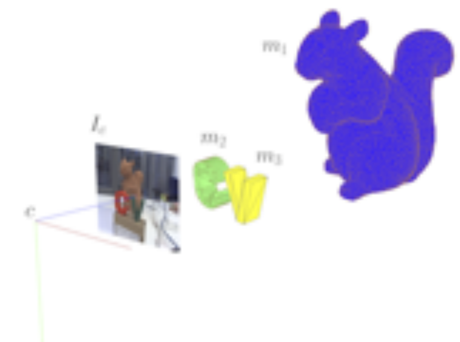
Applications of CNNs in Image Processing and Geometric Reconstruction




Uli Schwanecke

Computer Graphics and Vision
RheinMain University of Applied Sciences

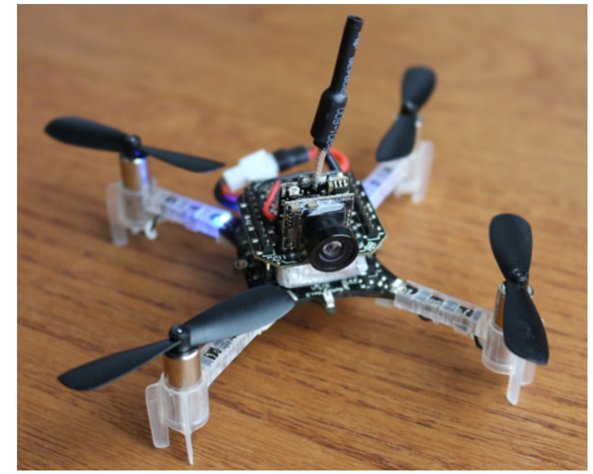
My Research Fields

- Human Computer Interaction
- (3D) Object Detection & Tracking and Pose Estimation
- 3D Reconstruction

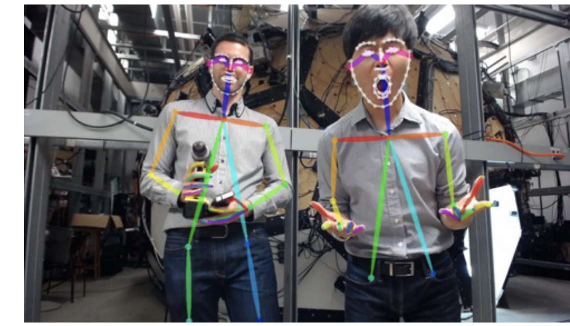
Human Computer Interaction

Human Computer Interaction

- Human Machine Interfaces
  - Tangible User Interfaces
  - Human UAV Interaction

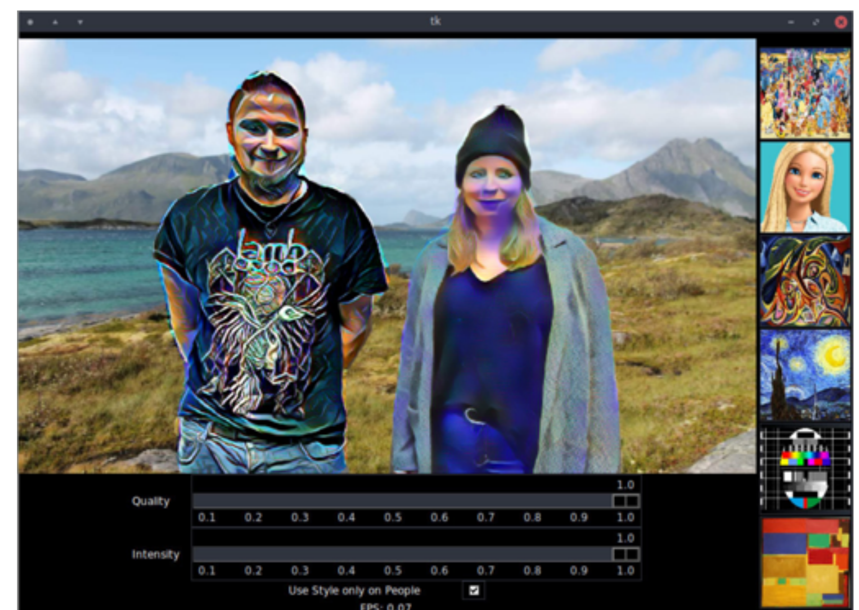


Microdrone with Camera



Cao et. al., OpenPose: realtime multi-person 2D pose estimation using Part Affinity Fields, CVPR 2017

- Mixed Reality
  - Visualization / Highlighting
  - Pose Estimation
  - Object Tracking



Stahl et. al., IST - Style Transfer with Instance Segmentation, ISPA 2019

(3D) Object Detection & Tracking and Pose Estimation

(3D) Object Detection & Tracking and Pose Estimation

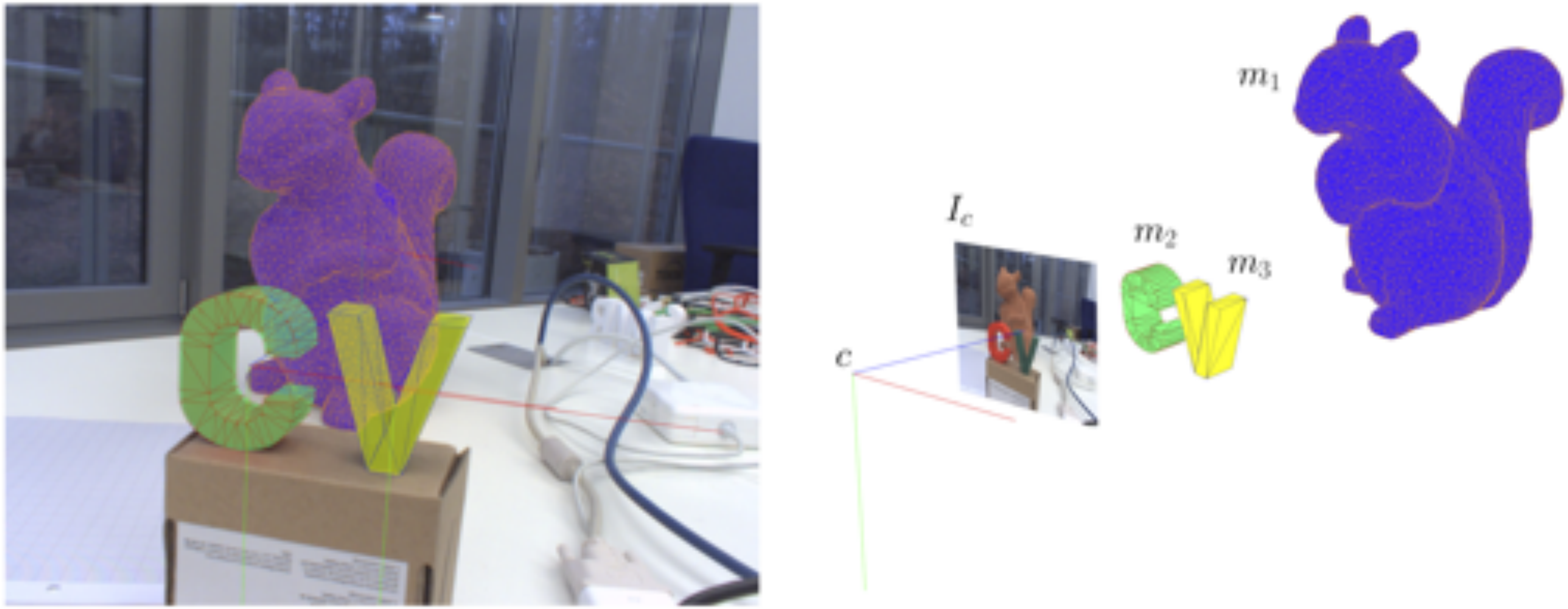
Direct Photometric Tracking [Dense Method]

- Minimize error between observed image I_I and reference image I_R with pose $(R, \mathbf{t})$, i.e.

$$E(R, \mathbf{t}) = \sum_{x \in \Omega} \left(I_O(\mathbf{x}) - I_R(\Pi(R\mathbf{X} + \mathbf{t}))\right)^2 \longrightarrow \min$$

Dense Photometric Tracking

Model-based Object-Tracking [Dense Method]



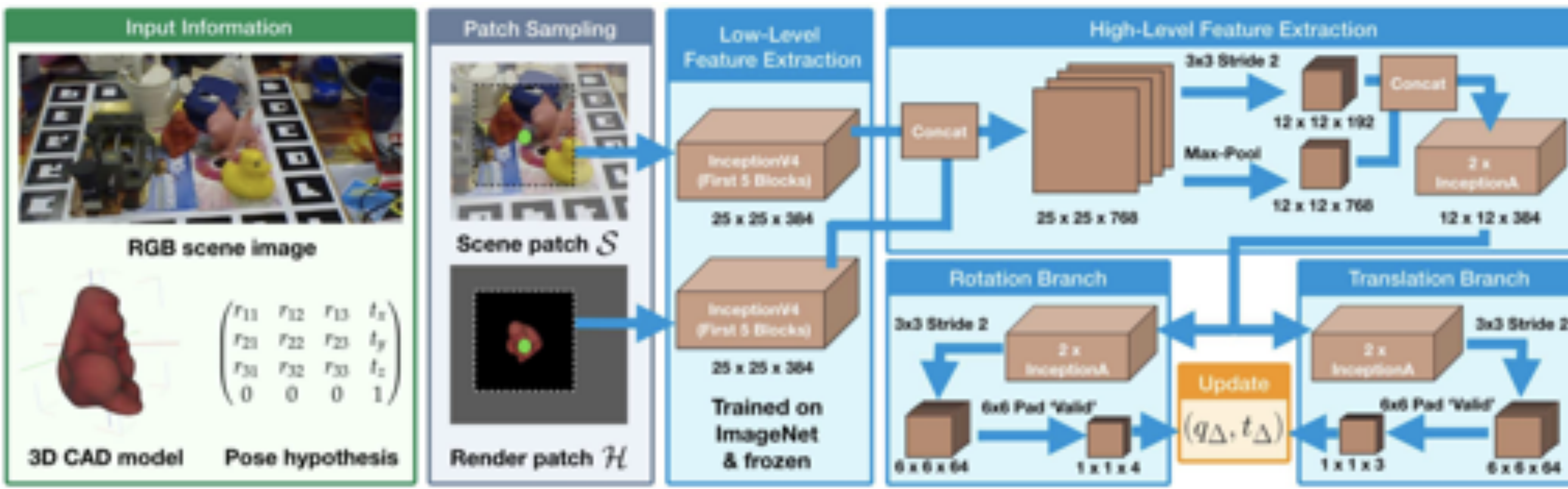
Model-based 3D Object Pose Estimation

$$P(\Phi^i, \mathbf{I}) = \prod_{\mathbf{x}_c \in \Omega} \Big(H_e(\Phi^i(\mathbf{x}_c)) P_f^i(\mathbf{x}_c) + (1 - H_e(\Phi^i(\mathbf{x}_c))) P_b^i(I(\mathbf{x}_c)) \Big) \longrightarrow \max$$

$$\text{with } \mathbf{x}_c = \Pi(K(T_{cm}\tilde{\mathbf{X}})_{3\times 1})$$

CNN supported Model-based 6D Pose Refinement

- Deep neural network to predict a translational and rotational update
  - Model-based 6D pose refinement using a contour-based approach
  - Networks are trained from purely synthetic data



F. Manhardt, W. Kehl, N. Navab, F. Tombari, Deep Model-Based 6D Pose Refinement in RGB, ECCV 2018

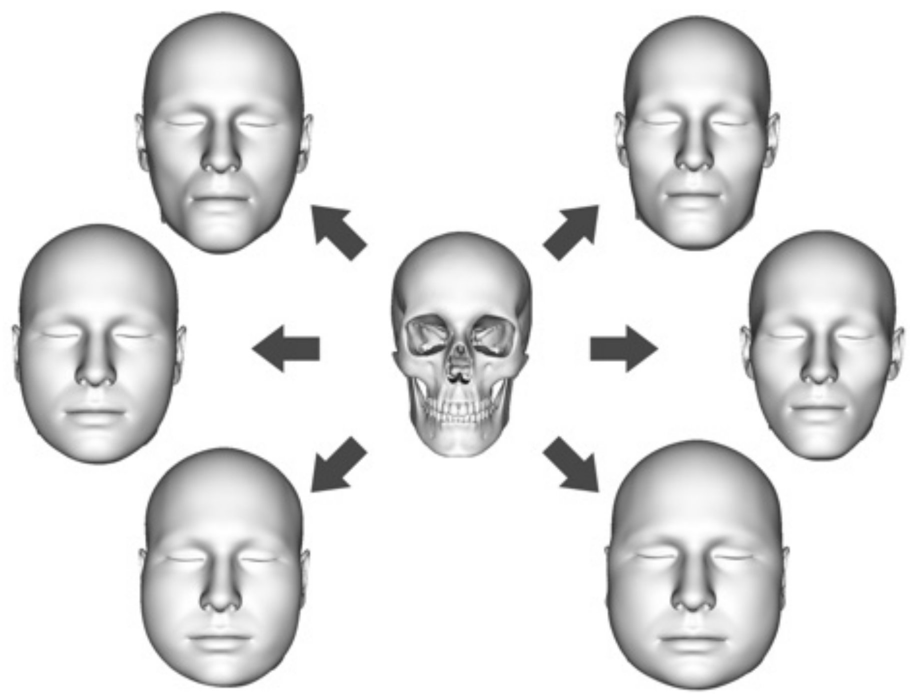
3D Reconstruction

3D Reconstruction

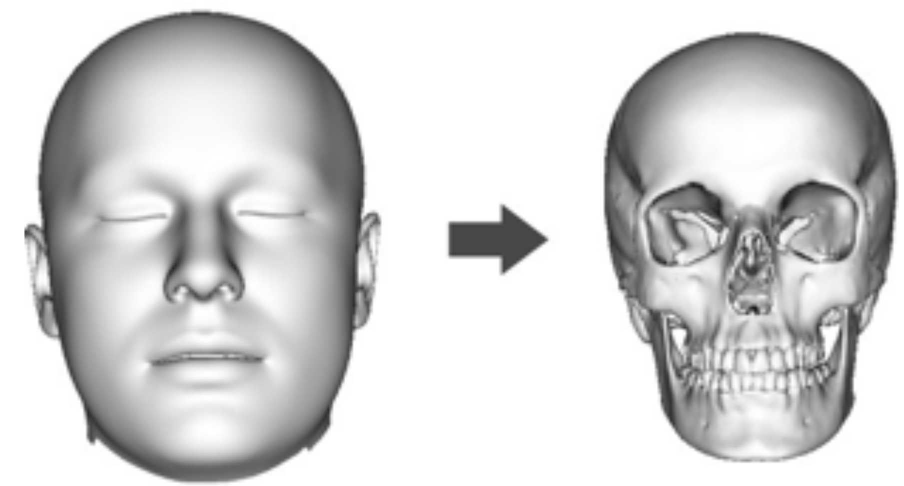
- Mainly research in the field of dentistry
- Intraoral (Surface) Scanner
  - Development of the world's samllest intraoral scanner
- Cone Beam Computed Tomography
  - Low dose reconstructions
  - Automatic calibration of a CBCT device
  - Artifact suppression [regularized reconstruction]
  - Recognition and compensation of patient movements
- Craniofacial Reconstruction
  - Without X-ray image
  - With one conventional X-ray image for regularization

Craniofacial Reconstruction

Craniofacial Reconstruction?



