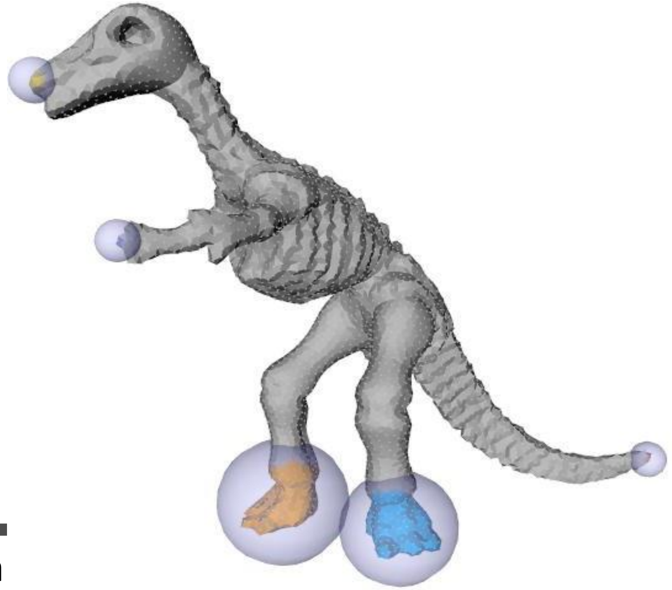
3D Scanning & Motion Capture

Parametric Face Models

Prof. Dr. Nießner

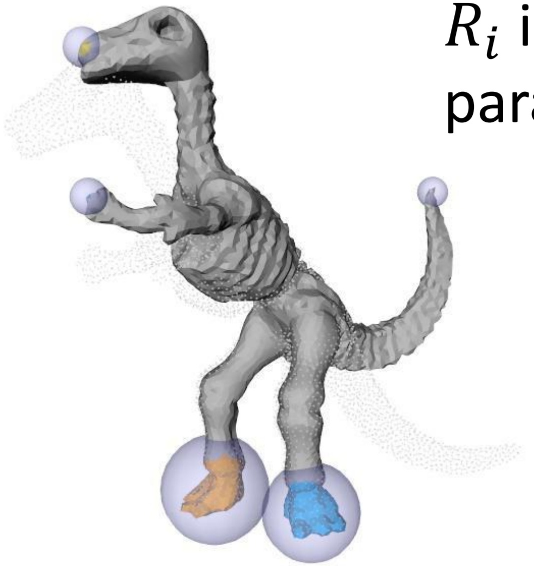
As-Rigid-as-Possible Deformation

- One rotation matrix per fan; sum up deviations from rigidity
- $E_{ARAP}(\boldsymbol{R}, \boldsymbol{V}) = \sum_i \sum_{j \in \mathcal{N}(i)} \left|\left|(v_i - v_j) - R_i(v_i' - v_j')\right|\right|_2^2$
- $E_{fit}(\boldsymbol{V}) = \sum_{i \in C} \left|\left|v_i{}' - v_i\right|\right|_2^2$



Initial mesh V'

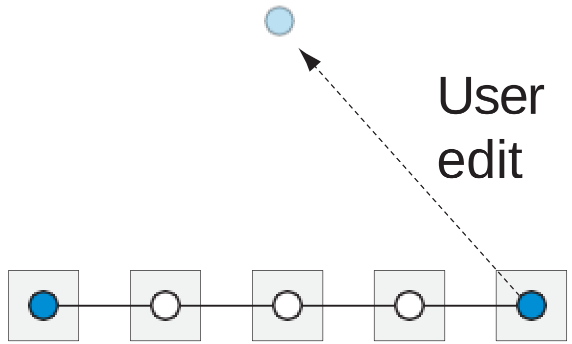
[Sorkine et al. 08]

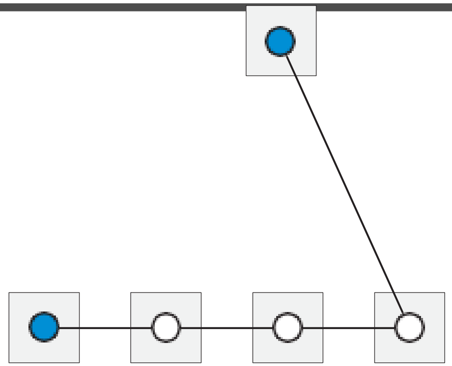


Deformed mesh V

R_i is rotation matrix and parametrized by Euler angles α, β, γ

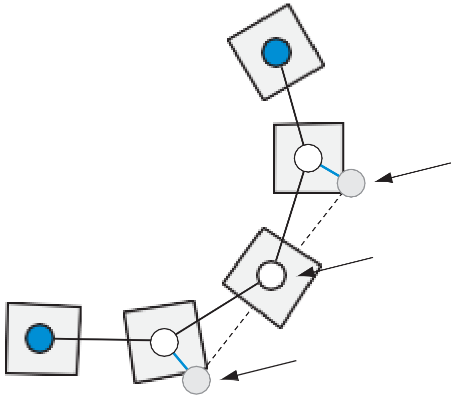
Embedded Deformation



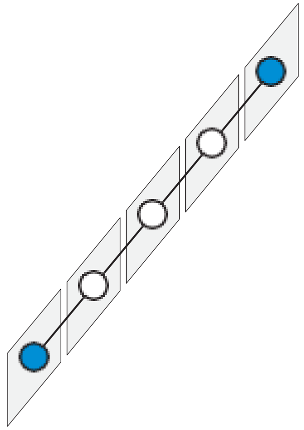


$\mathrm{E}_{\mathrm{con}}$

$$E_{ED}(\boldsymbol{M},\boldsymbol{V}) = \sum_{i} \sum_{j \in \mathcal{N}(i)} \left\| \left(v_i - v_j\right) - M_i\left(v_i' - v_j'\right) \right\|_2^2$$



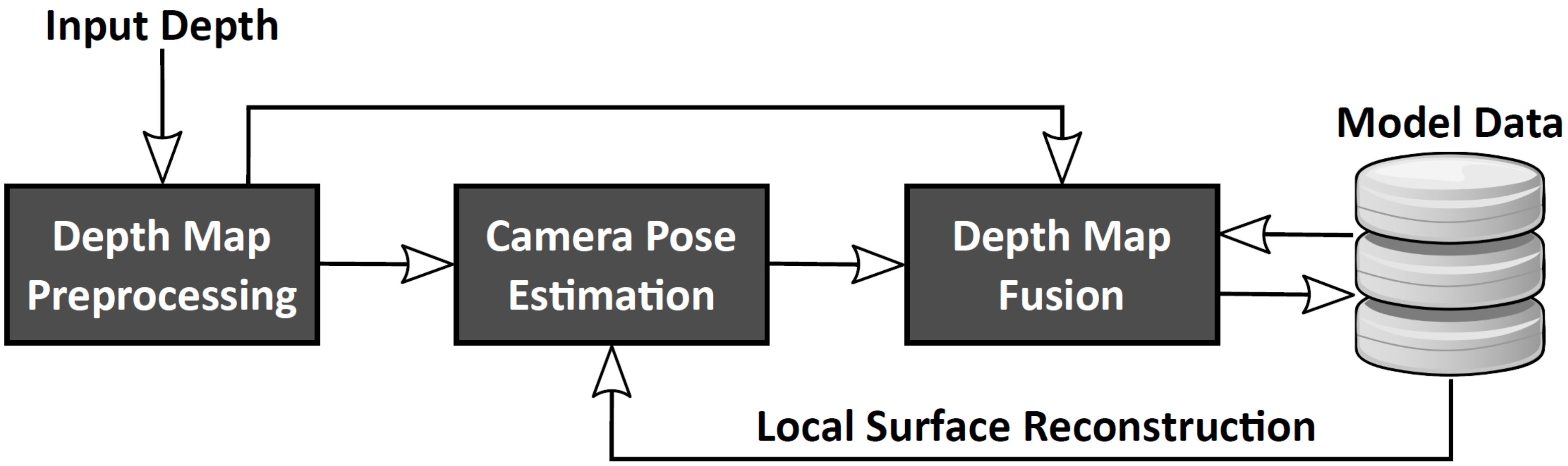
$\mathrm{E}_{\mathrm{con}} + \mathrm{E}_{\mathrm{reg}} + \mathrm{E}_{\mathrm{rot}}$

