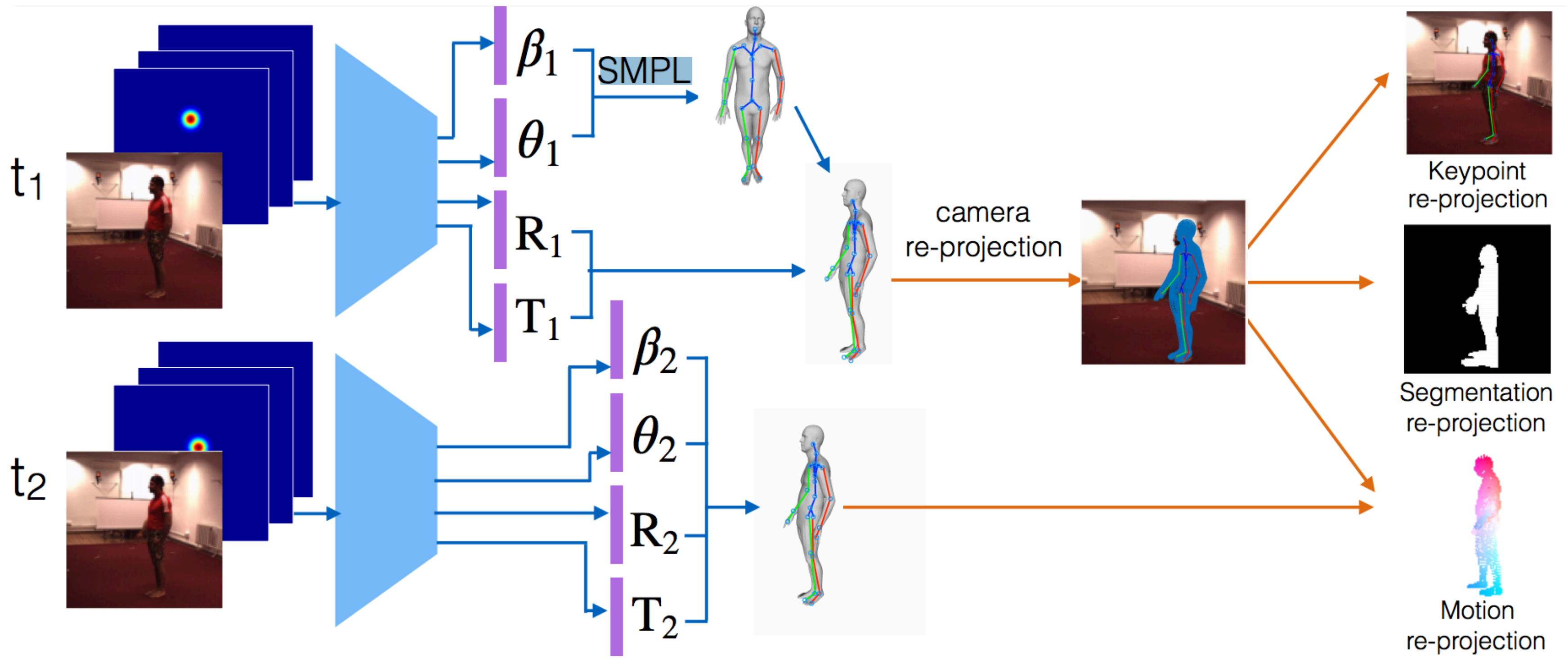


SIMPLify (ECCV 2016)

- Learn to fit a human body model to a single image
- First estimate pose points using CNN (only CNN methods perform well)
- Fit body parameters to pose points

Fitting Parameters…

- …using deep learning

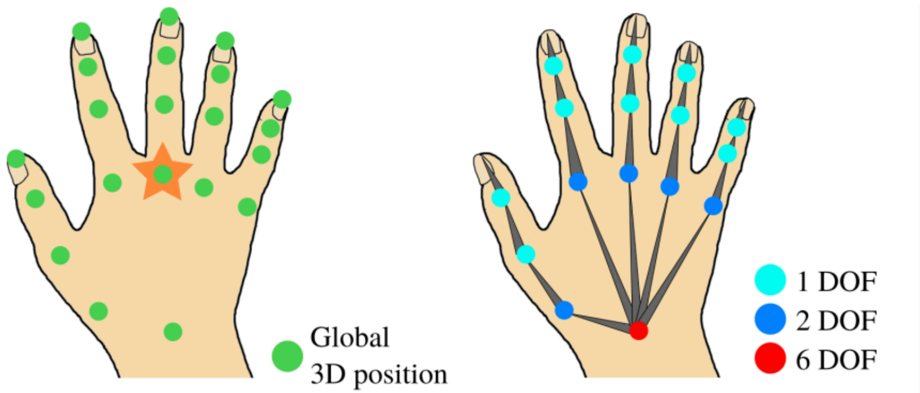


[Tung et al., NIPS 2017]

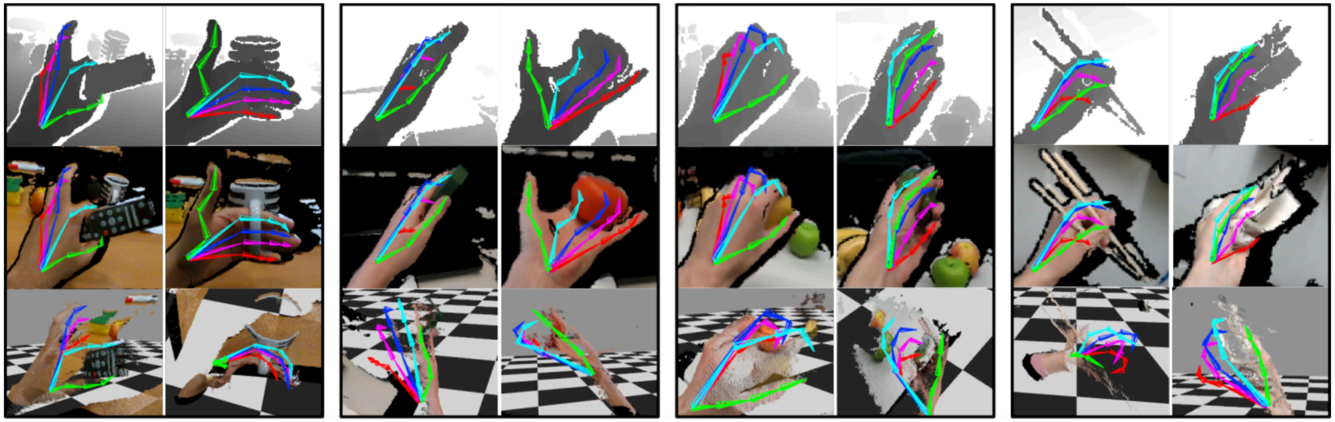
Some Parametric Models

- Some common parametric models include:
  - Faces
  - Bodies
  - **Hands**

Hand Models



$\mathbf{p}^{\mathbf{G}}$ Θ



[Mueller et al., ICCV 2017]

Conclusion

- 3D structures carry a geometric and physical meaning that classical geometers and graphics people have exploited
- Many graphics/geometry problems make use of knowledge and assumptions of our world for better performance
  - Exploit commonalities
  - Low-Dimensional Parametric Models
  - Easy Deformation
  - Easy 3D estimation from a single image
- Deep learning is everywhere
- Can we learn these assumptions?