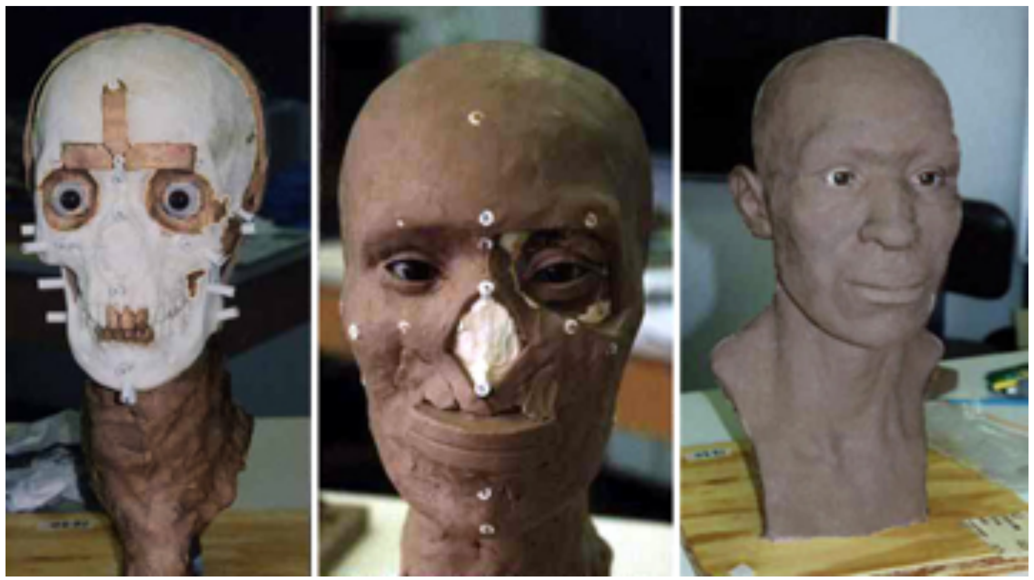
Infer skin from skull



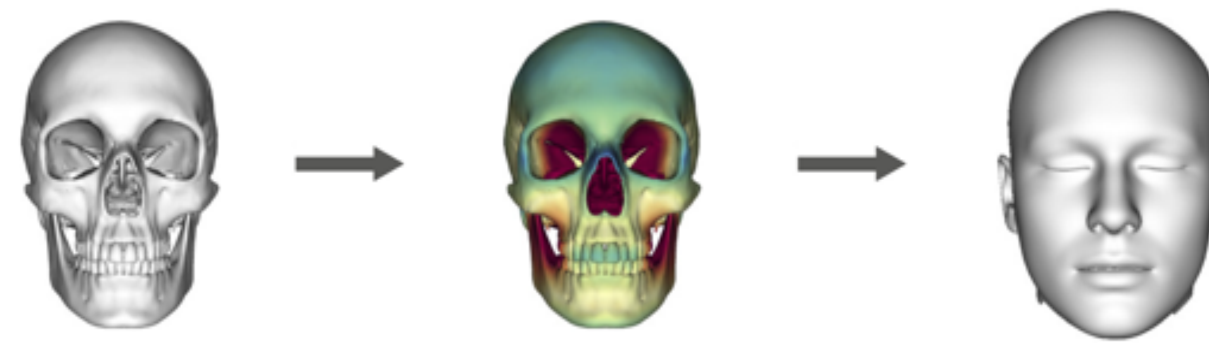
Infer skull from skin

BMBF Project *Kephalos*

Infer Skin from Skull

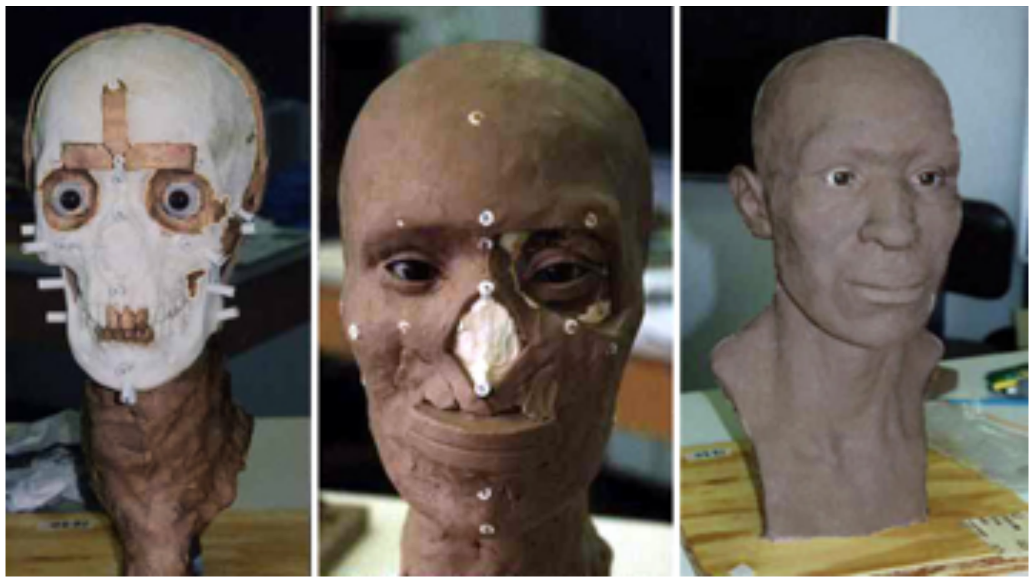


*Add facial soft tissue thickness (**FSTT**) by clay*
[Carrie Olsen, sculptor]

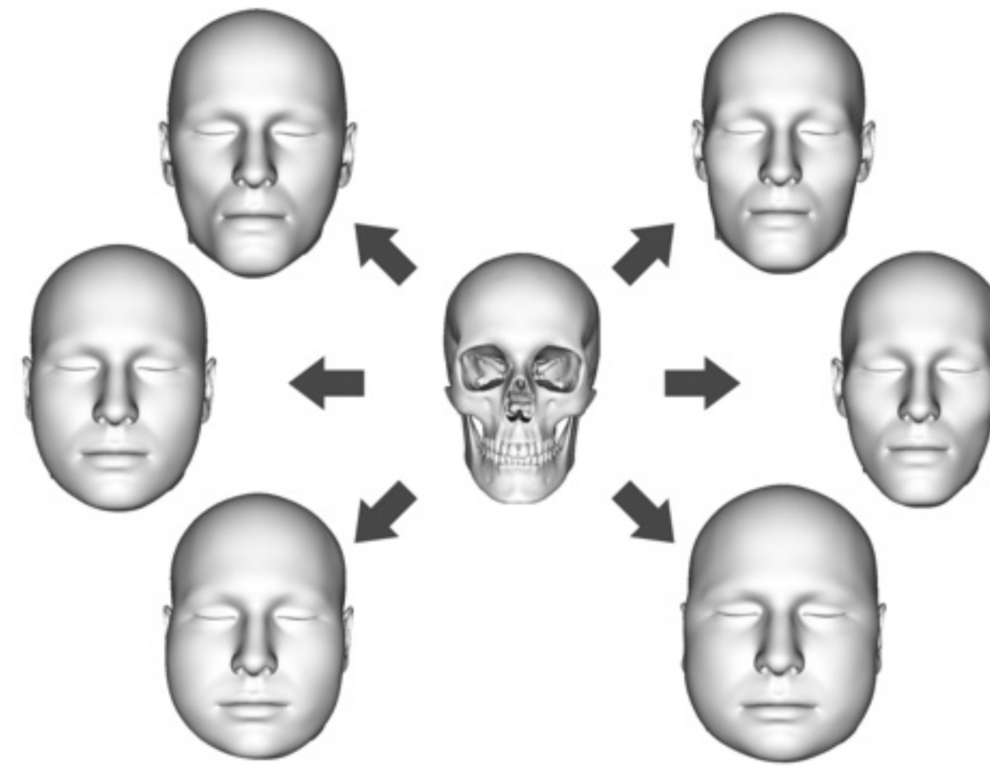


Skull + FSTT = Skin

Infer Skin from Skull



*Add facial soft tissue thickness (**FSTT**) by clay*
[Carrie Olsen, sculptor]

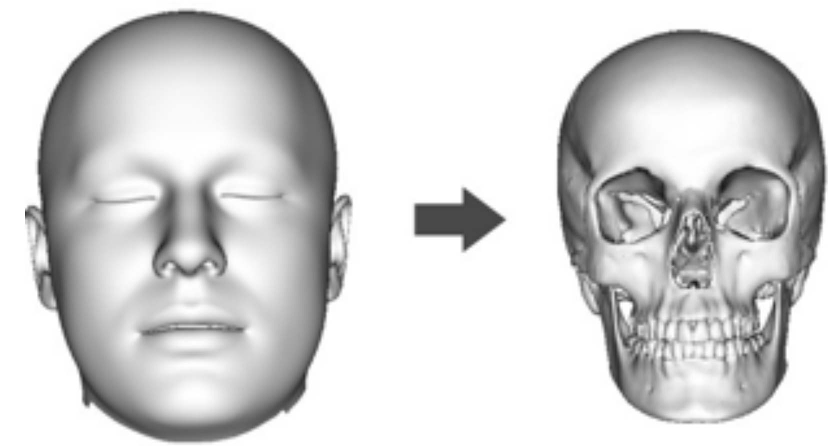


Virtual skin surface variants

Infer Skull from Skin

CT imaging

DVT imaging



Model-based skull estimate

Input Data

Input Data

- Skulls
  - 60 CT scans
  - 2 surface scans

Skull extracted from CT
(University Medical Center Mainz)

Input Data

- Skulls
  - 60 CT scans
  - 2 surface scans
- Heads
  - 43 CT scans
  - 39 surface scans

Skin extracted from CT
(University Medical Center Mainz)

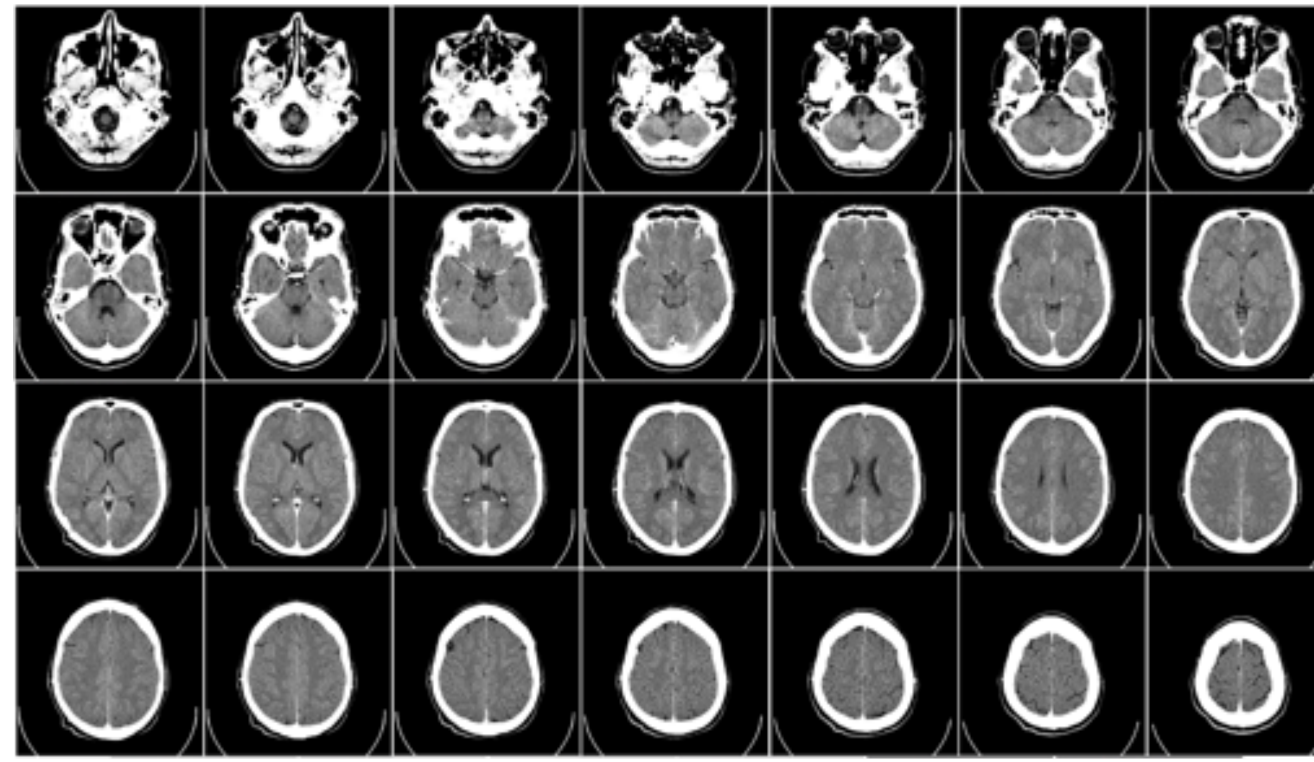
Input Data

- Skulls
  - 60 CT scans
  - 2 surface scans
- Heads
  - 43 CT scans
  - 39 surface scans

Head scan
(ten24 3D Scanstore)

Input Data

- Skulls
  - 60 CT scans
  - 2 surface scans
- Heads
  - 43 CT scans
  - 39 surface scans
- FSTT
  - 43 corresponding skull/head pairs from CT scans



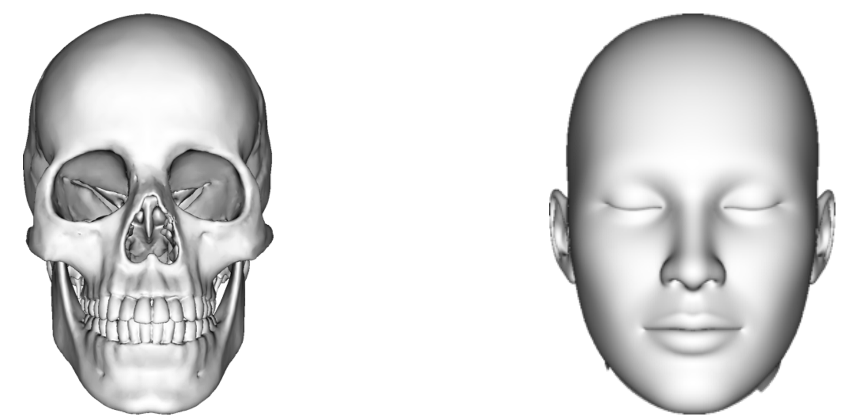
FSTT from CT scans

Input Data

- Skulls
  - 60 CT scans
  - 2 surface scans
- Heads
  - 43 CT scans
  - 39 surface scans
- FSTT
  - 43 corresponding skull/head pairs from CT scans

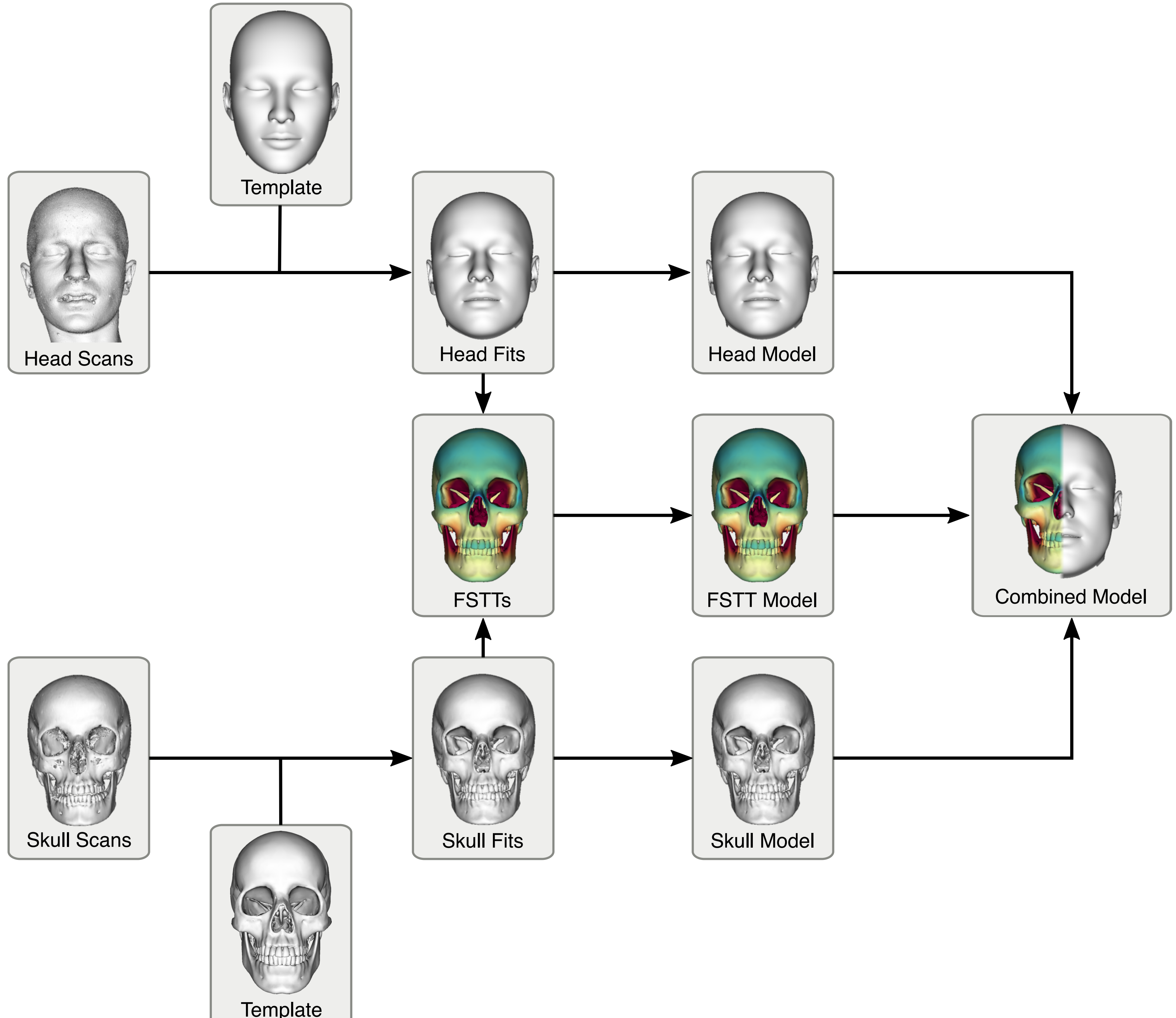
- Input models have different triangulations!