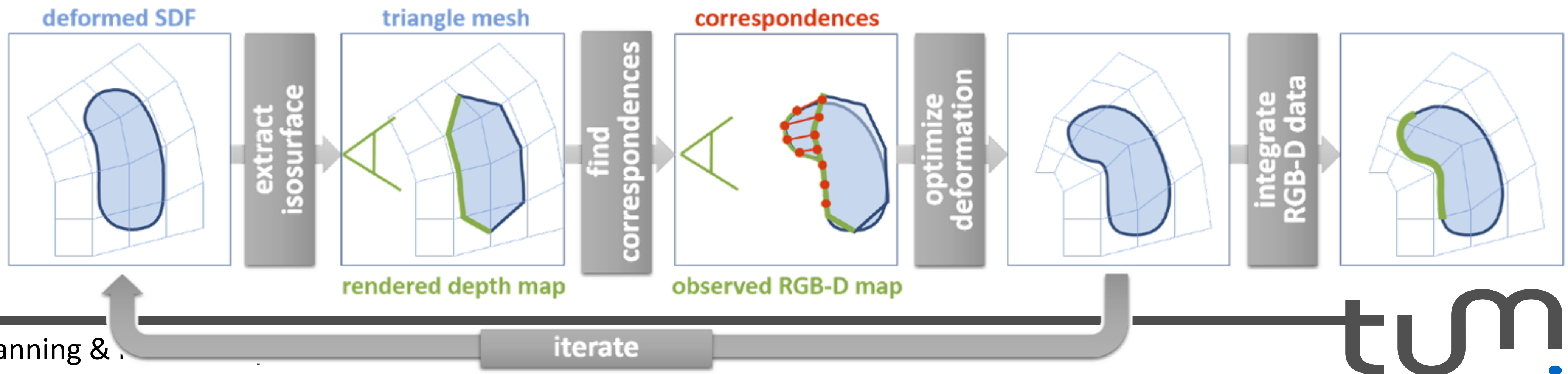


$\mathrm{E}_{\mathrm{con}} + \mathrm{E}_{\mathrm{reg}}$

[Sumner et al. 07]

Rigid vs Non-Rigid Reconstruction





Today: Parametric Face Models

Dynamic 3D Capture

Need more regularization with fewer DoF to make it practical!

Domain-Specific

- Human body
- Hands
- Faces

Free-form Reconstruction

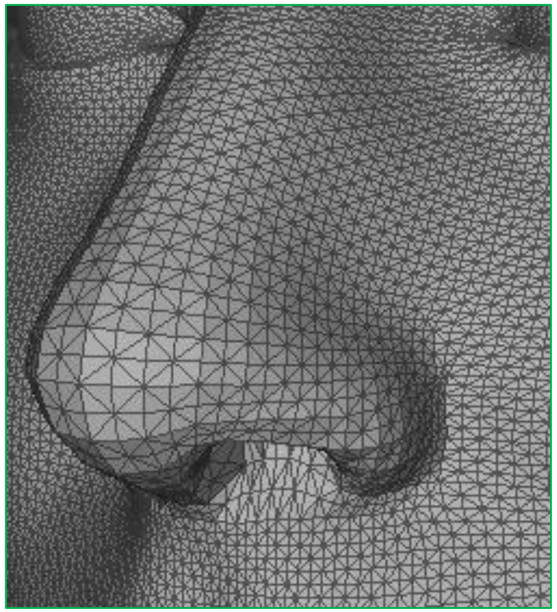
- Joint reconstruction and tracking

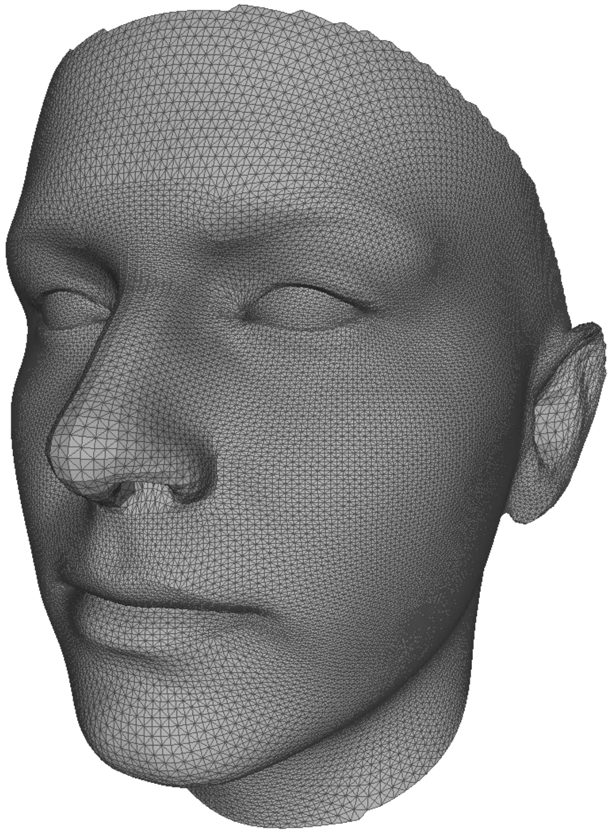
Free-form Tracking

- Offline template reconstruction
- Online template tracking

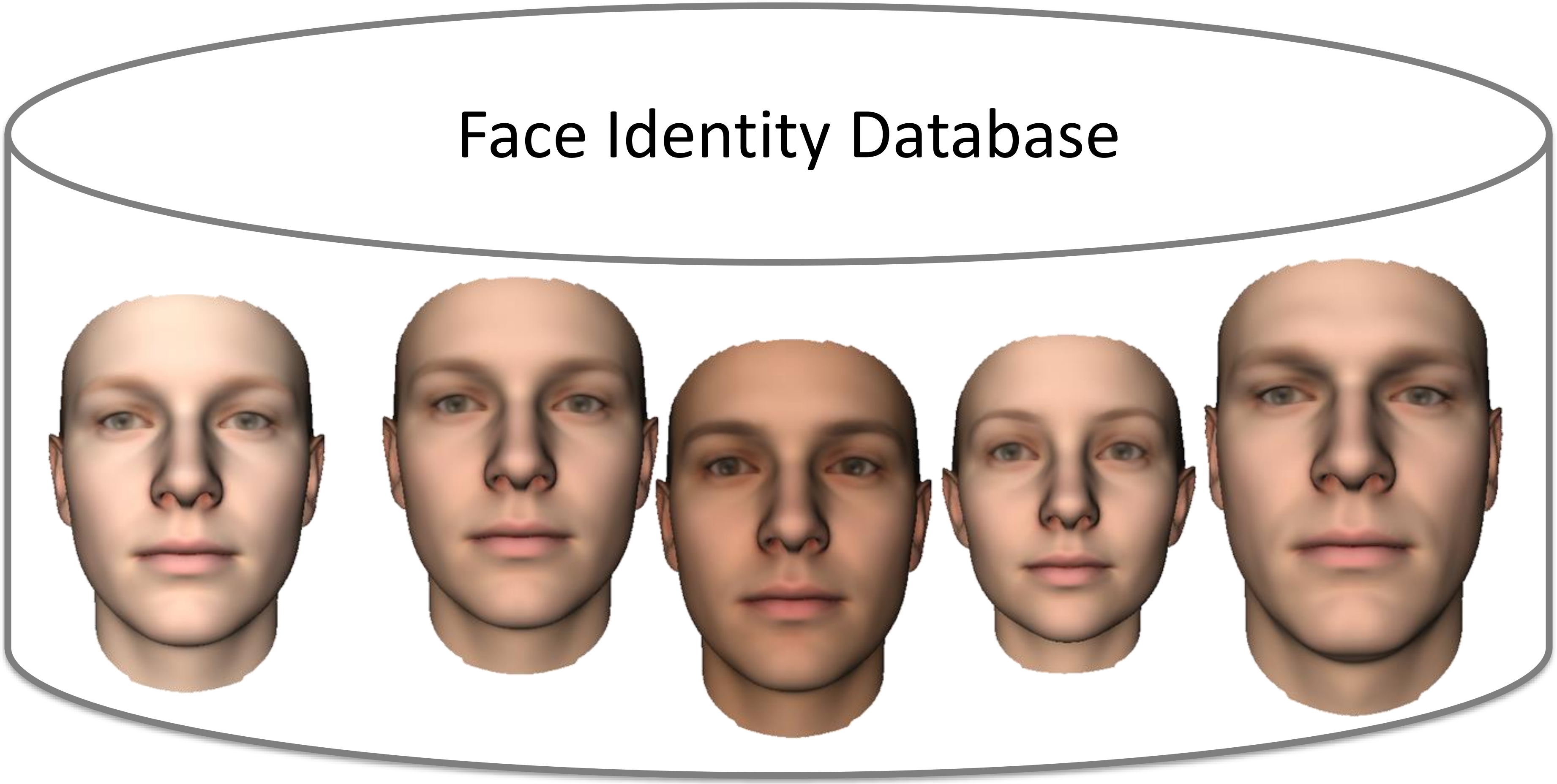
Parametric Models -> Dimensionality Reduction