- Fit template models to input data



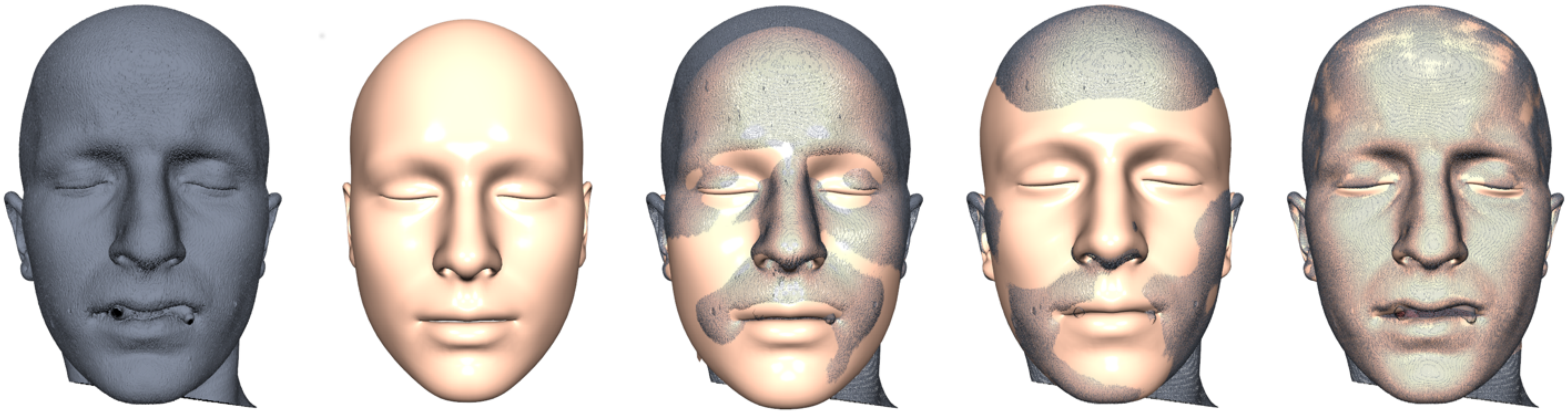
Volumetric skull template (69k vertices, tetrahedra)

Surface head template (25k vertices, triangles)


Template
Head Scans
Head Fits
Head Model
FSTTs
FSTT Model
Combined Model
Skull Scans
Skull Fits
Skull Model
Template

Template Fitting

Template Fitting

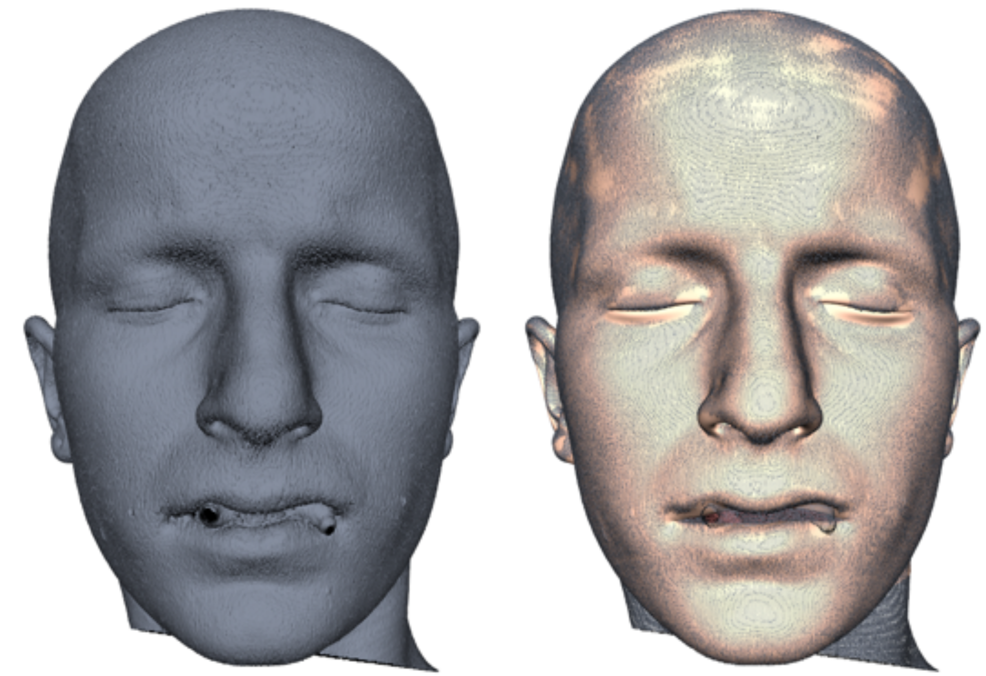


CT Scan *Template* *Coarse alignment* *PCA* *Fine-scale alignment*

Skull and Head Reconstructions

Fit skull template to 62 skulls

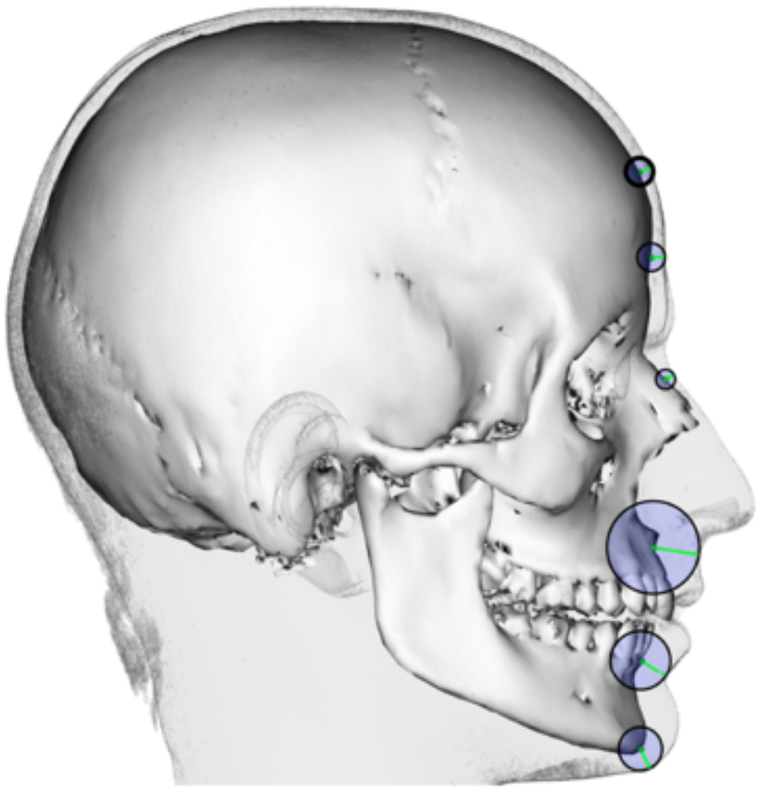
Fit head template to 82 heads

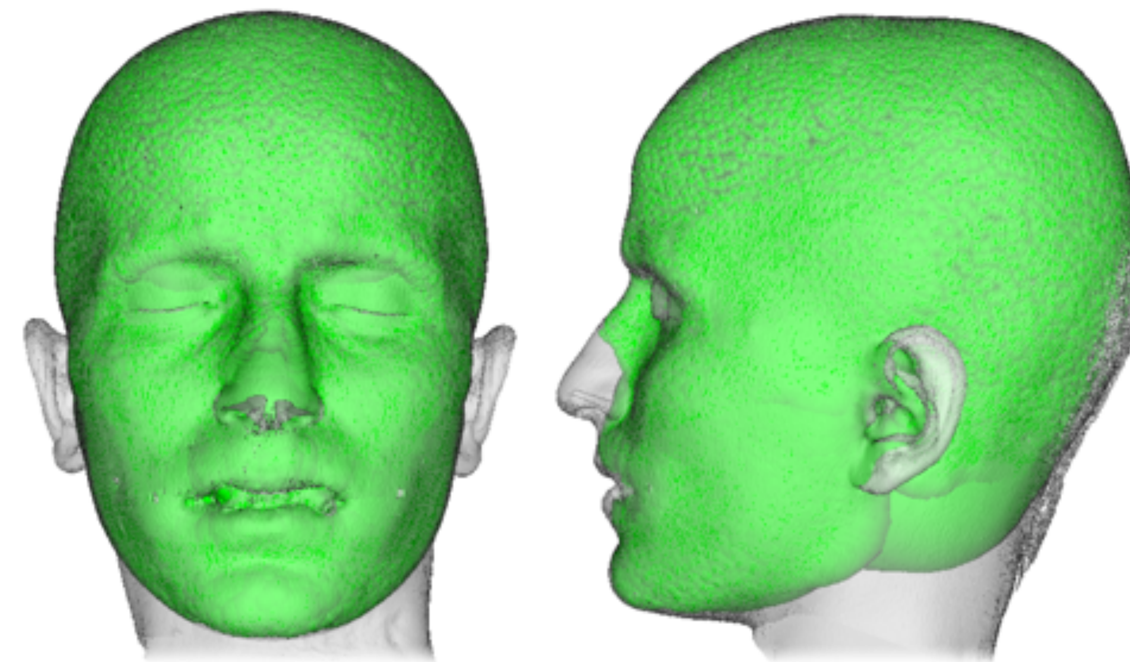


- Average RMS error < 0.5 mm in face area
- All scans/models have same triangulation
- Allows for statistical evaluation and model learning

Facial Soft Tissue Thickness

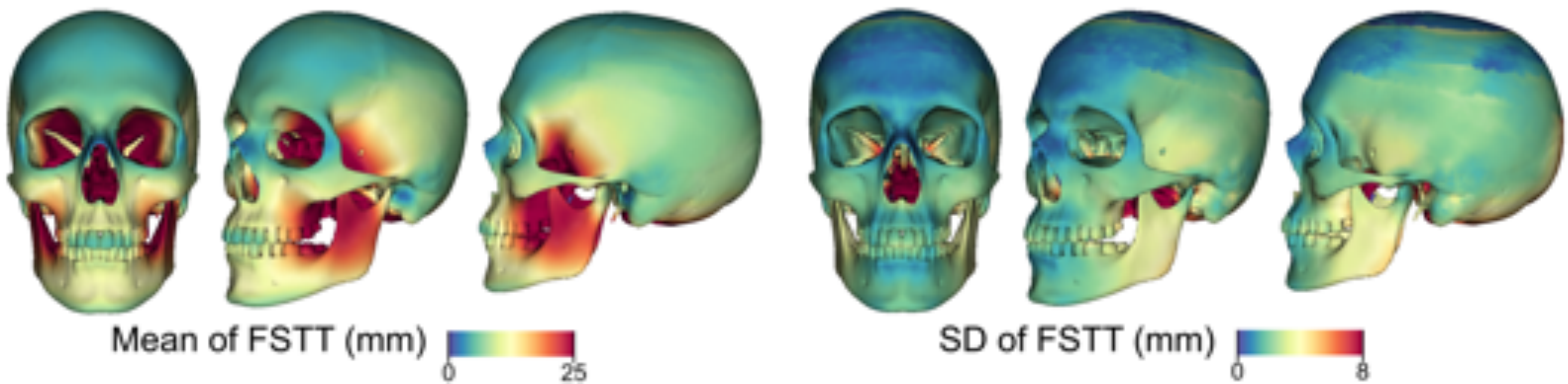
Facial Soft Tissue Thickness (FSTT)



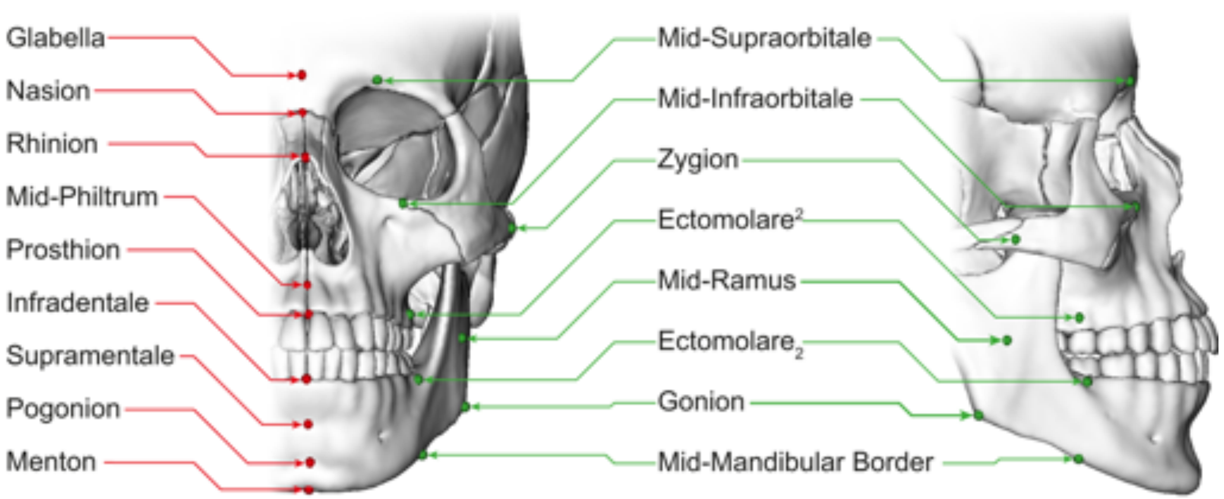


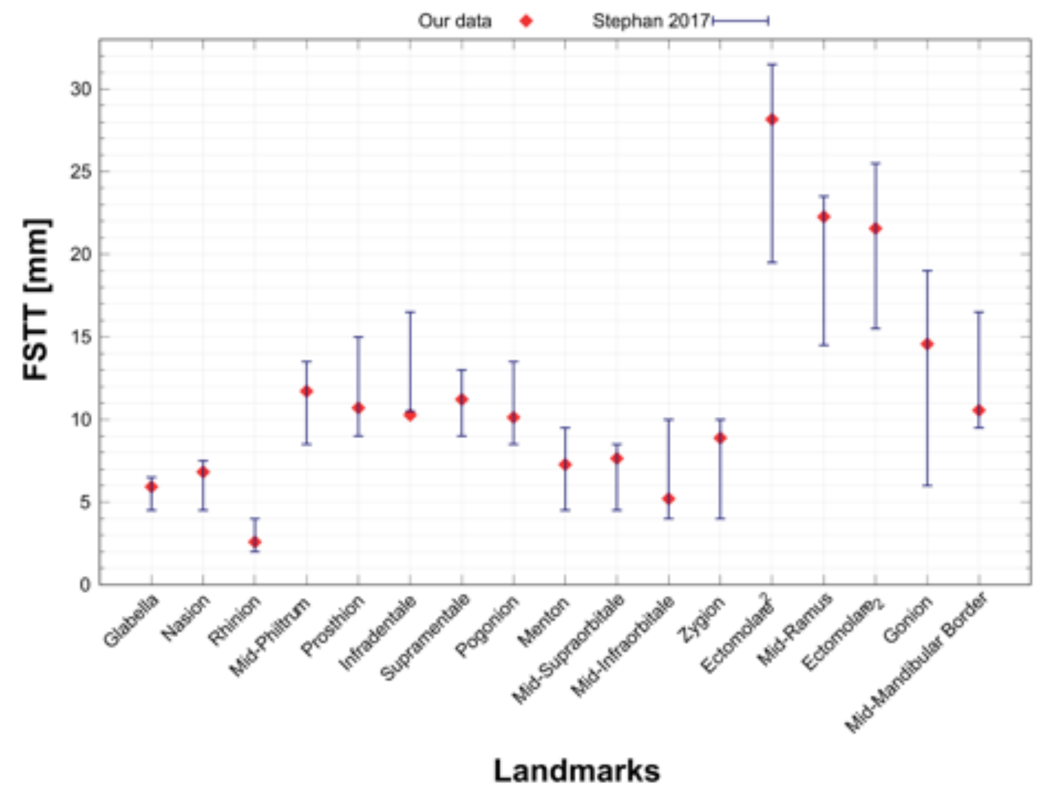
- Max-balls at outer skull vertices
- FSTT corresponds to ball radii

Facial Soft Tissue Thickness (FSTT)