- Small number of parameters change larger number of vertices
  - E.g., basis functions + skinning
- For Faces, Hands, Bodies
- Simplest one:
  - Rigid 6-DOF poses between frames
    - 6 parameters transform entire frames
    - Nobody would really call it this way





Parametric Face Model: Shape Identity

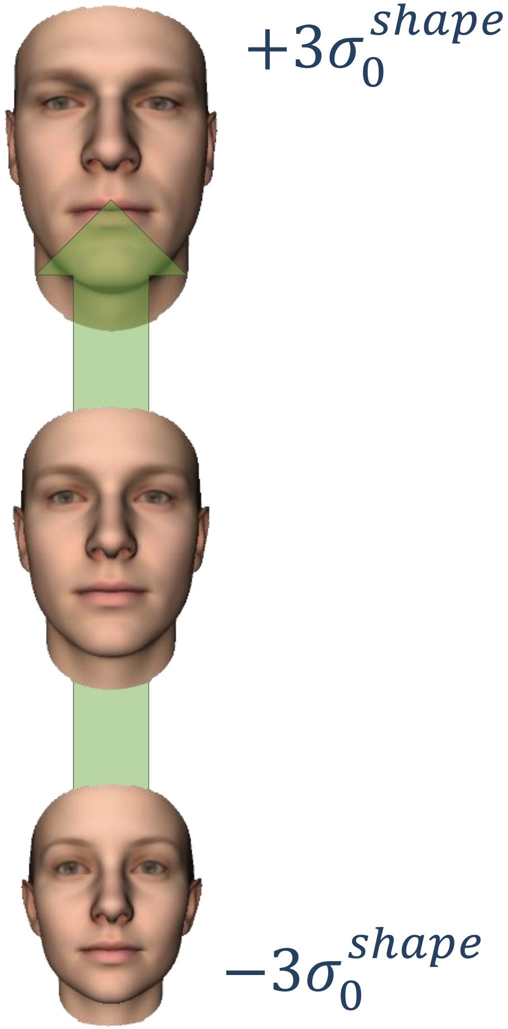


[Blanz and Vetter 99]: BlendShapes

Parametric Face Model: Shape Identity

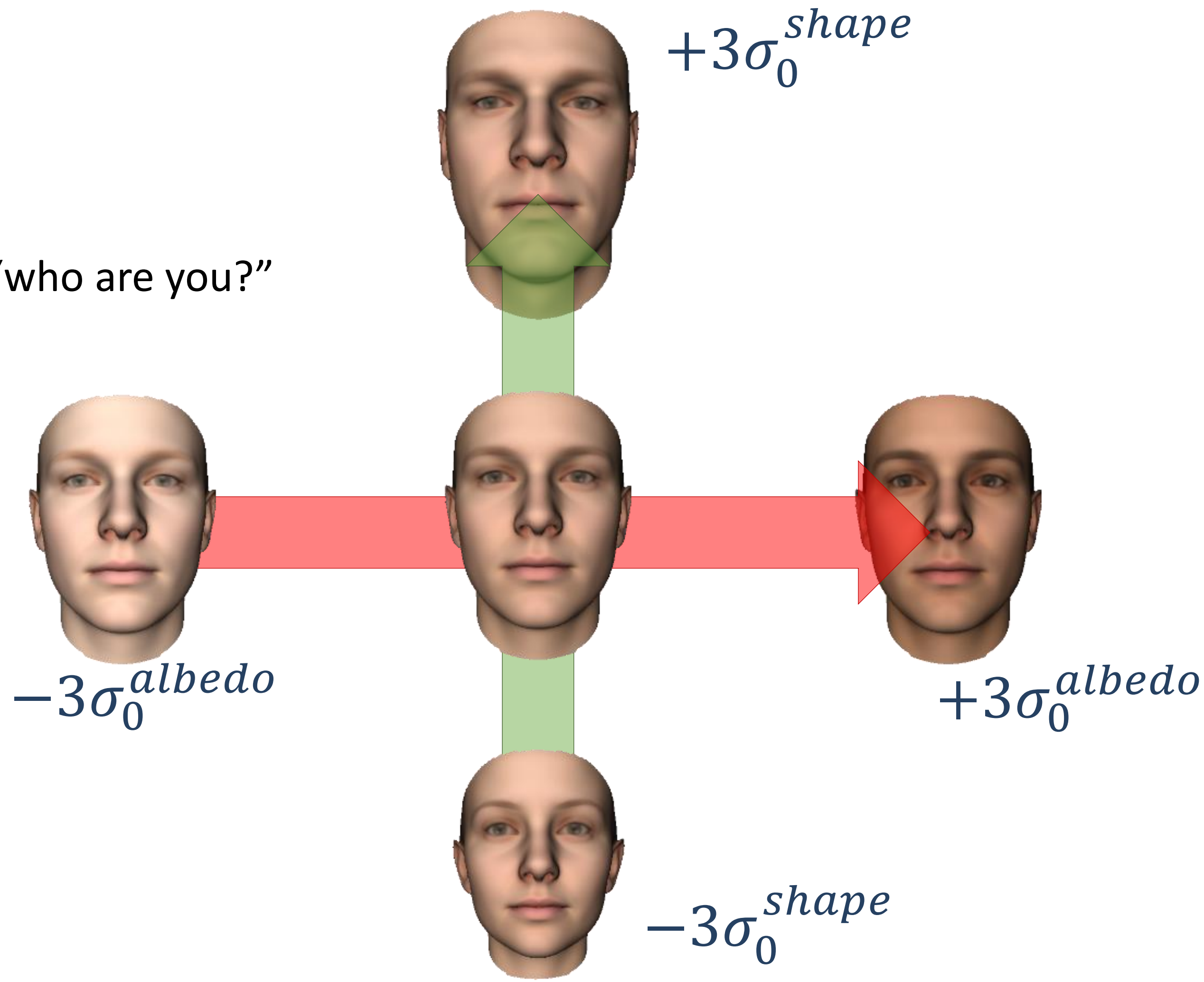
Average Face

Parametric Face Model: Shape Identity

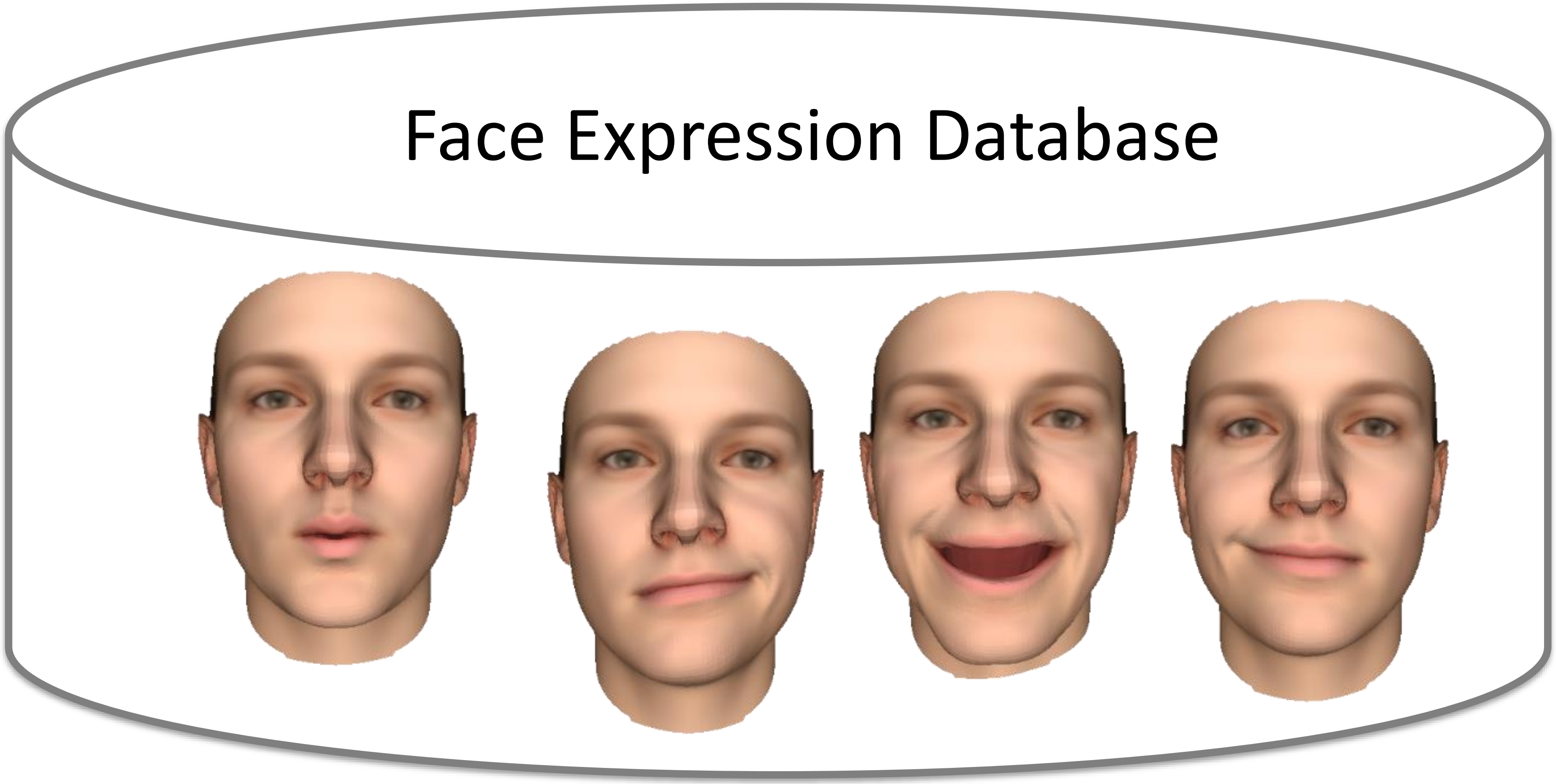


Parametric Face Model: Shape Identity

Shape Identity defines "who are you?"