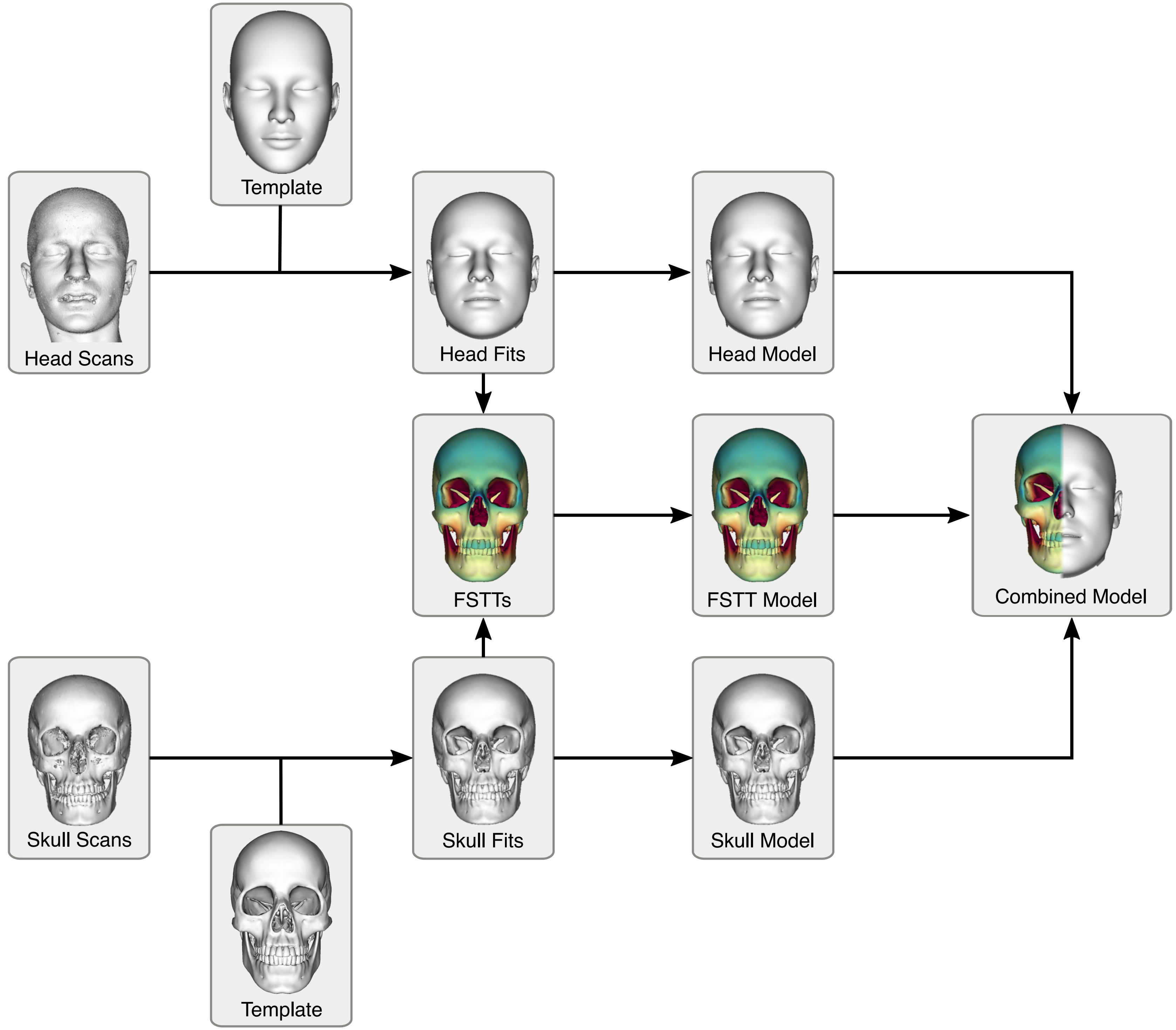


Facial Soft Tissue Thickness (FSTT)





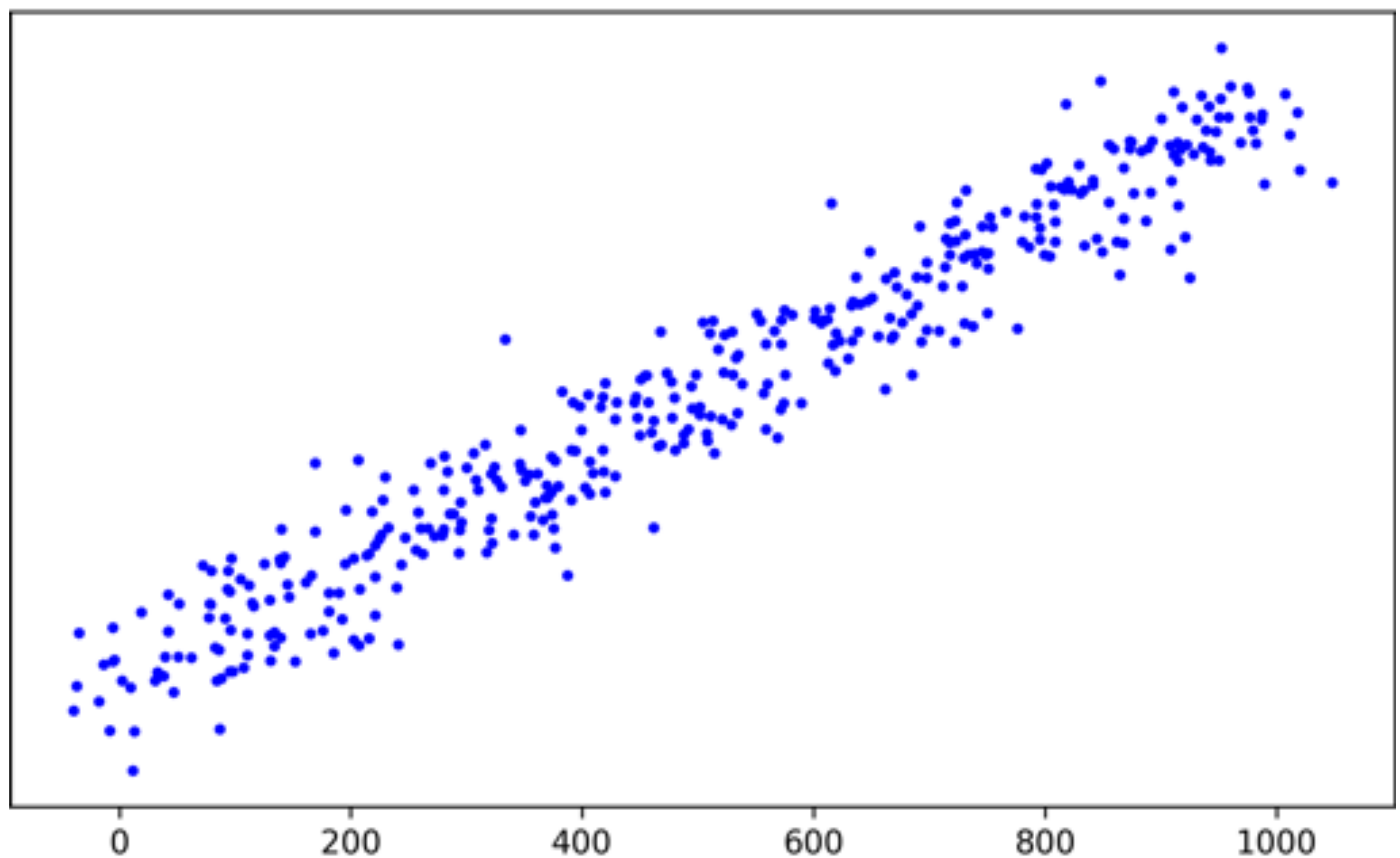
Model Learning



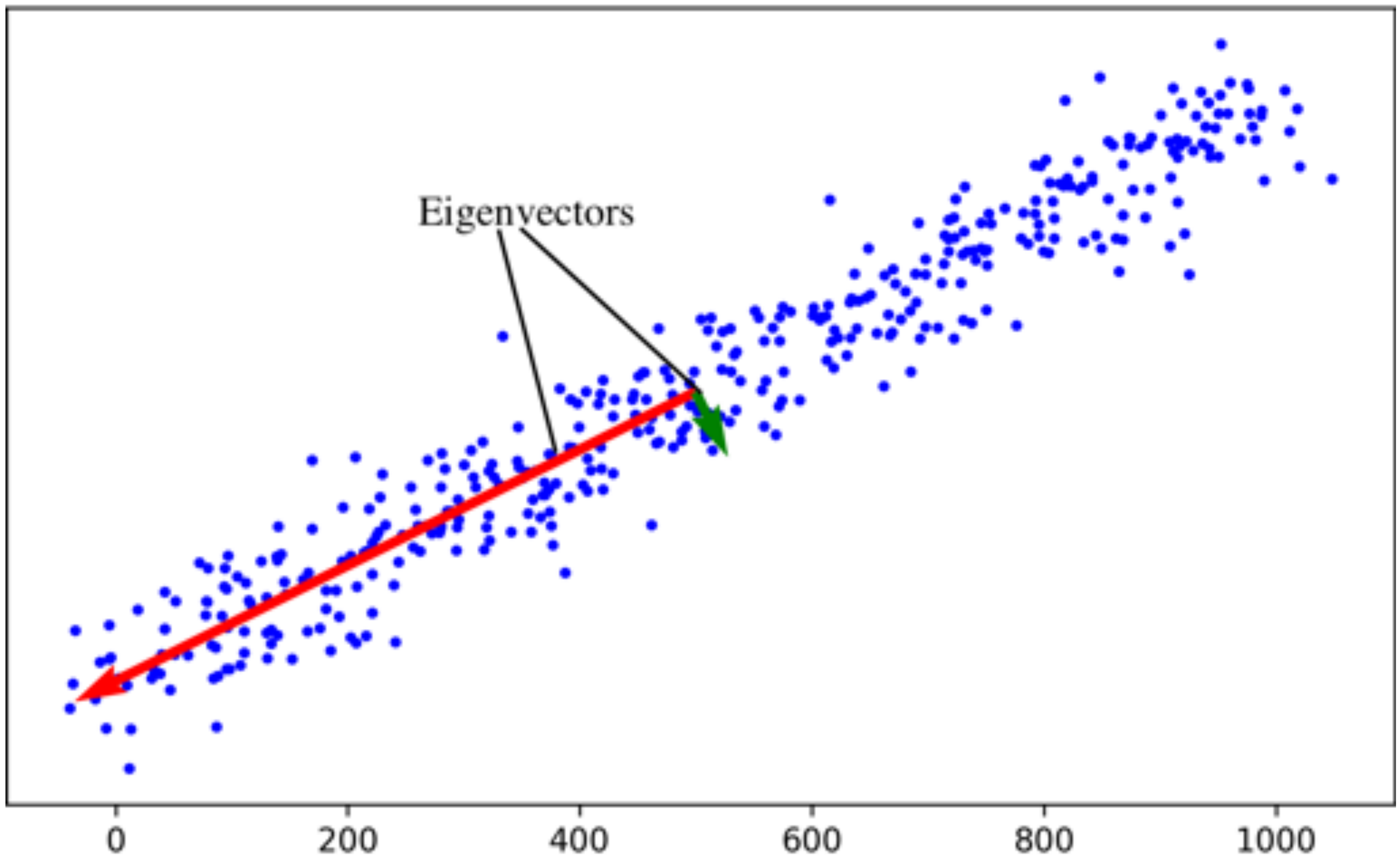
Template
Head Scans
Head Fits
Head Model
FSTTs
FSTT Model
Combined Model
Skull Scans
Skull Fits
Skull Model
Template

Side note [Principal Component Analysis (PCA)]

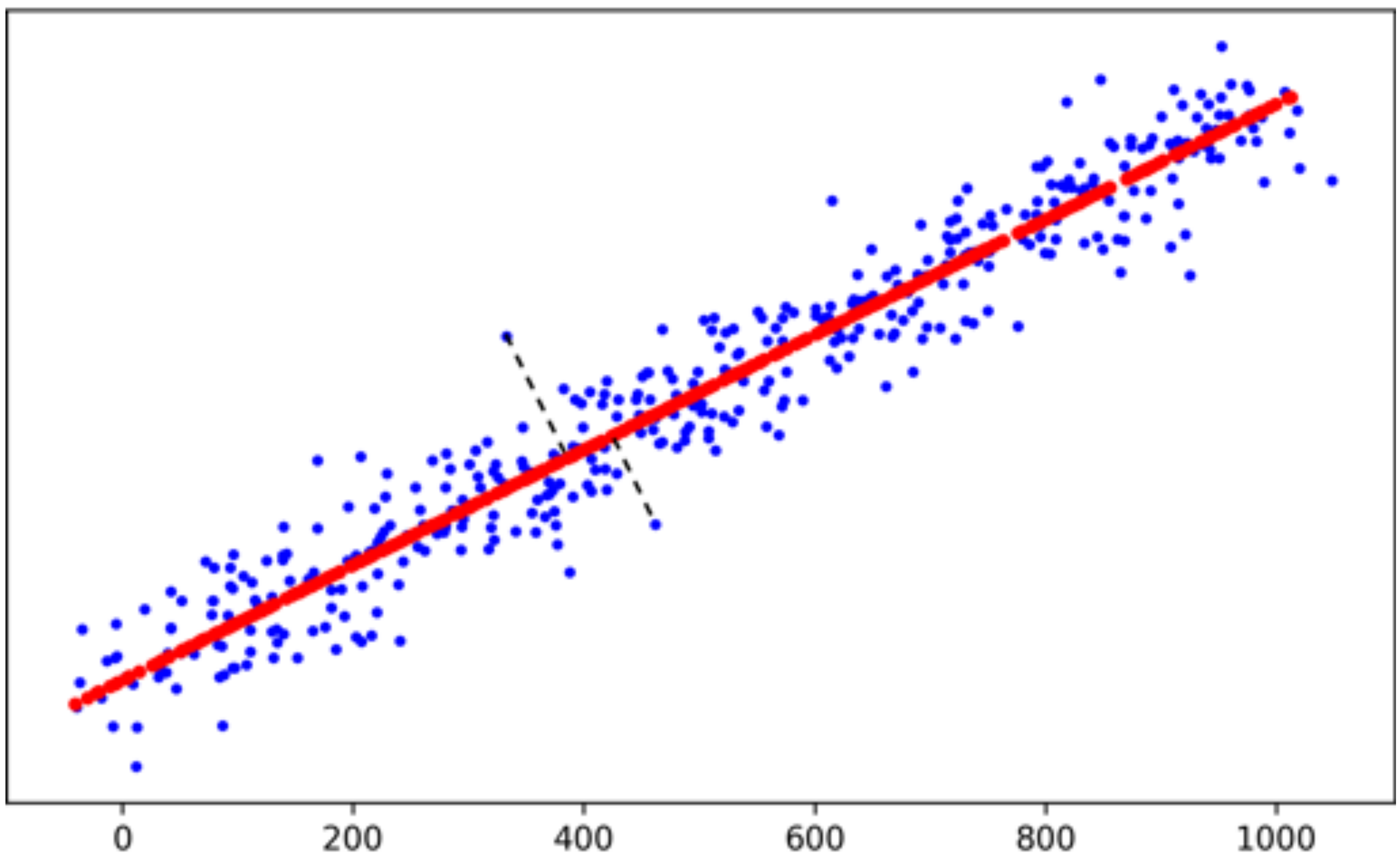
Data almost always comes with noise



PCA helps to extract relevant information



Project the data on the most relevant subspace



Pictures …

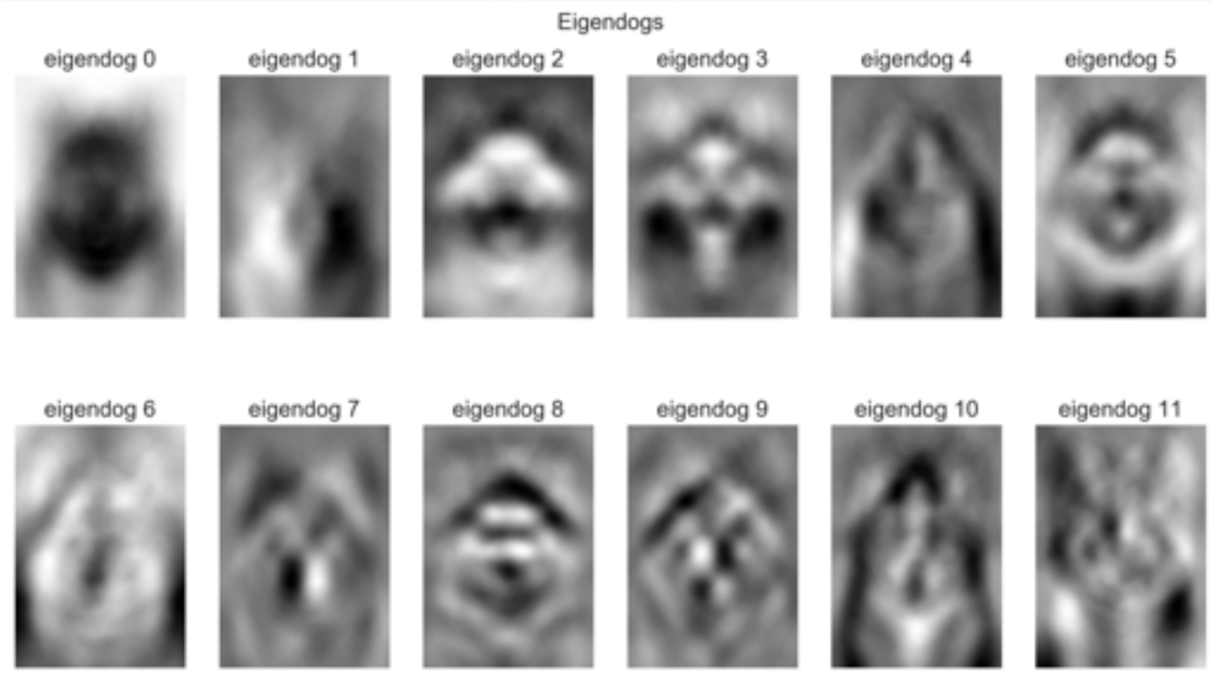
- … are elements of a high dimensional vector space $\mathbb{R}^{\#Pixels}$

- Make thousands of similar dogs by rotating, flipping, scaling, … the images

Example from Andrew Glassner, *Deep learning: a crash course*, SIGGRAPH 2018

“Most important” components of the (dog) images ...