Parametric Face Model

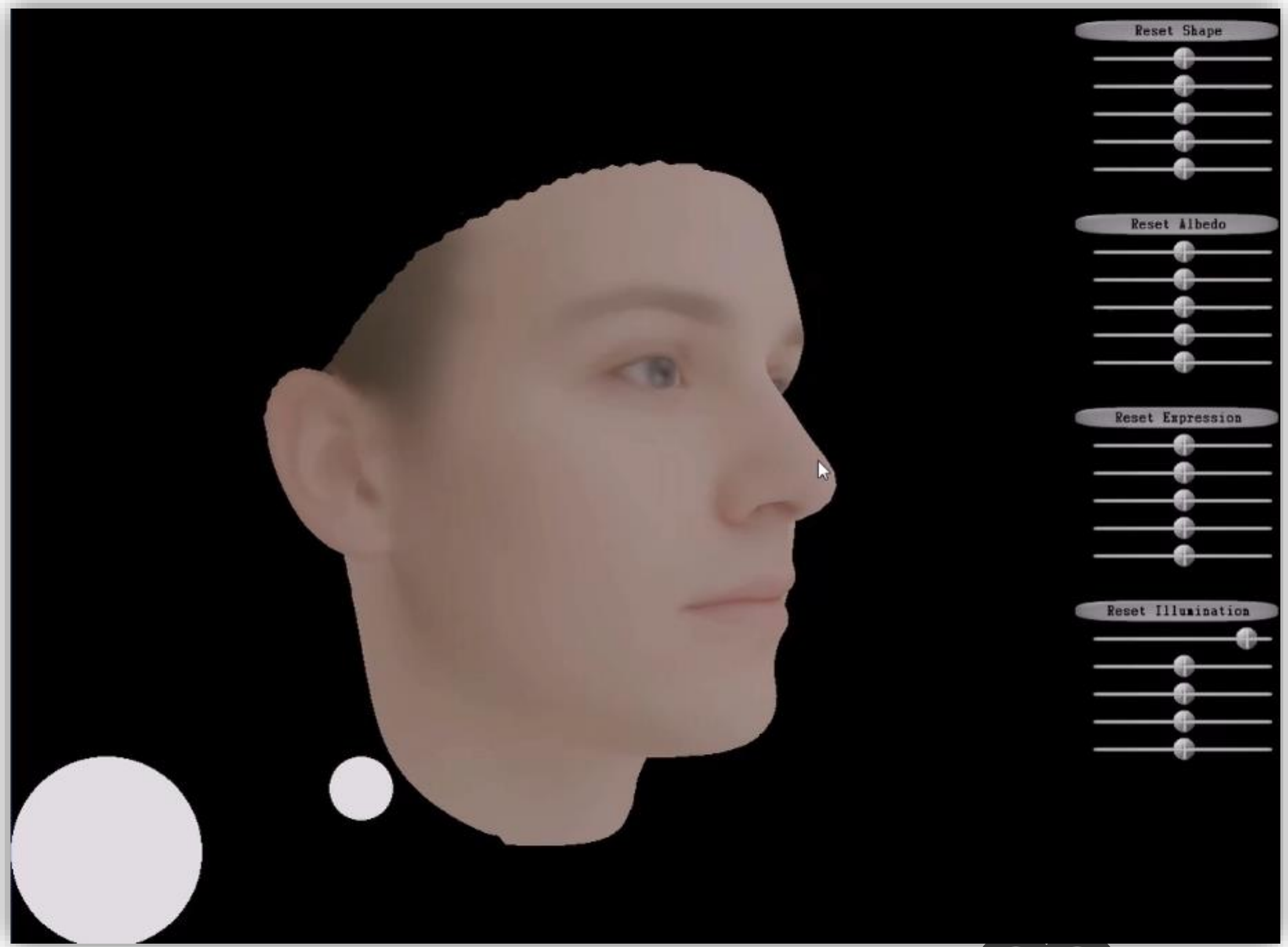


Parametric Face Model

Rigid Pose

$$P = \begin{pmatrix} \Phi \\ \\ \\ \\ \end{pmatrix}$$

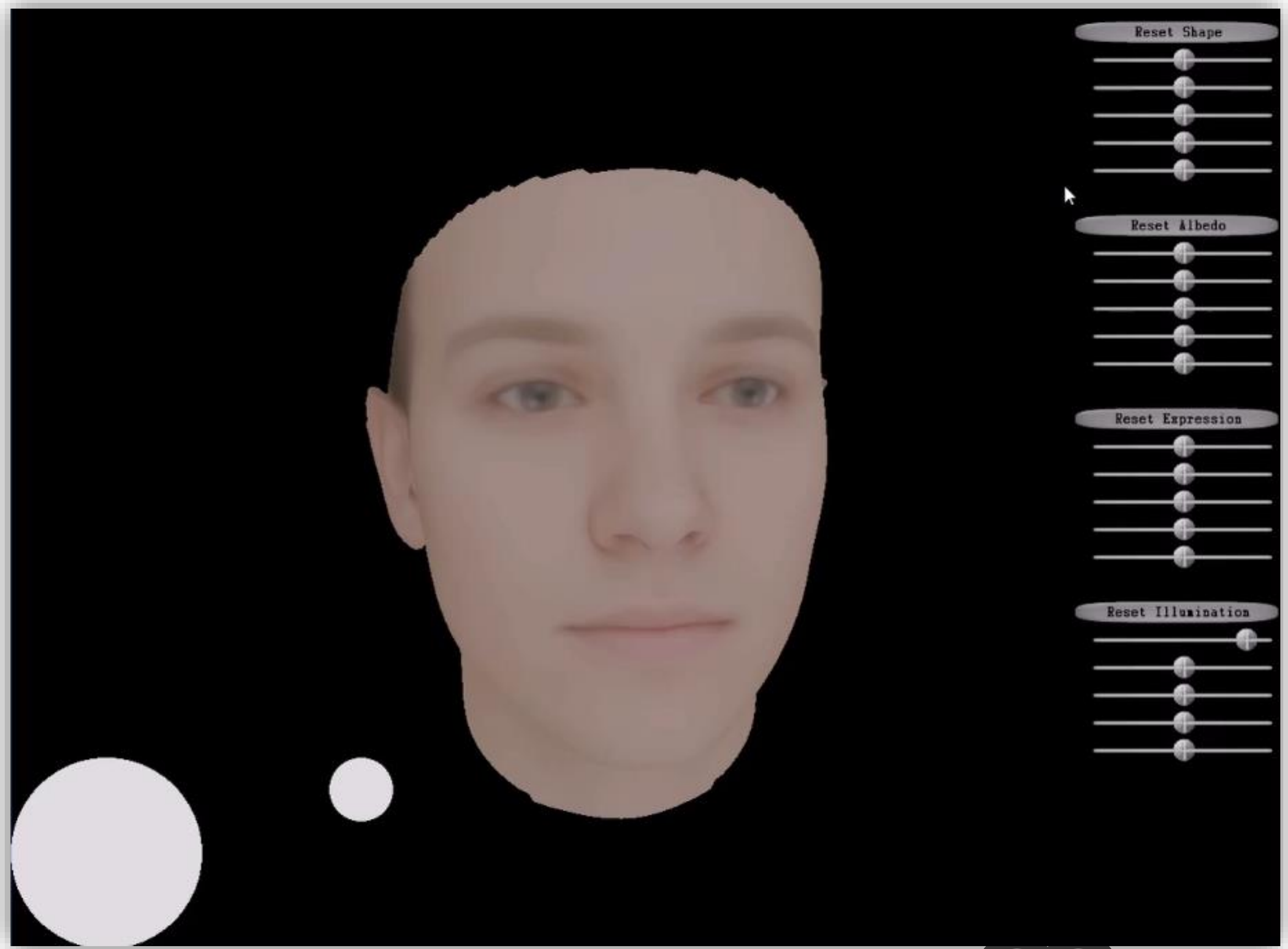
$|P| = 6$



Parametric Face Model

Shape Identity

$$\boldsymbol{P} = \begin{pmatrix} \Phi \\ \alpha \\ \\ \\ \\ \end{pmatrix}$$



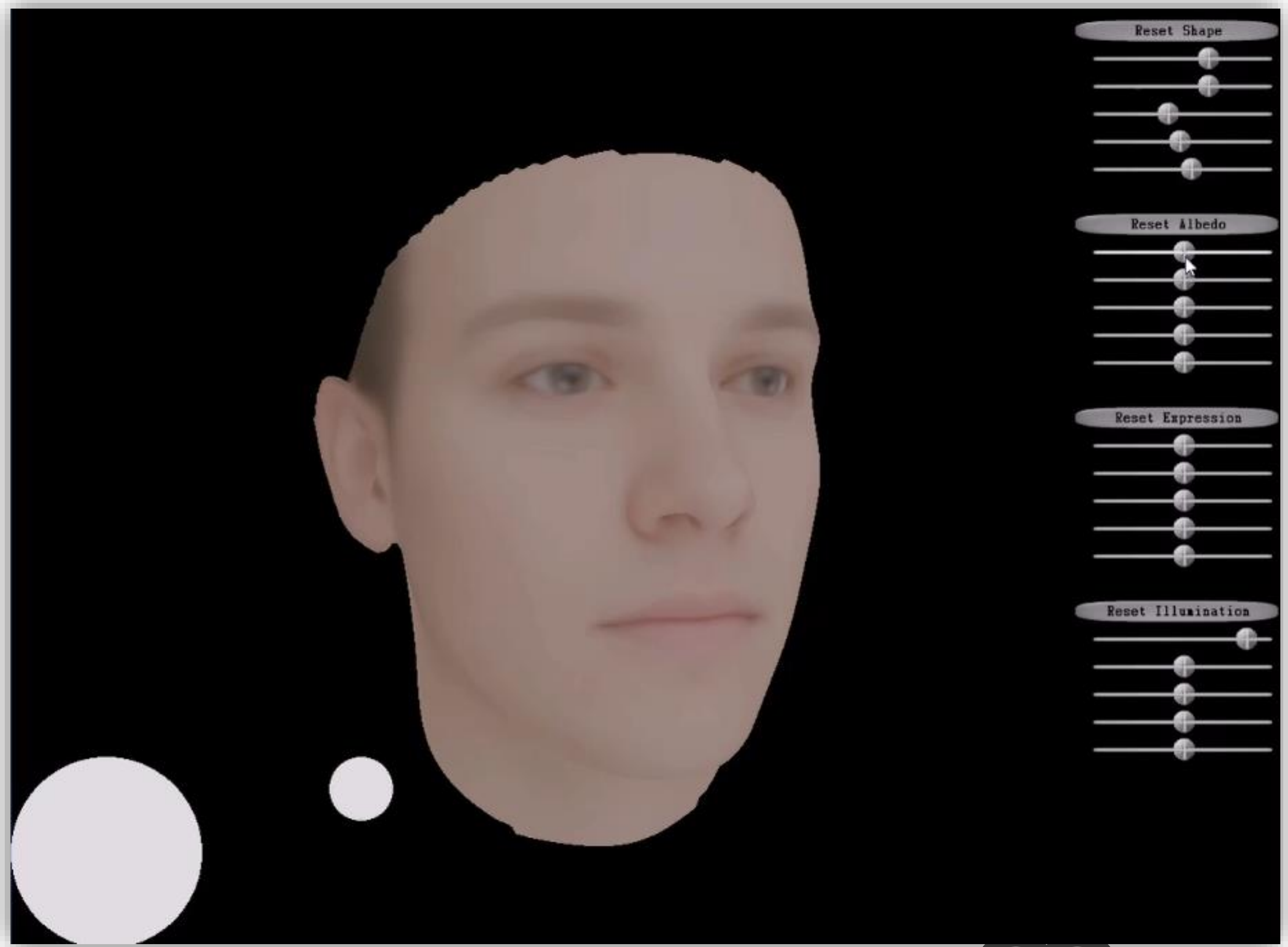
$|\boldsymbol{P}| = 6 + 80$

Parametric Face Model

Material / Reflection

$$\boldsymbol{P} = \begin{pmatrix} \Phi \\ \alpha \\ \beta \\ \\ \\ \\ \end{pmatrix}$$