- … are the eigenvectors (eigendogs $\in \mathbb{R}^{\#Pixels}$) that can be found by PCA

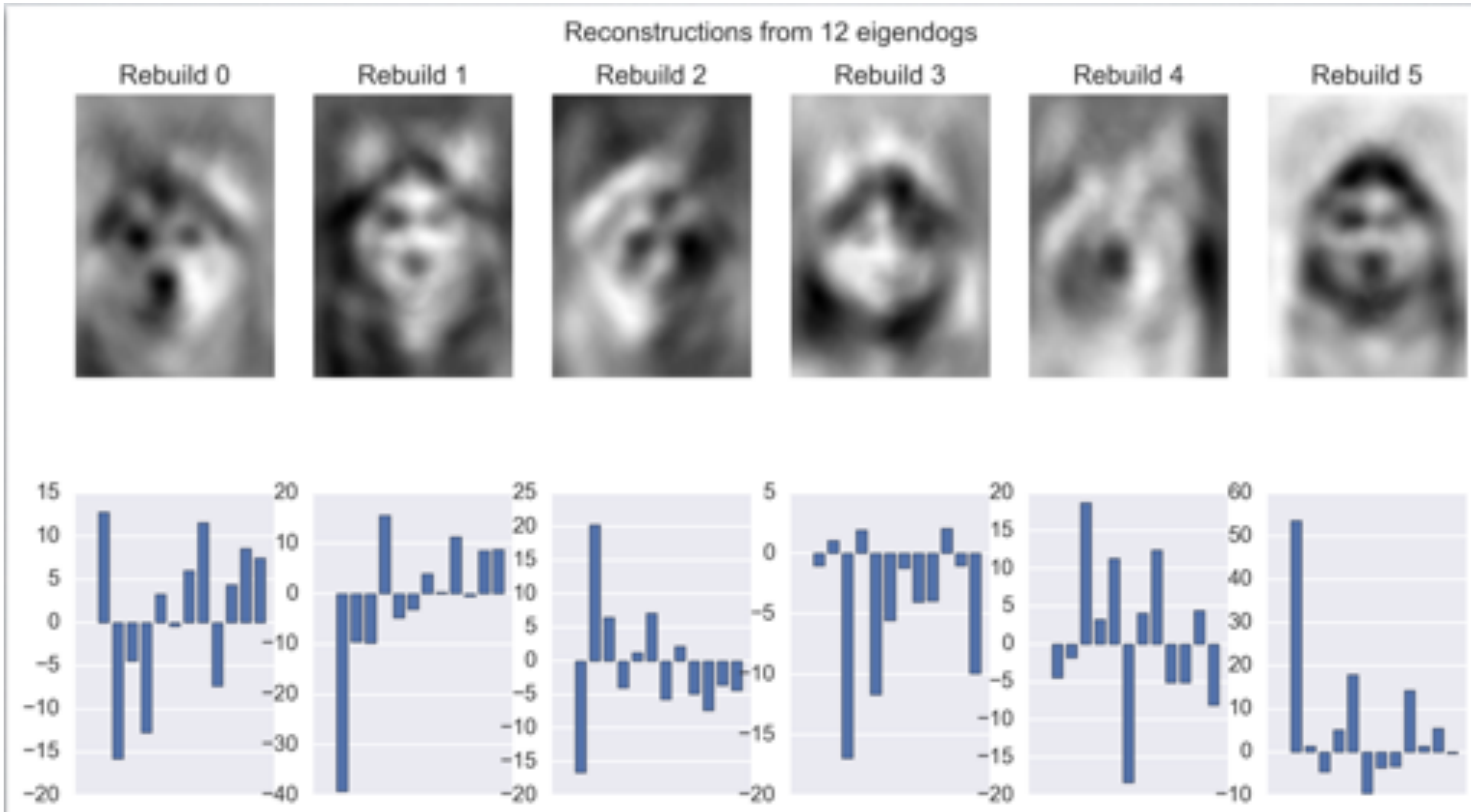


The first 12 eigendogs

- Any of the inputs can be recreated by a weighted sum of eigenvectors
  - Here we need just 12 numbers (weights) (and the 12 eigendogs) to descrive any input image
  - PCA will tell us how to weight the images to recover any ot the input images
    - Project input image onto the different eigendogs: Dot product of image and respective eigendogs

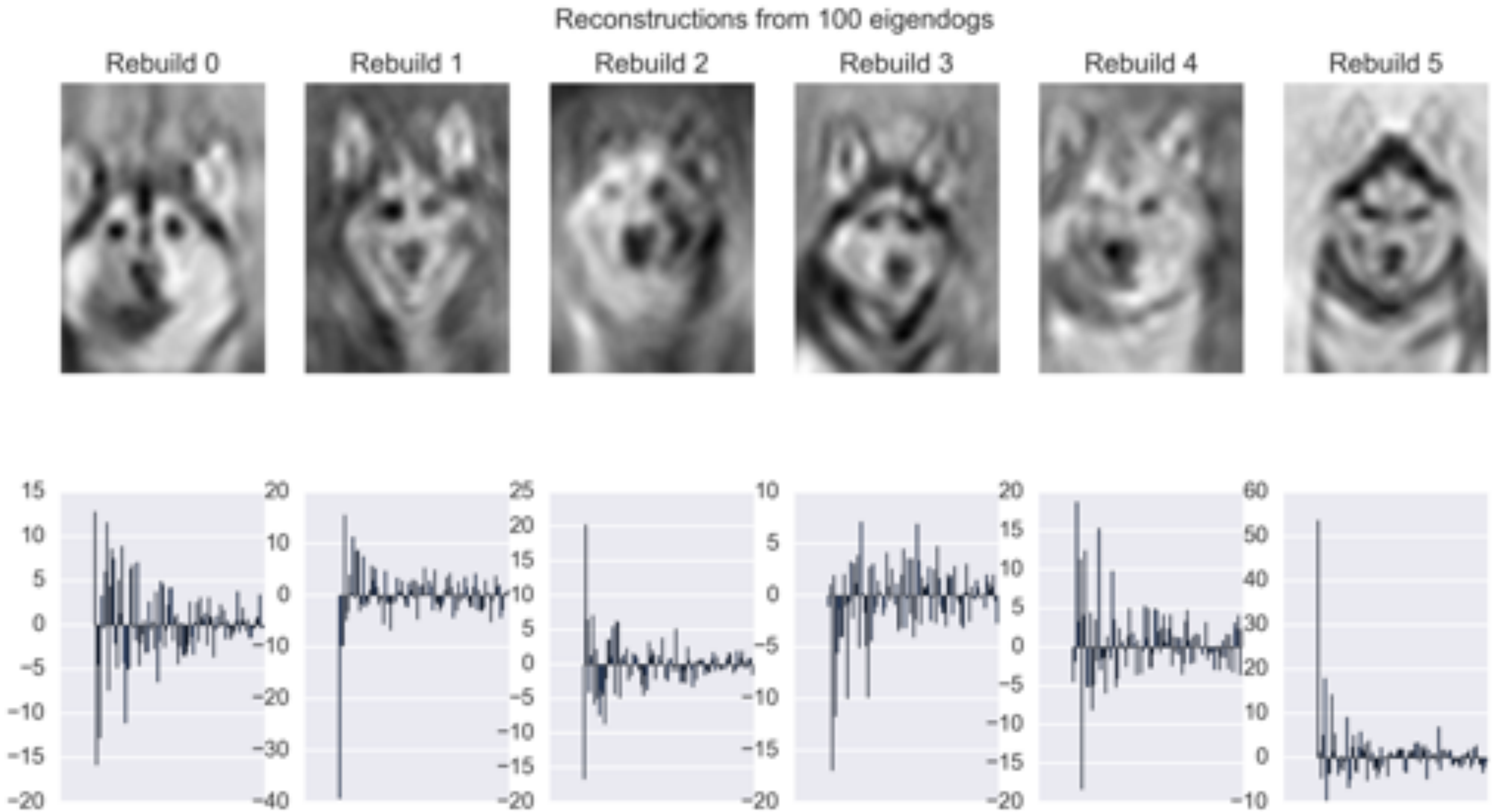
Example from Andrew Glassner, *Deep learning: a crash course*, SIGGRAPH 2018

More eigenvectors (eigendogs) result in more details



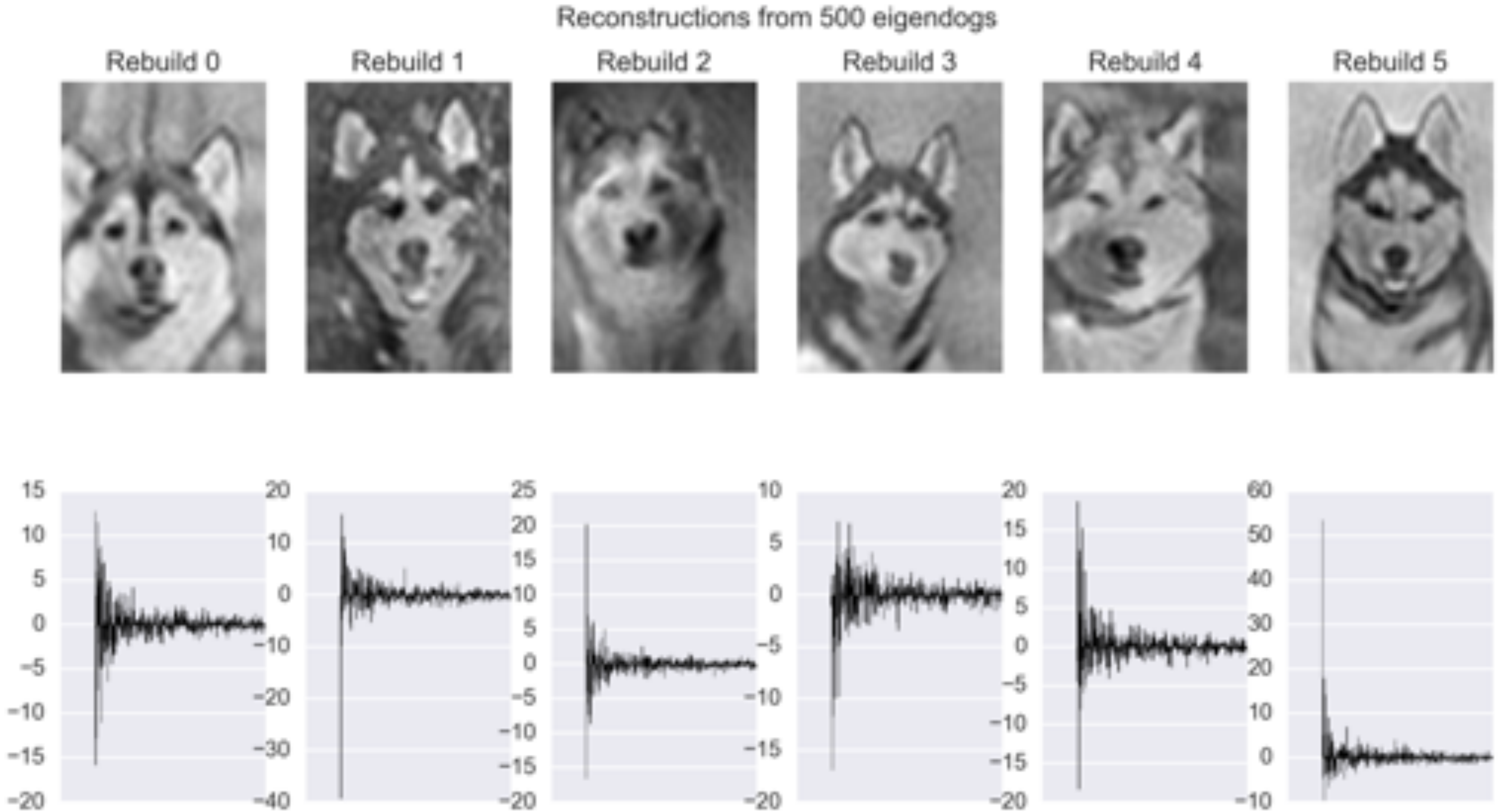
Reconstructions from 12 eigendogs

More eigenvectors (eigendogs) result in more details



Reconstructions from 100 eigendogs

More eigenvectors (eigendogs) result in more details



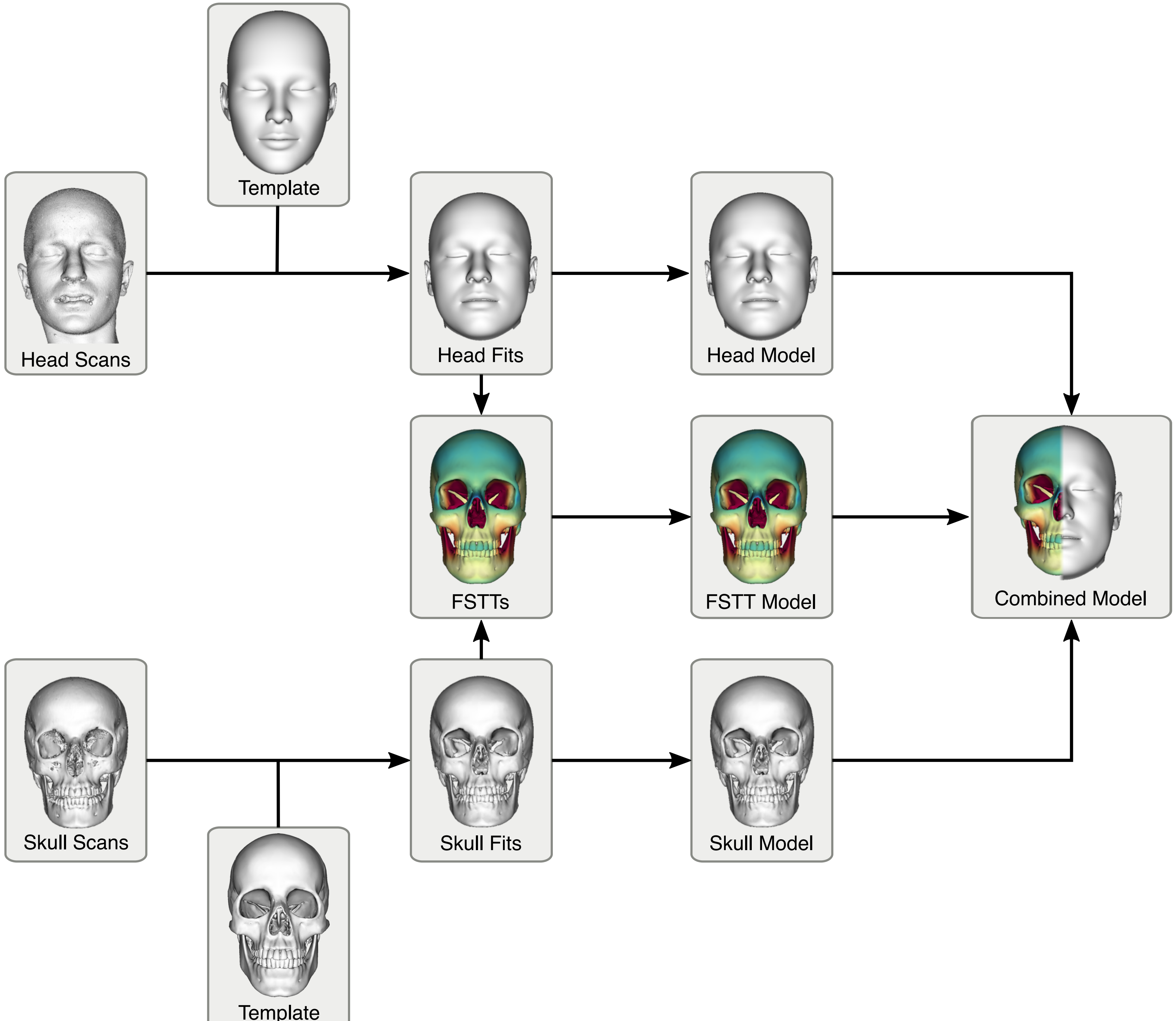
Reconstructions from 500 eigendogs

More eigenvectors (eigendogs) result in more details

500 numbers per image plus 500 eigendog images to approximate any input image

- Every image is represented by 500 numbers (together with the 500 eigendogs)
    - To store 500 eigendog images together with 500 numbers per image is much more efficient than to store e.g. 100.000 complete input images
- Any combination of 500 values is likely to produce an image of a Husky

Model Learning


Template
Head Scans
Head Fits
Head Model
FSTTs
FSTT Model
Combined Model
Skull Scans
Skull Fits
Skull Model
Template