$|\boldsymbol{P}| = 6 + 80 + 80$

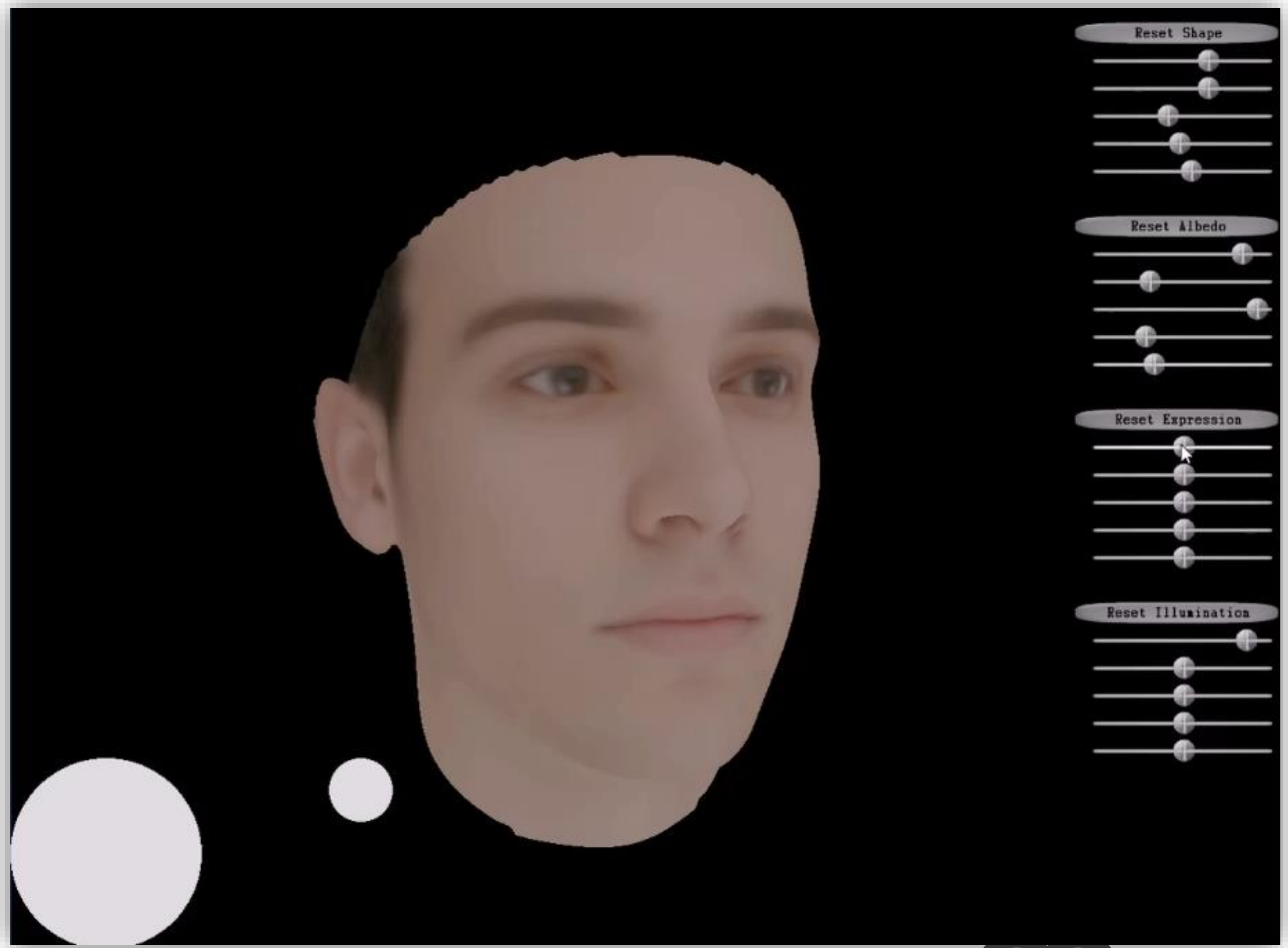


Parametric Face Model

Expression Parameters

$$\boldsymbol{P} = \begin{pmatrix} \Phi \\ \alpha \\ \beta \\ \delta \end{pmatrix}$$

$|\boldsymbol{P}| = 6 + 80 + 80 + 76$

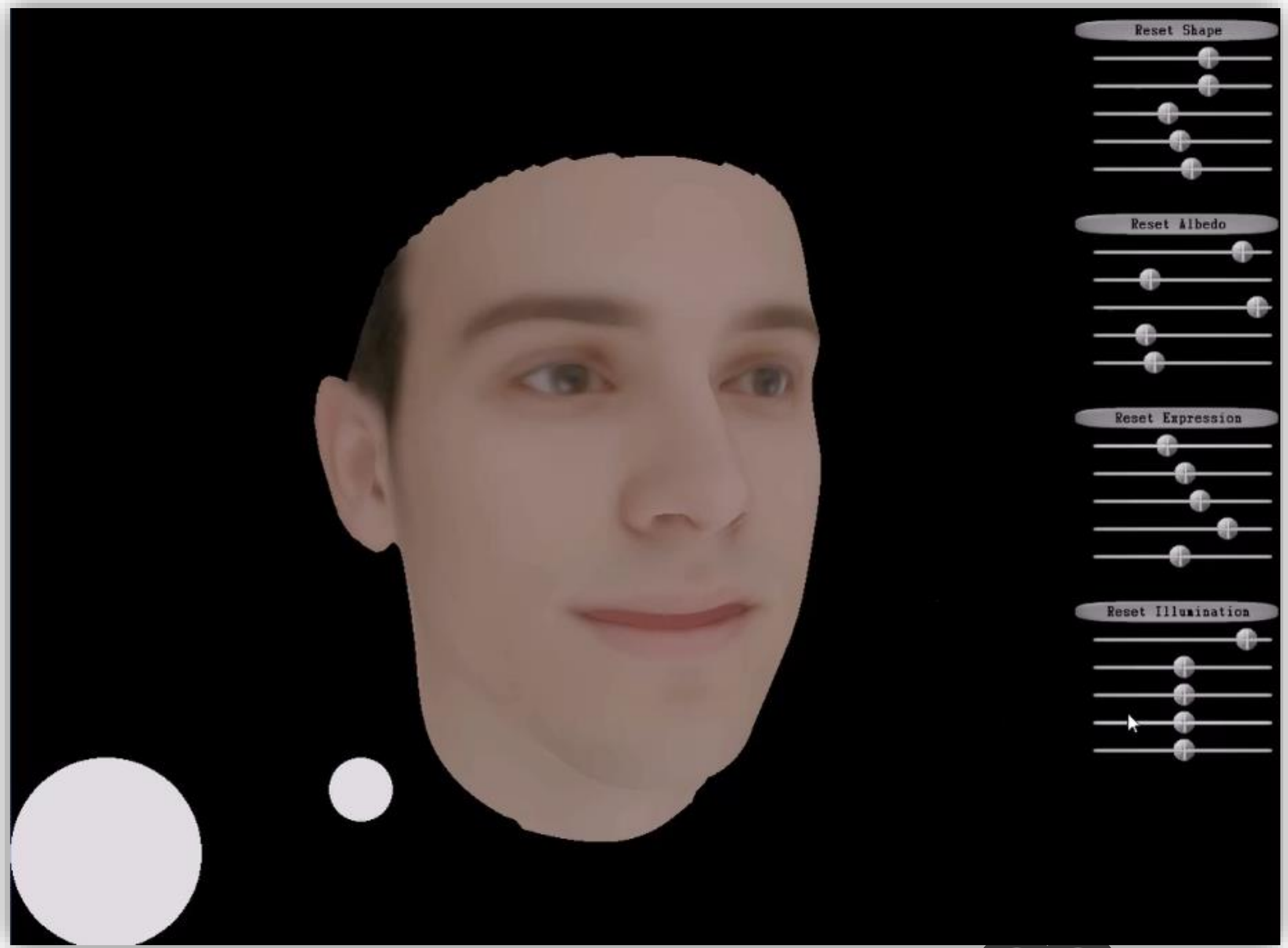


Parametric Face Model

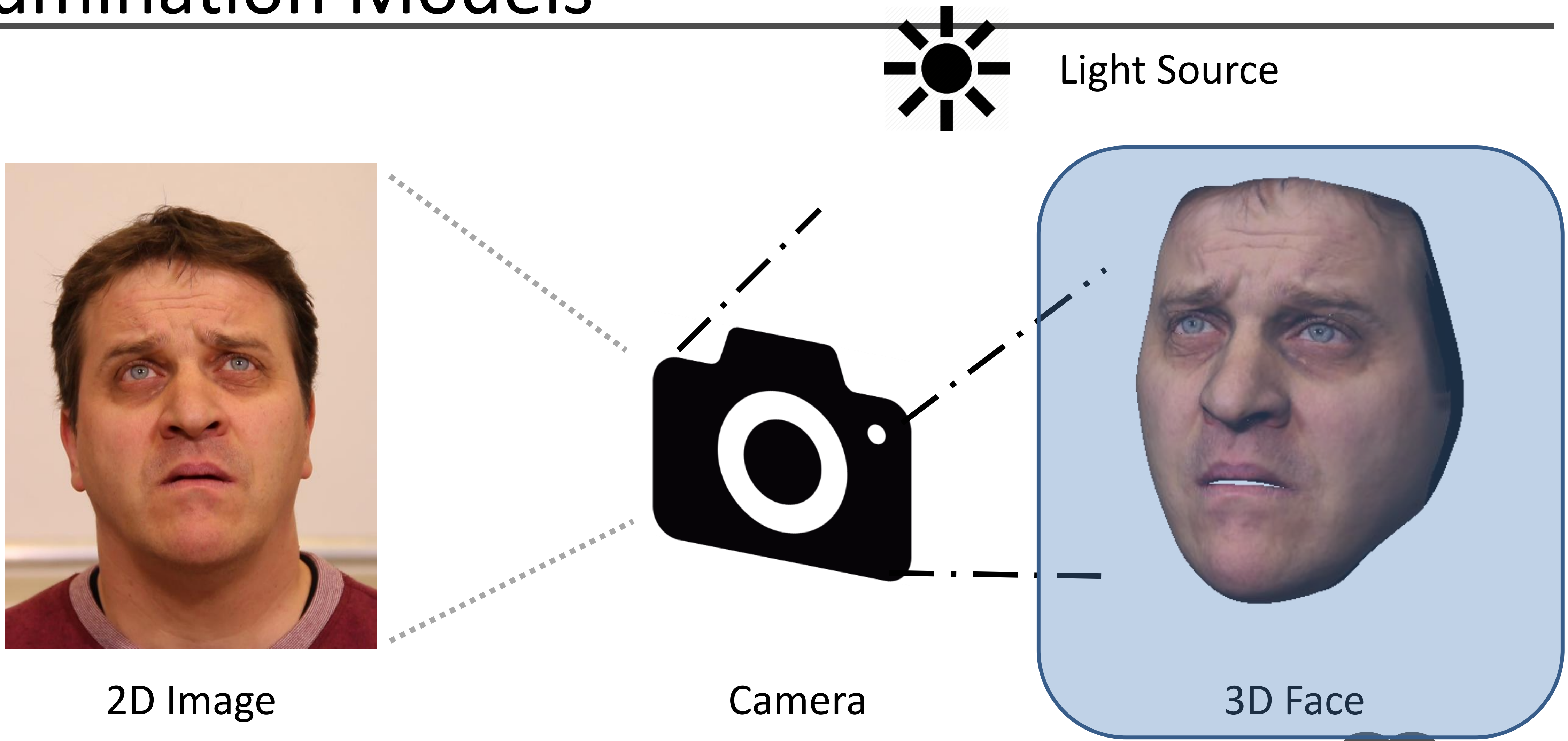
Lighting Parameters

$$\boldsymbol{P} = \begin{pmatrix} \Phi \\ \alpha \\ \beta \\ \delta \\ \gamma \end{pmatrix}$$

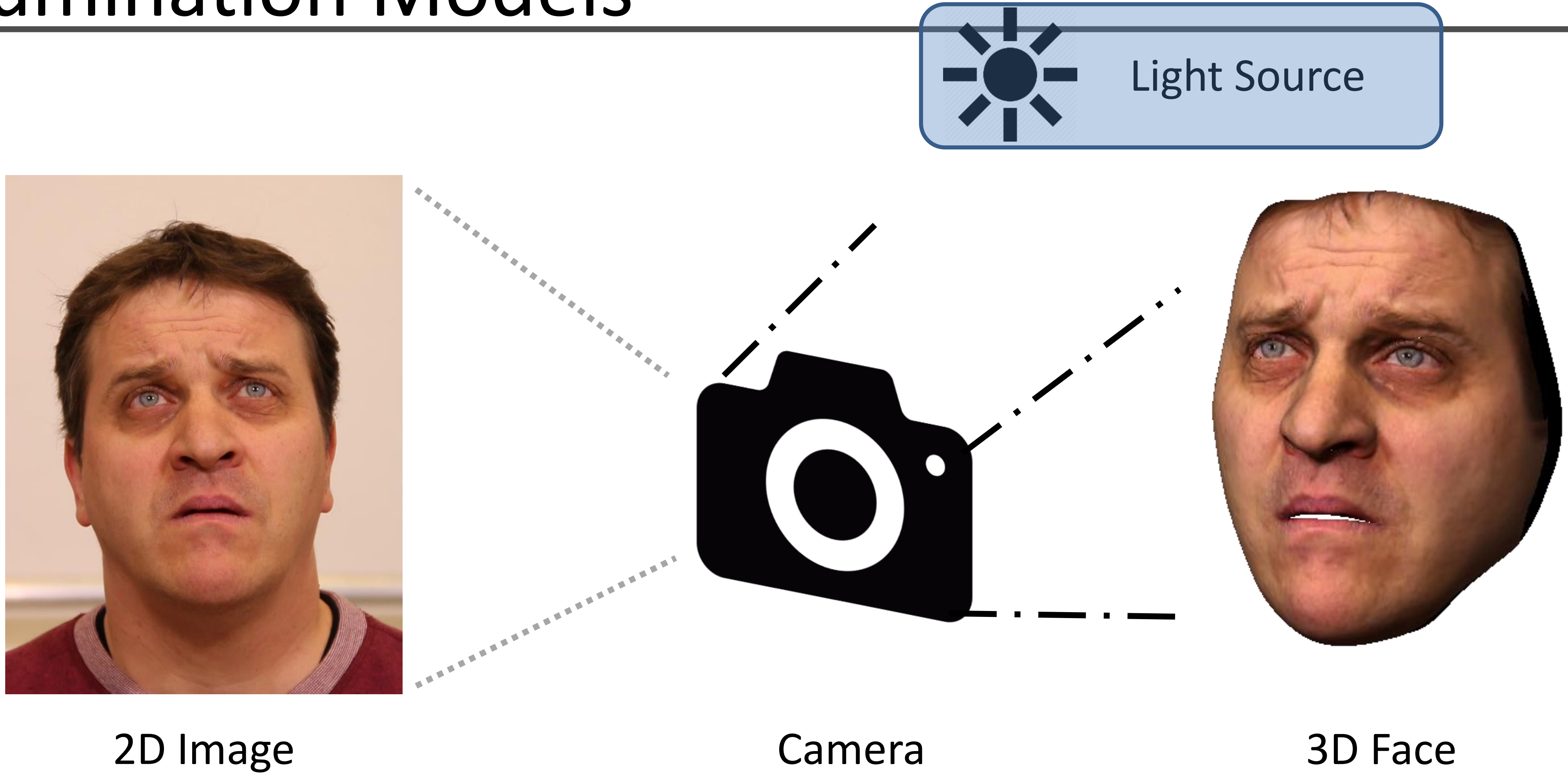
$|\boldsymbol{P}| = 6 + 80 + 80 + 76 + 27$



Illumination Models



Illumination Models



Illumination Models

- Lighting-face interaction is complex (e.g., highlights, scattering)