Linear Skull Model

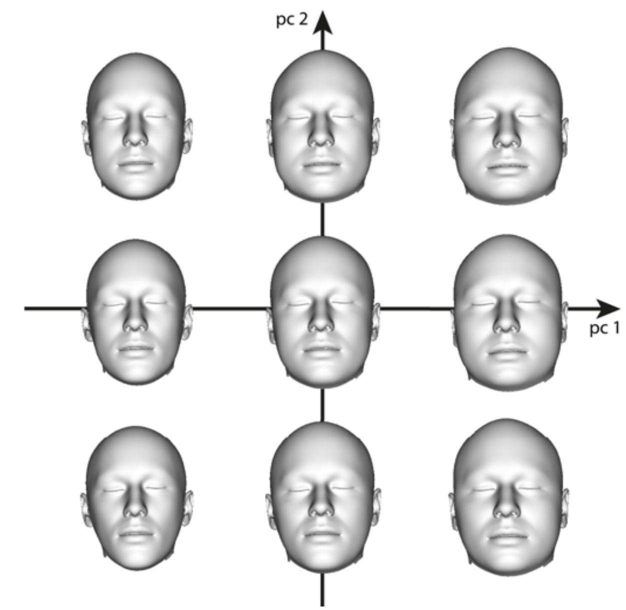
1. Mean-center the 62 training skulls
   - $\bar{\mathbf{s}} = \frac{1}{62} \sum_{i=1}^{62} \mathbf{s}_i$
   - $\hat{\mathbf{s}}_i = \mathbf{s}_i - \bar{\mathbf{s}}$
2. Construct and decompose data matrix
   - $\mathbf{D} = \left[\, \hat{\mathbf{s}}_1 \;\; \hat{\mathbf{s}}_2 \;\; \cdots \;\; \hat{\mathbf{s}}_{62} \,\right]$
   - $\mathbf{D} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^{\mathsf{T}} = \mathbf{U}_1\boldsymbol{\Sigma}\mathbf{U}_2$
3. Build model matrix
   - $\mathbf{M}_{\text{skull}} = \mathbf{D} \cdot \mathbf{U}_2^{\mathsf{T}}$
   - $\mathbf{s}(\boldsymbol{\alpha}) = \bar{\mathbf{s}} + \mathbf{M}_{\text{skull}} \cdot \boldsymbol{\alpha}$

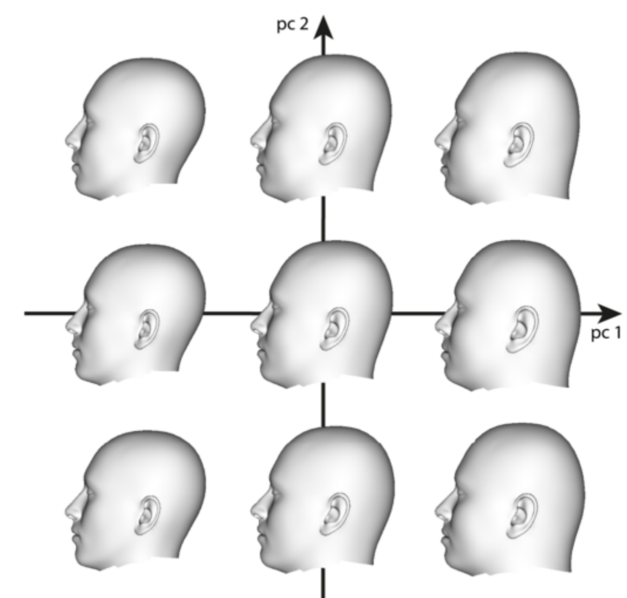
Linear Head Model

1. Mean-center the 82 training heads
   - $\bar{\mathbf{h}} = \frac{1}{82} \sum_{i=1}^{82} \mathbf{h}_i$
   - $\hat{\mathbf{h}}_i = \mathbf{h}_i - \bar{\mathbf{h}}$
2. Construct and decompose data matrix
   - $\mathbf{D} = \begin{bmatrix} \hat{\mathbf{h}}_1 & \hat{\mathbf{h}}_2 & \cdots & \hat{\mathbf{h}}_{82} \end{bmatrix}$
   - $\mathbf{D} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\mathsf{T} = \mathbf{U}_1\mathbf{\Sigma}\mathbf{U}_2$
3. Build model matrix
   - $\mathbf{M}_{\text{head}} = \mathbf{D} \cdot \mathbf{U}_2^\mathsf{T}$
   - $\mathbf{h}(\boldsymbol{\gamma}) = \bar{\mathbf{h}} + \mathbf{M}_{\text{head}} \cdot \boldsymbol{\gamma}$

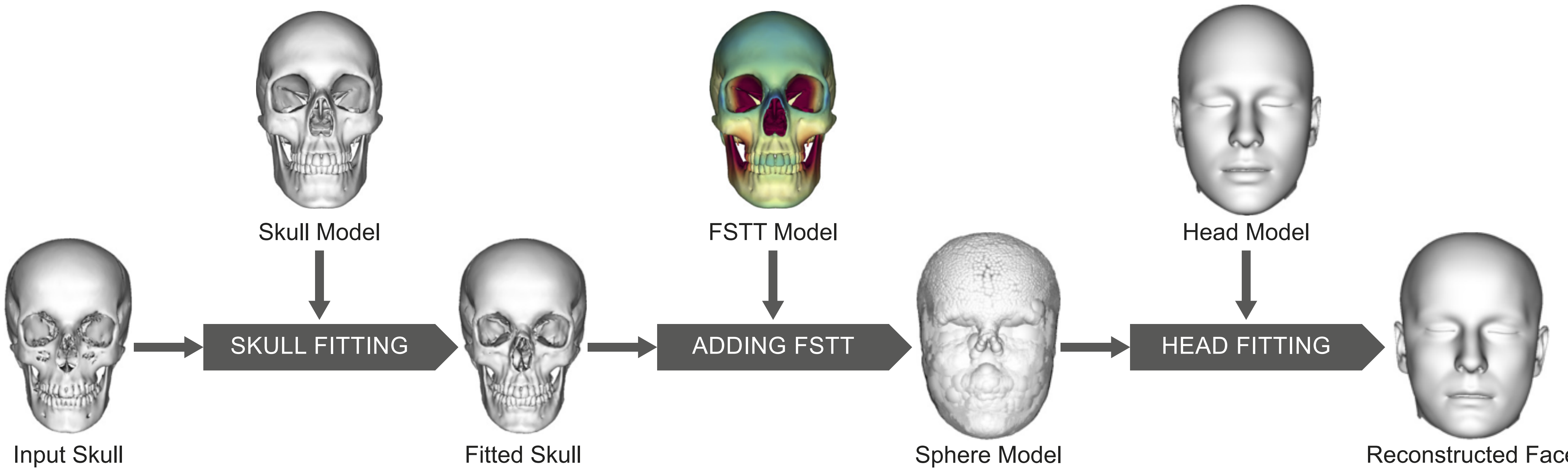
Linear FSTT Model

1. Mean-center the 43 training FSTTs
   - $\bar{\mathbf{f}} = \frac{1}{43}\sum_{i=1}^{43}\mathbf{f}_i$
   - $\hat{\mathbf{f}}_i = \mathbf{f}_i - \bar{\mathbf{f}}$
2. Construct and decompose data matrix
   - $\mathbf{D} = \left[\hat{\mathbf{f}}_1\ \hat{\mathbf{f}}_2\ \cdots\ \hat{\mathbf{f}}_{43}\right]$
   - $\mathbf{D} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^\mathsf{T} = \mathbf{U}_1\boldsymbol{\Sigma}\mathbf{U}_2$
3. Build model matrix
   - $\mathbf{M}_{\text{fstt}} = \mathbf{D}\cdot\mathbf{U}_2^\mathsf{T}$
   - $\mathbf{f}(\boldsymbol{\beta}) = \bar{\mathbf{f}} + \mathbf{M}_{\text{fstt}}\cdot\boldsymbol{\beta}$



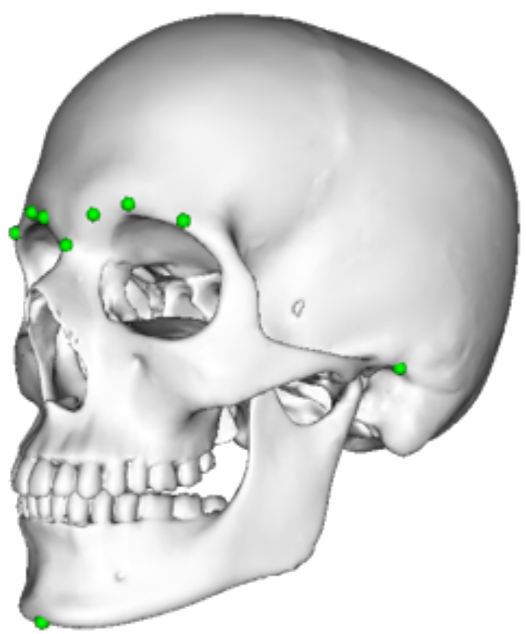


Craniofacial Reconstruction

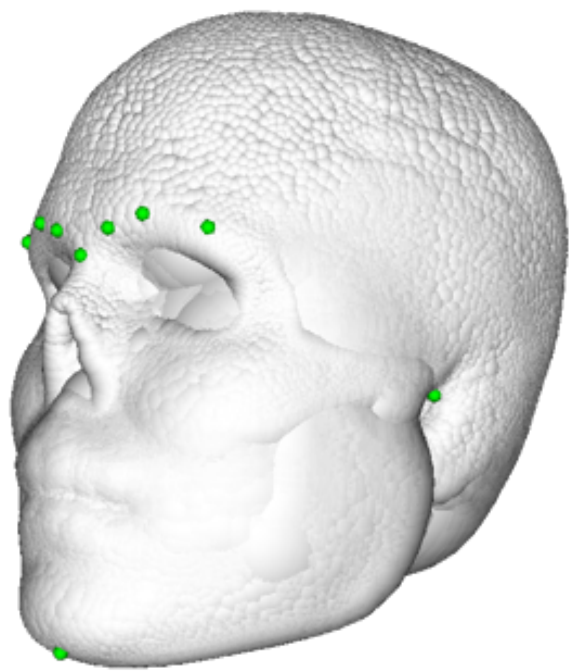


- Regularize skull/head fitting by PCA models
- Choose plausible FSTT distributions
- Automatic landmarks, no manual work

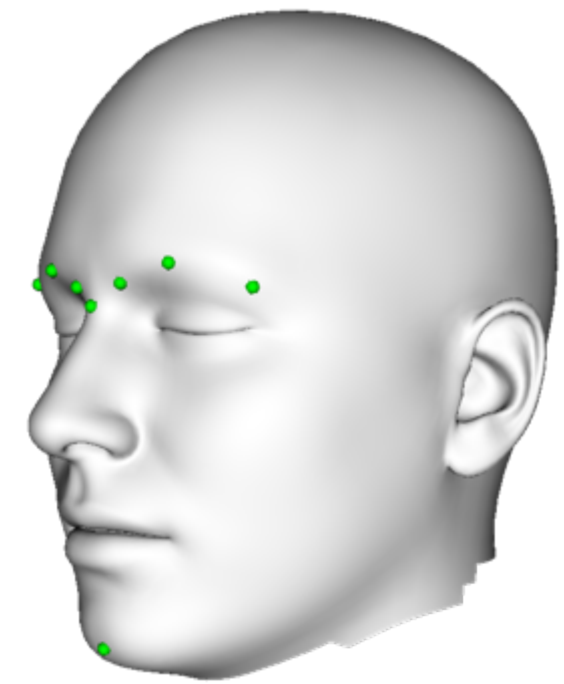
Craniofacial Reconstruction



Input skull

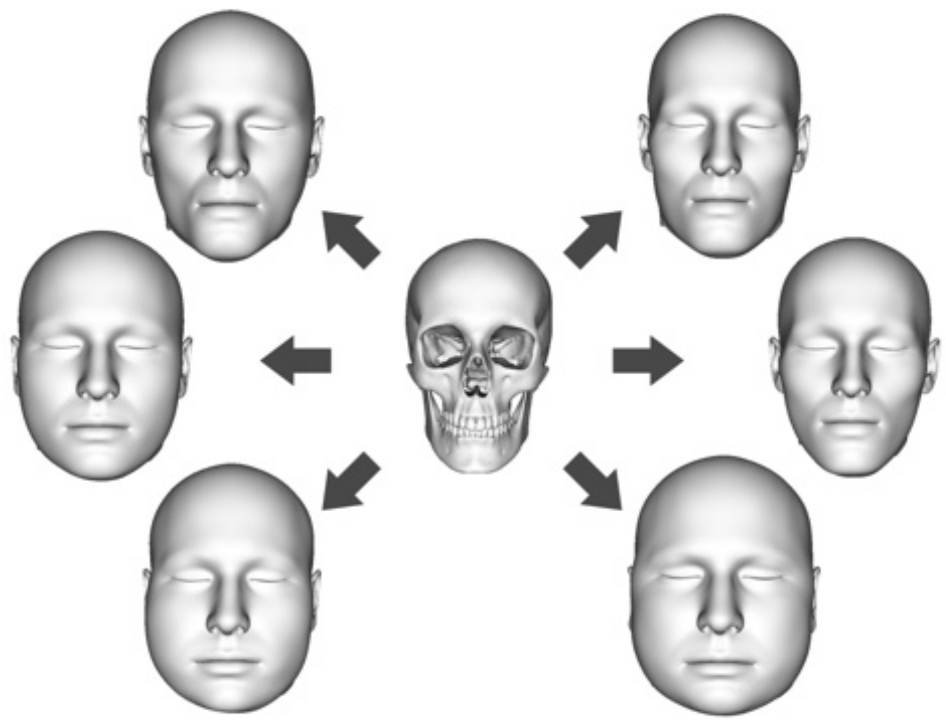


Add FSTT

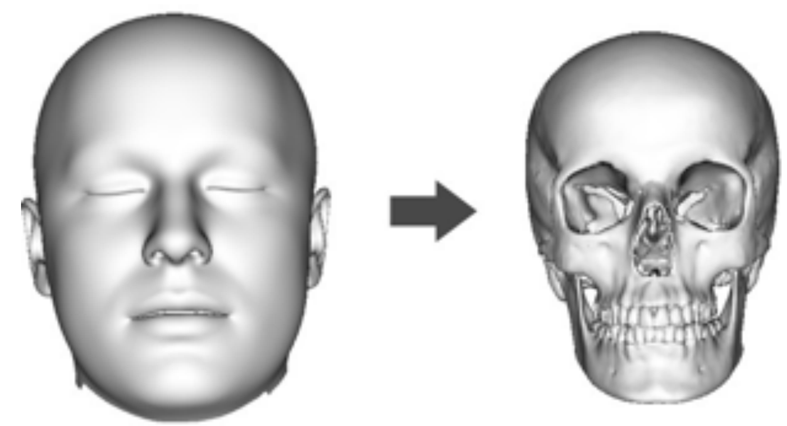


Fit skin

Are we done?

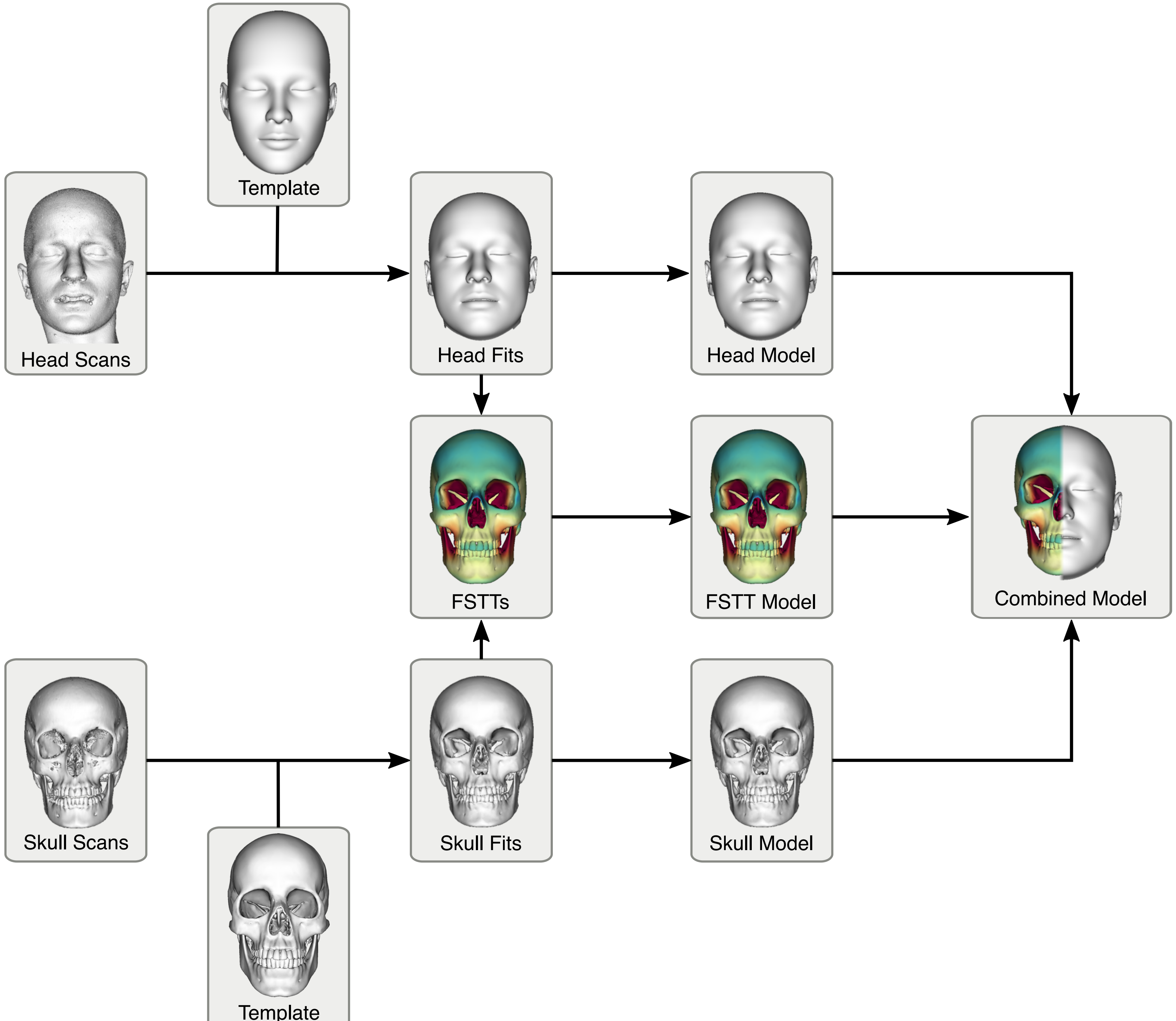


Infer skin from skull

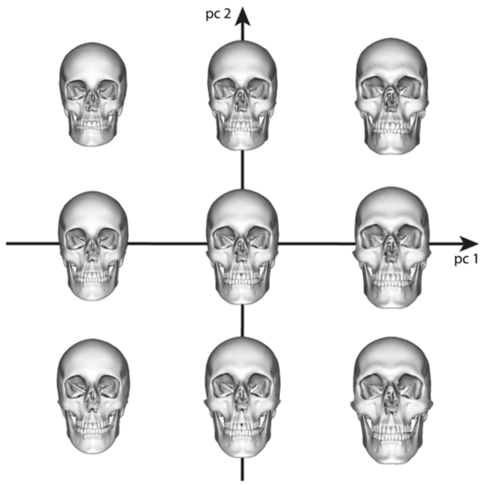


Infer skull from skin

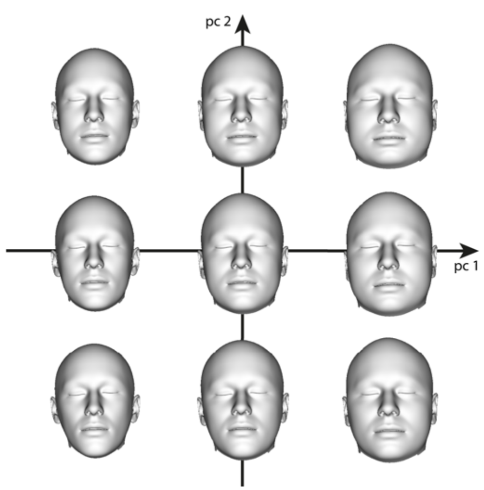
Multilinear Model


Template
Head Scans
Head Fits
Head Model
FSTTs
FSTT Model
Combined Model
Skull Scans
Skull Fits
Skull Model
Template

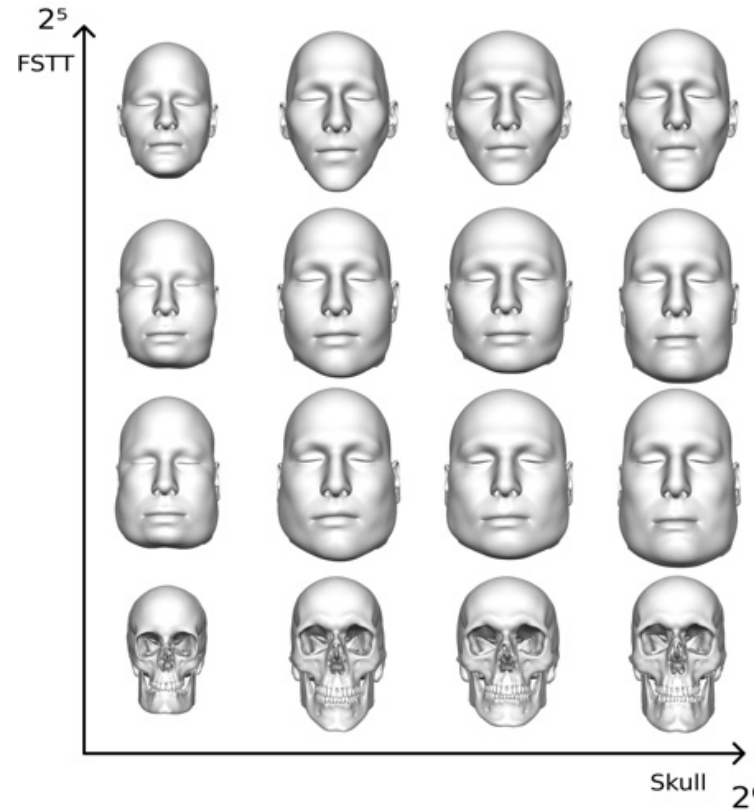
Generate a Multilinear Model



Linear skull model $\mathbf{s}(\boldsymbol{\alpha})$



Linear FSTT model $\mathbf{f}(\boldsymbol{\beta})$



Generate synthetic training data
$\mathbf{x}(\boldsymbol{\alpha}_j, \boldsymbol{\beta}_k)$ *for 64 skulls × 32 FSTTs*
(2048 training data)

Multilinear skull/head model

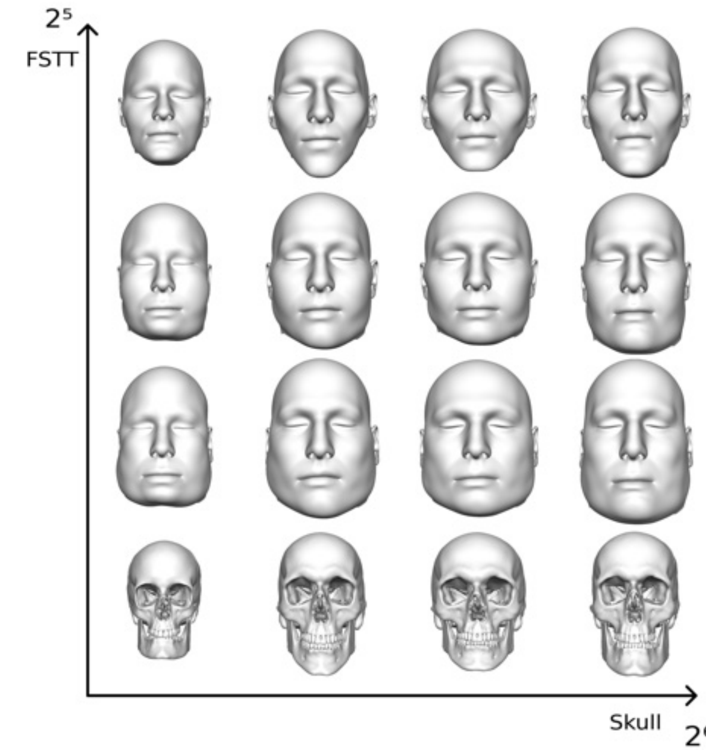
$$\mathbf{x}(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \begin{pmatrix} \mathbf{s}(\boldsymbol{\alpha}) \\ \mathbf{s}(\boldsymbol{\alpha}) \oplus \mathbf{f}(\boldsymbol{\beta}) \end{pmatrix}$$

Generate a Multilinear Model

1. Mean-center the 2048 training models
   - $\bar{\mathbf{x}} = \frac{1}{2048} \sum_{i=1}^{2048} \mathbf{x}_i$
   - $\hat{\mathbf{x}}_i = \mathbf{x}_i - \bar{\mathbf{x}}$

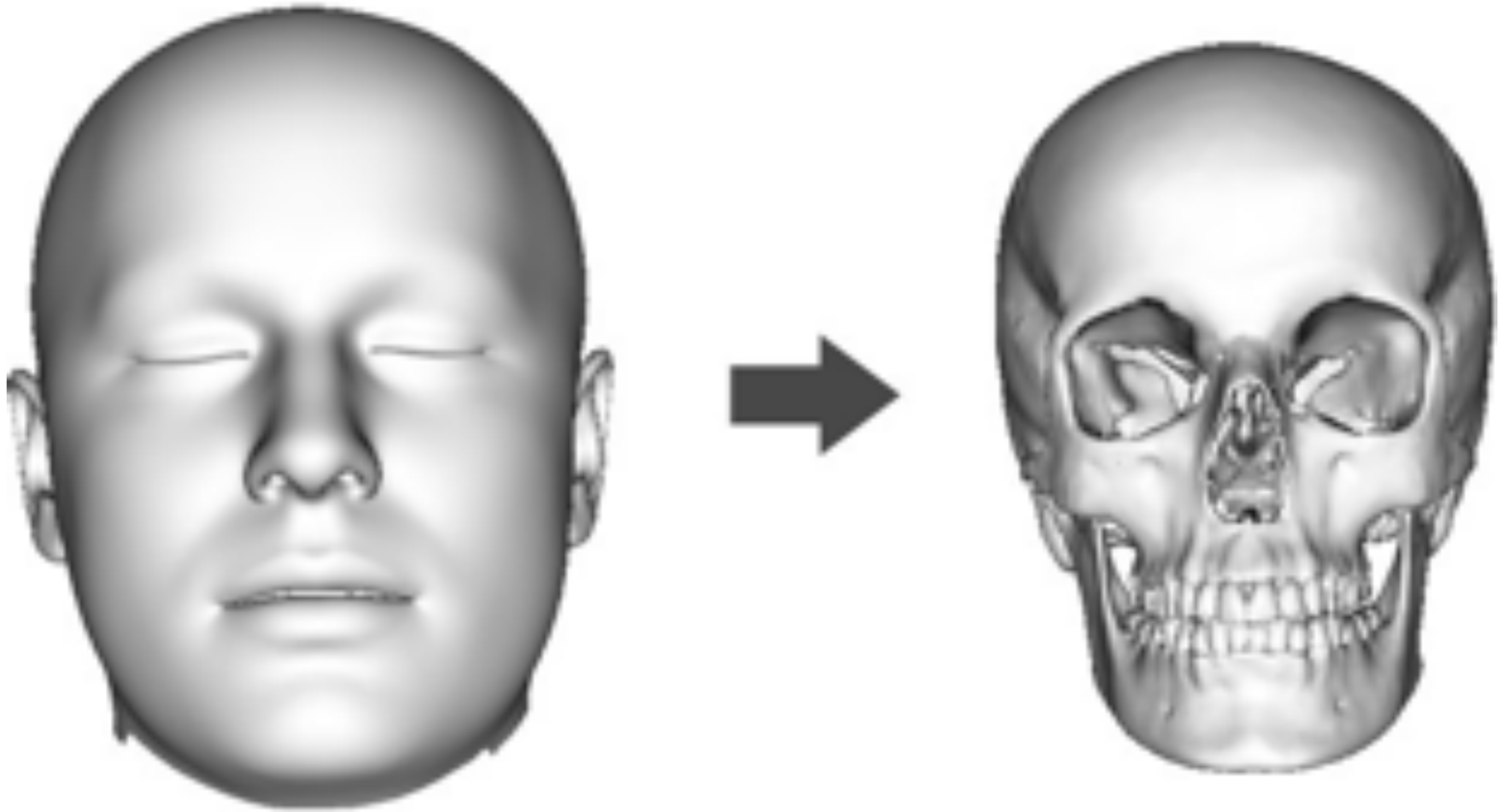
2. Construct and decompose data tensor
   - $\mathcal{D}_{i,j,k} = \mathbf{x}(\boldsymbol{\alpha}_j, \boldsymbol{\beta}_k)[i]$
   - $\mathcal{D} = \mathcal{M} \times_2 \mathbf{U}_{\text{skull}} \times_3 \mathbf{U}_{\text{fstt}}$

3. Build model tensor
   - $\mathcal{M} = \mathcal{D} \times_2 \mathbf{U}_{\text{skull}}^{\mathsf{T}} \times_3 \mathbf{U}_{\text{fstt}}^{\mathsf{T}}$
   - $\mathbf{x}(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \bar{\mathbf{x}} + \mathcal{M} \times_2 \boldsymbol{\alpha} \times_3 \boldsymbol{\beta}$



L. De Lathauwer, "Signal Processing Based on Multilinear Algebra", 1997

Infer Skull from Skin



Simulating Weight Changes for Face Scans

scan

skinny

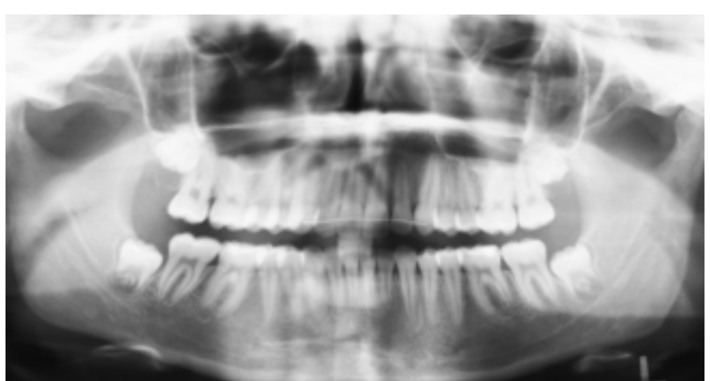
fat

Multilinear Model and Deep Learning

Craniofacial Reconstruction from a single X-ray

- Determine the 3D structure of the (craniofacial) skull from a single x-ray

Lateral cephalometric radiograph



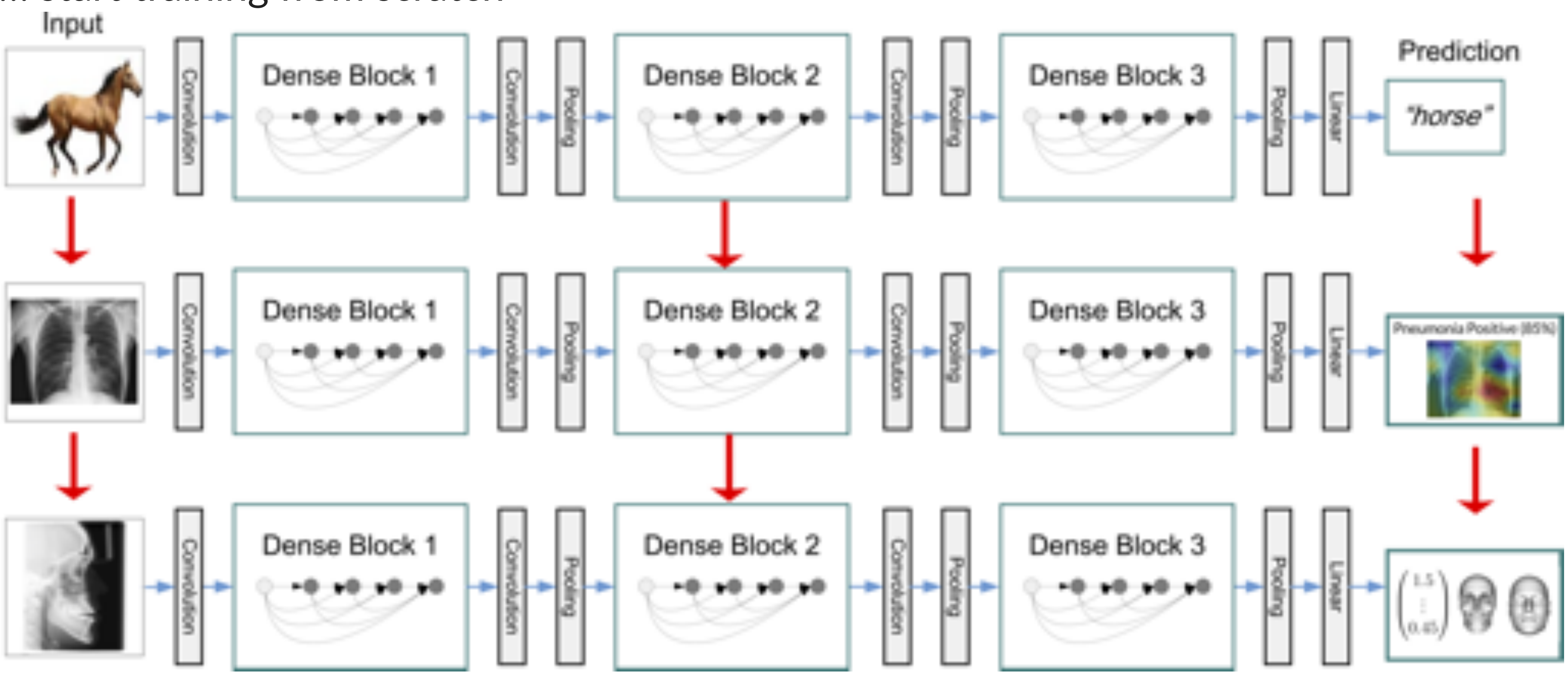
Panoramic radiograph

3D reconstruction of the skull

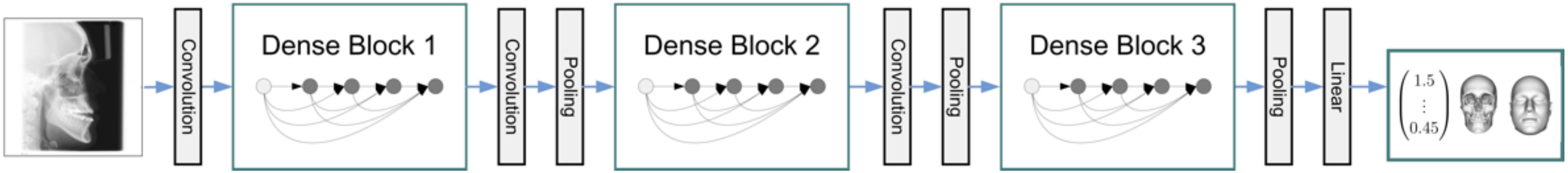
Craniofacial Reconstruction from a single X-ray