  - Advanced light transport models intractable for face reconstruction

- Simplified yet reasonable representations:
  - Environment map
  - Spherical harmonics

Illumination Models

- Environment maps [Klehm et al. '15]

Scene lighting

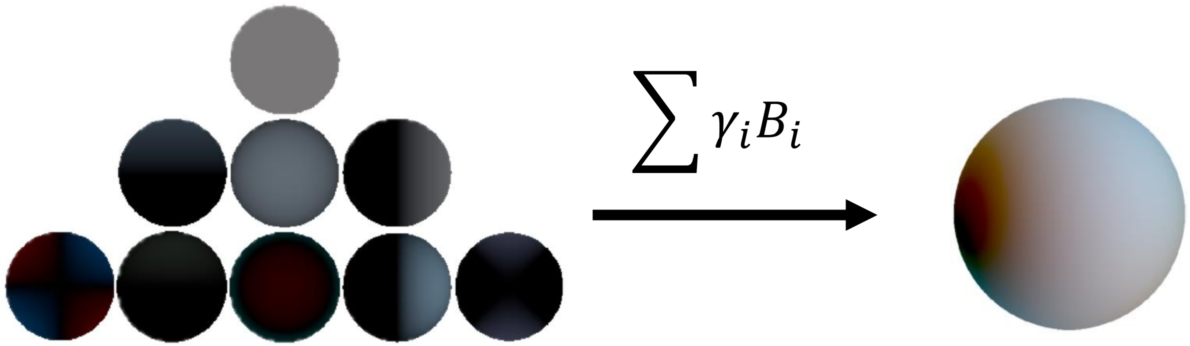
Cube map

Sphere map

- Assumptions:
  - Distant lighting
  - No self shadows, no scattering

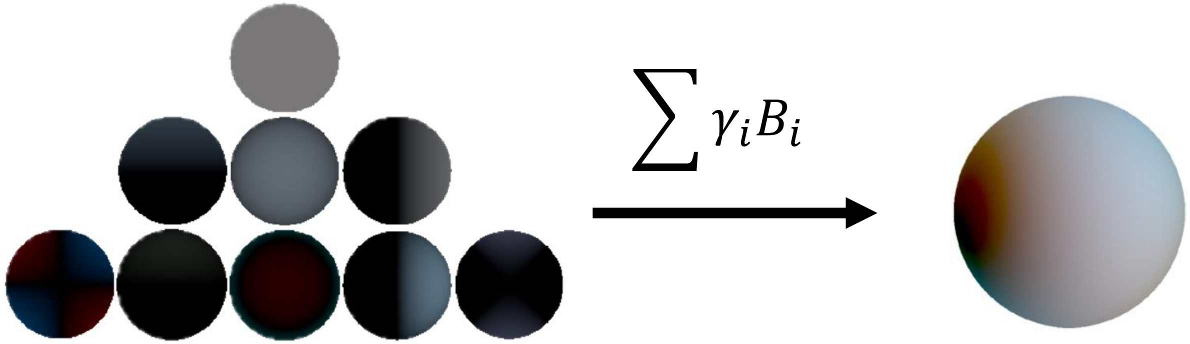
Illumination Models

- Spherical Harmonics [Mueller '66]
  - Orthogonal basis
  - Defined over a sphere