Craniofacial Reconstruction from a single X-ray

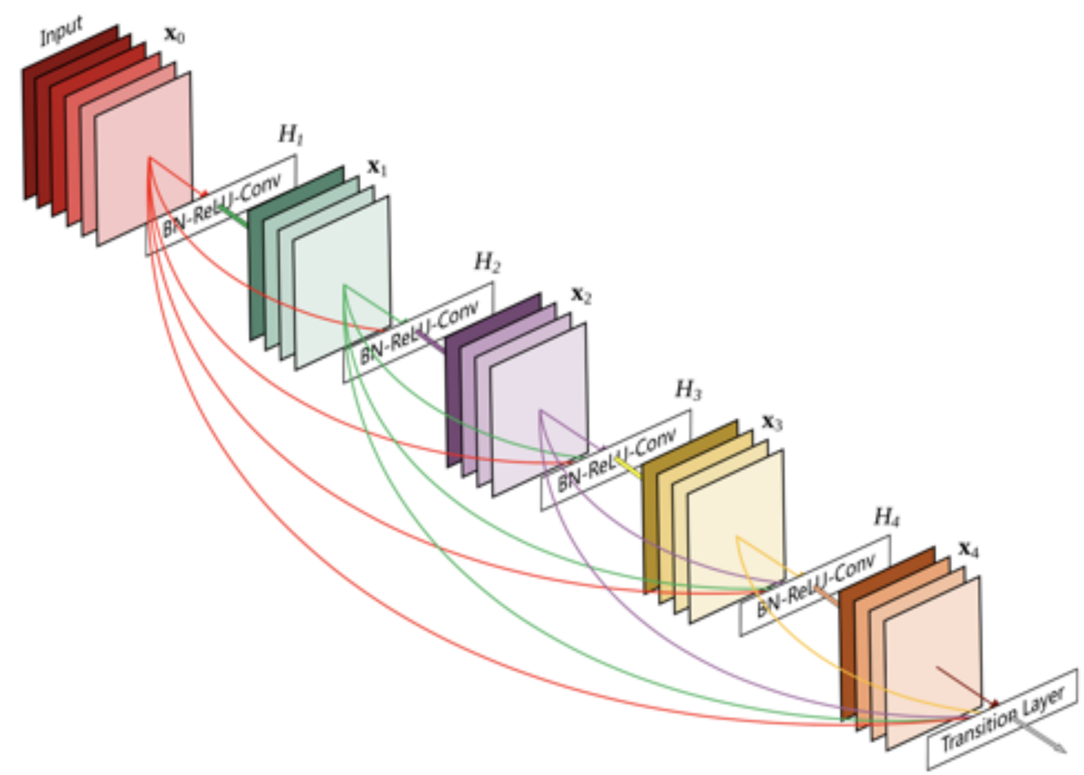
- Best results using *DensNet*, with same quality but faster convergence if we …
    - … start with weights from *CheXNet* (*DenseNet* trained with x-rays of the chest)
    - … start with weights from *DenseNet*, trained on *ImageNet*
    - … start training from scratch



Densely Connected Convolutional Networks



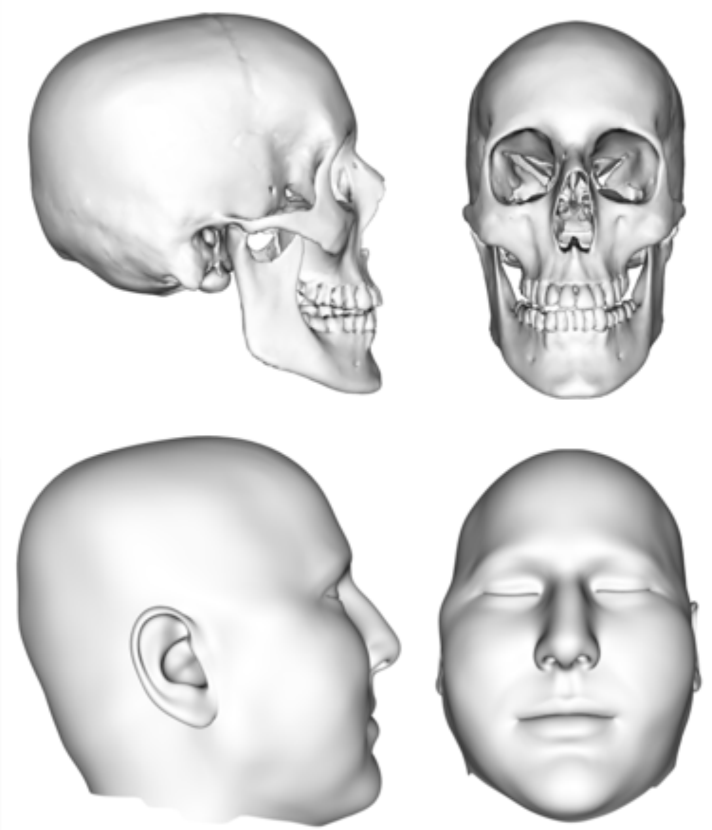
- Advantages of dense blocks
  - Alleviate the vanishing-gradient problem
  - Strengthen feature propagation
  - Encourage feature reuse
  - Reduce number of parameters



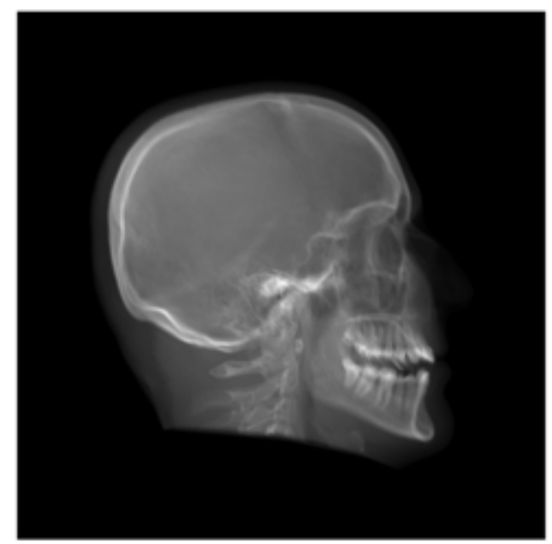
5-layer dense block with growth rate of $k = 4$

G. Huang, Z. Liu, L. v. d. Maaten, and K. Q. Weinberger, *Densely connected convolutional networks*, CVPR 2017

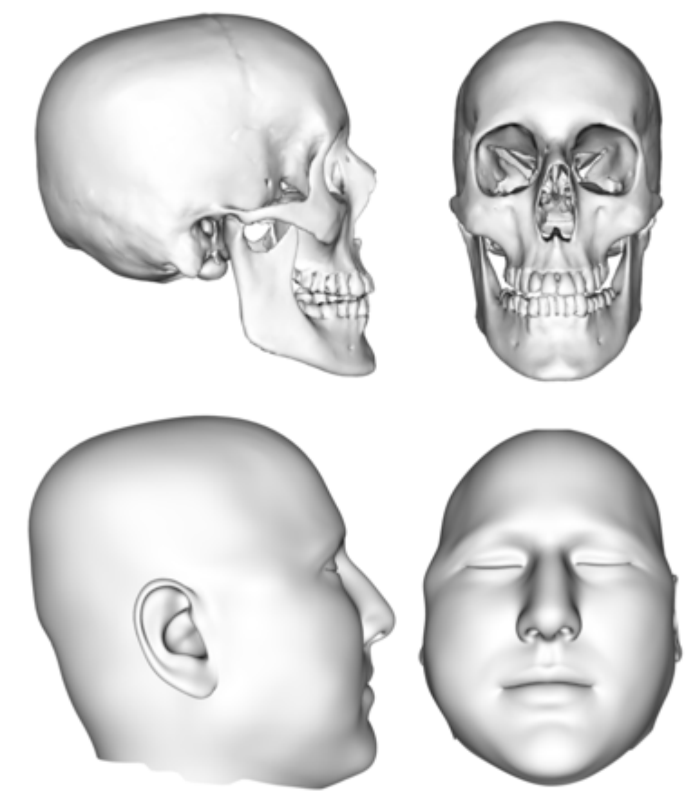
Training and Testing



Test Dataset

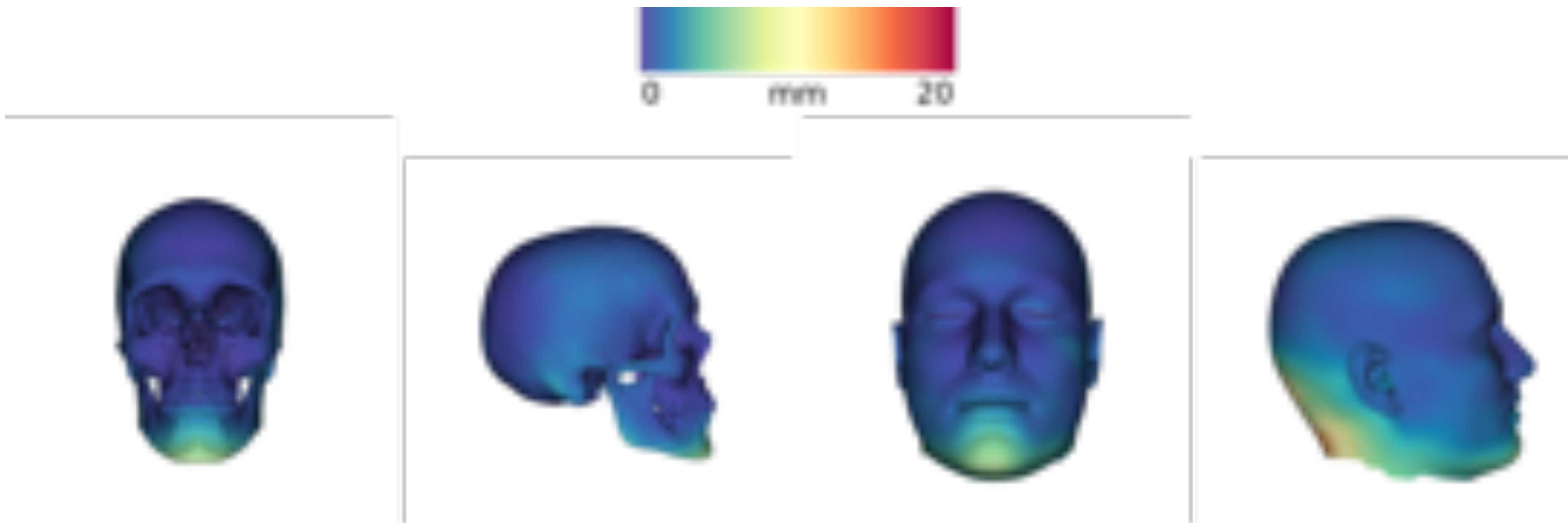


Artificial Lateral Radiographs (Network Input)



Evaluated Network Output

Visual Comparison of Reconstruction Results



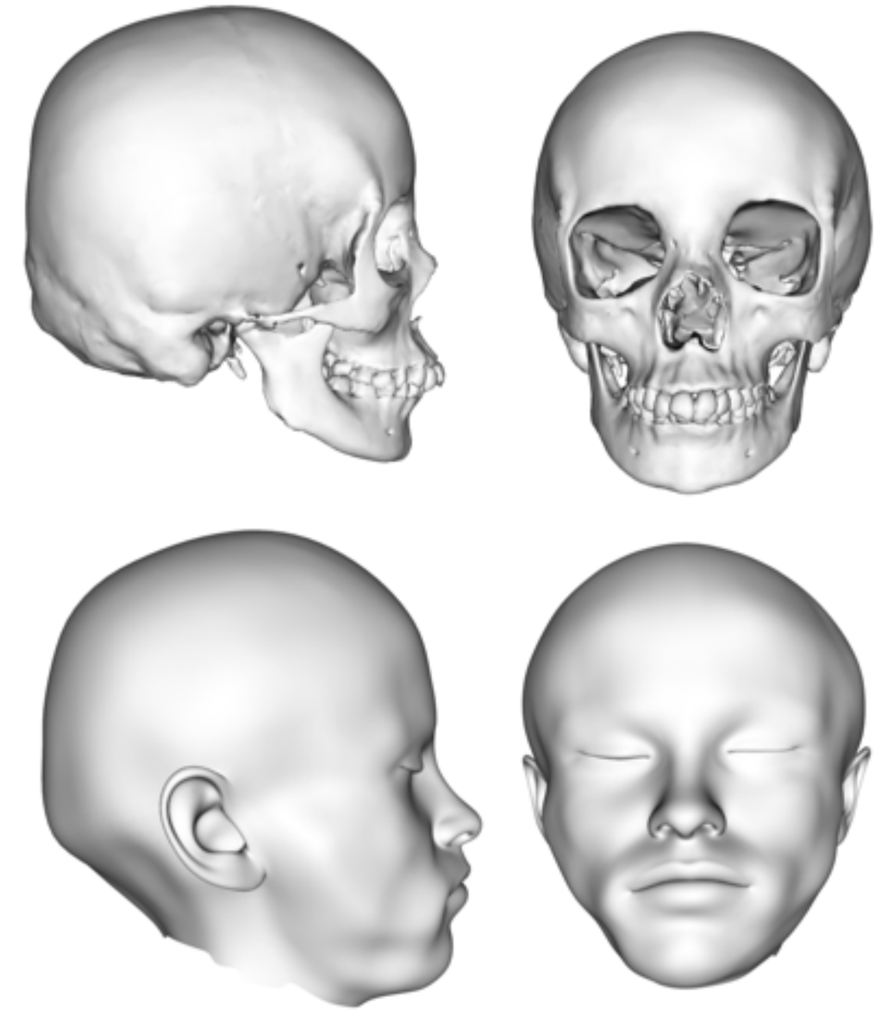
(a) CNN from Scratch

(b) Transfer Learning

A “Real World” Example

"Real World" Input

Preprocessed Input



Network Output

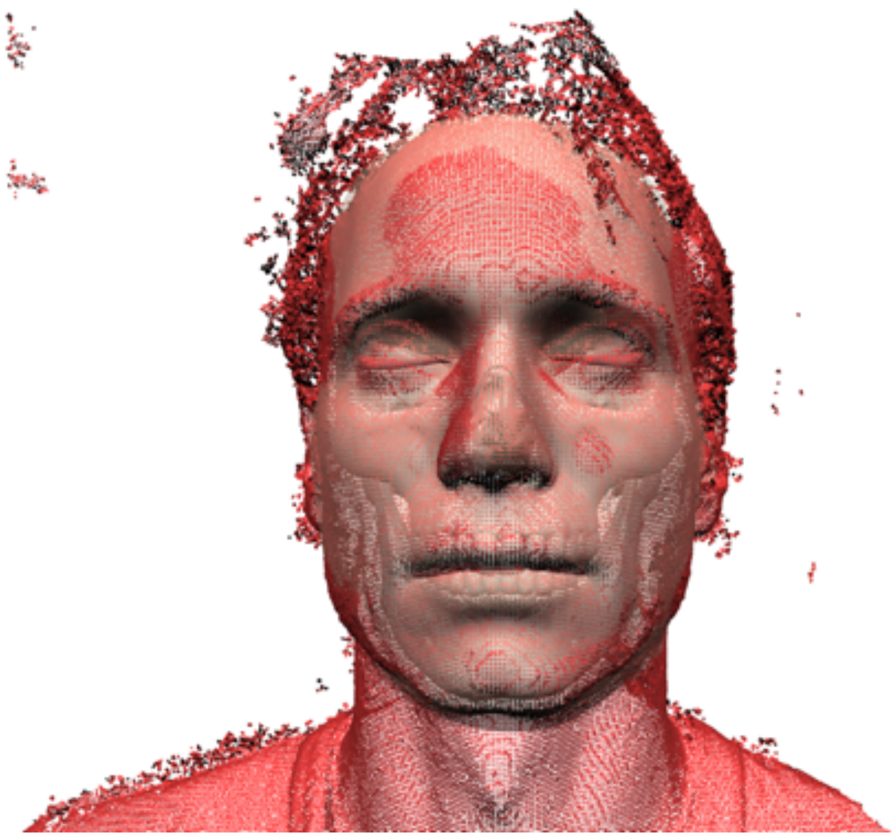
A “Real World” Example

"Real World" Input

Preprocessed Input

Artificial Generated Radiograph

Thank you



Prof. Dr. Ulrich Schwanecke

Computer Graphics and Vision

Faculty of Design–Computer Science–Media

RheinMain University of Applied Sciences

ulrich.schwanecke@hs-rm.de

cvmr.info