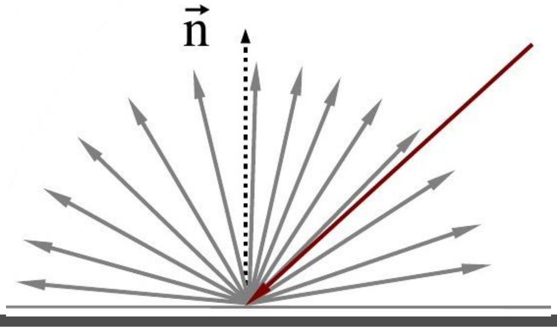
Illumination Models

- Spherical Harmonics [Mueller '66]
  - Orthogonal basis
  - Defined over a sphere



SH basis $[B_1 | \dots | B_9]^{\mathrm{T}}$

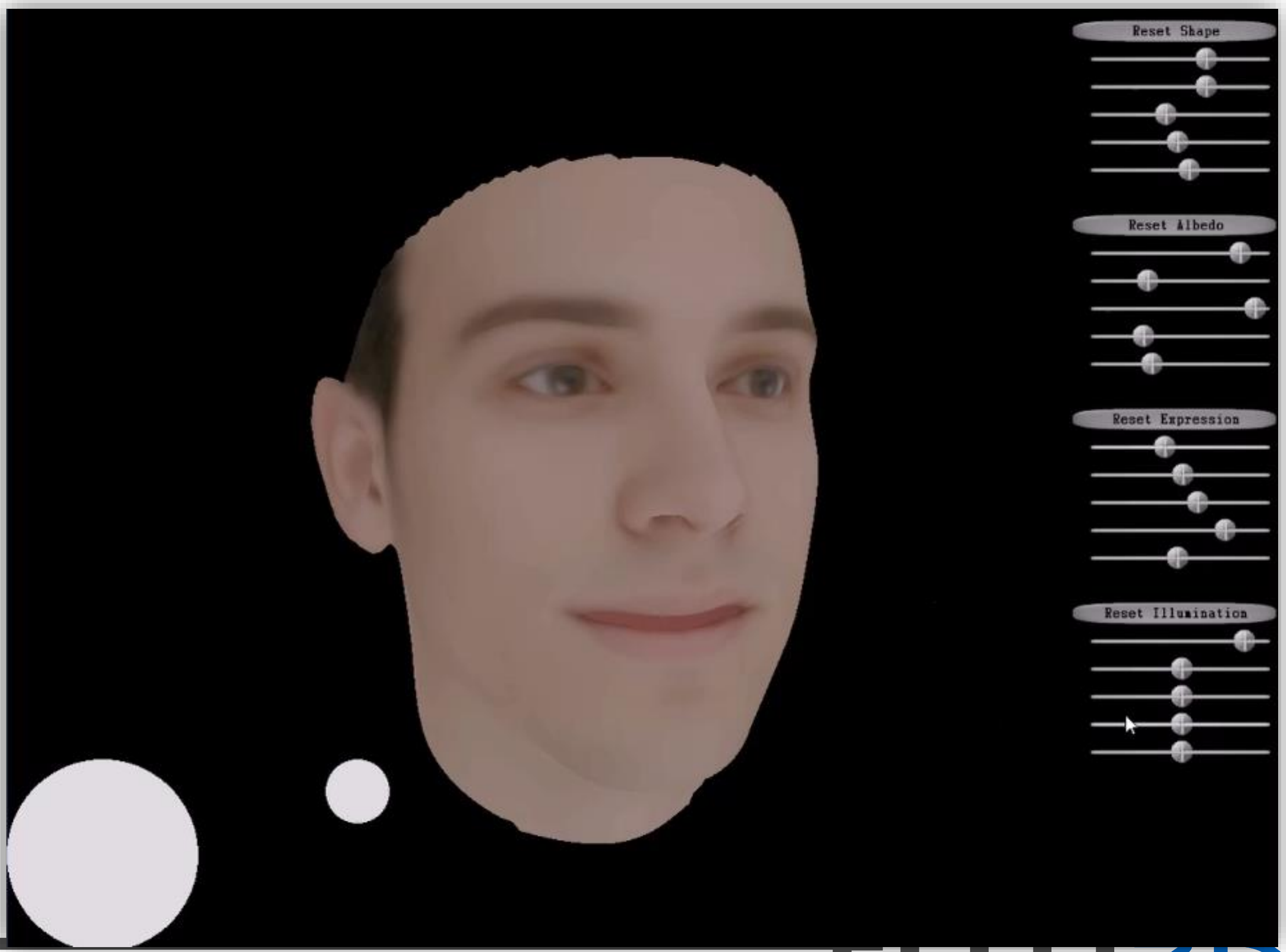
- Assumptions:
  - Lambertian surface
  - Distant smooth lighting



$$L = \sum \gamma_i B_i(\vec{n})$$

Illumination Models

- Spherical Harmonics [Mueller '66]
  - Orthogonal basis
  - Defined over a sphere



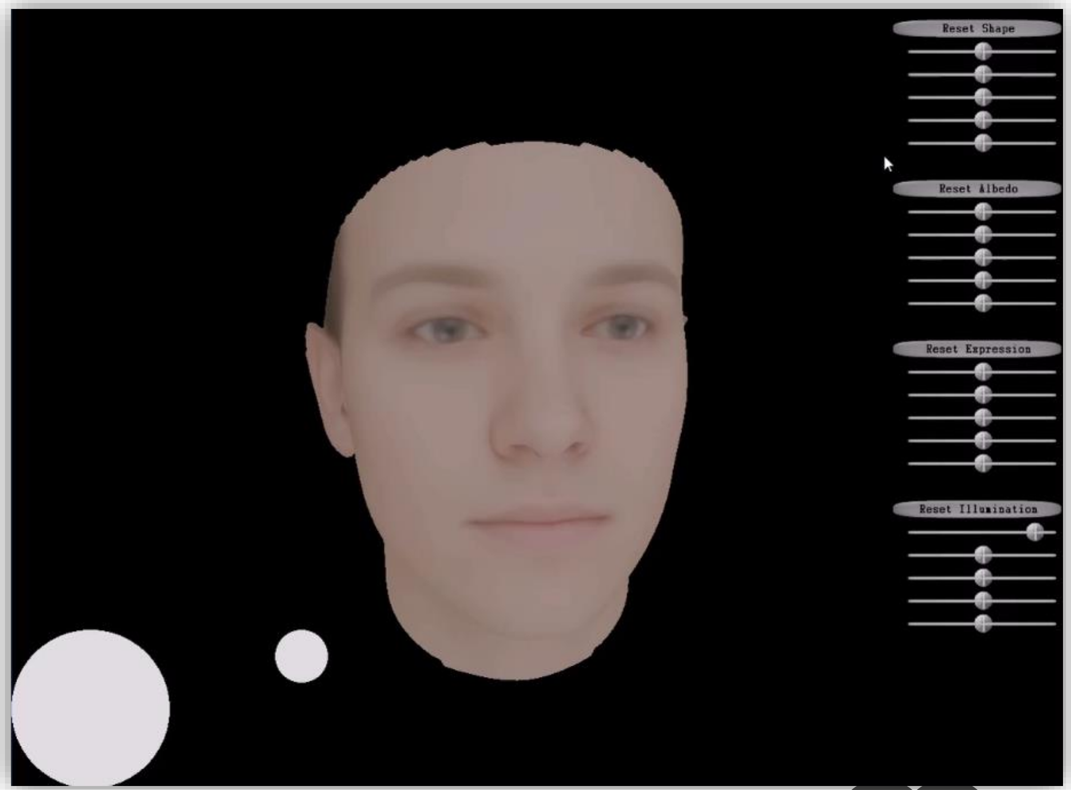
- Assumptions:
  - Lambertian surface
  - Distant smooth lighting

Parametric Model

- Morphable model [Blanz & Vetter '99]
  - Derived from 200 neutral scans (100 male + 100 female)
  - Two independent PCA models:

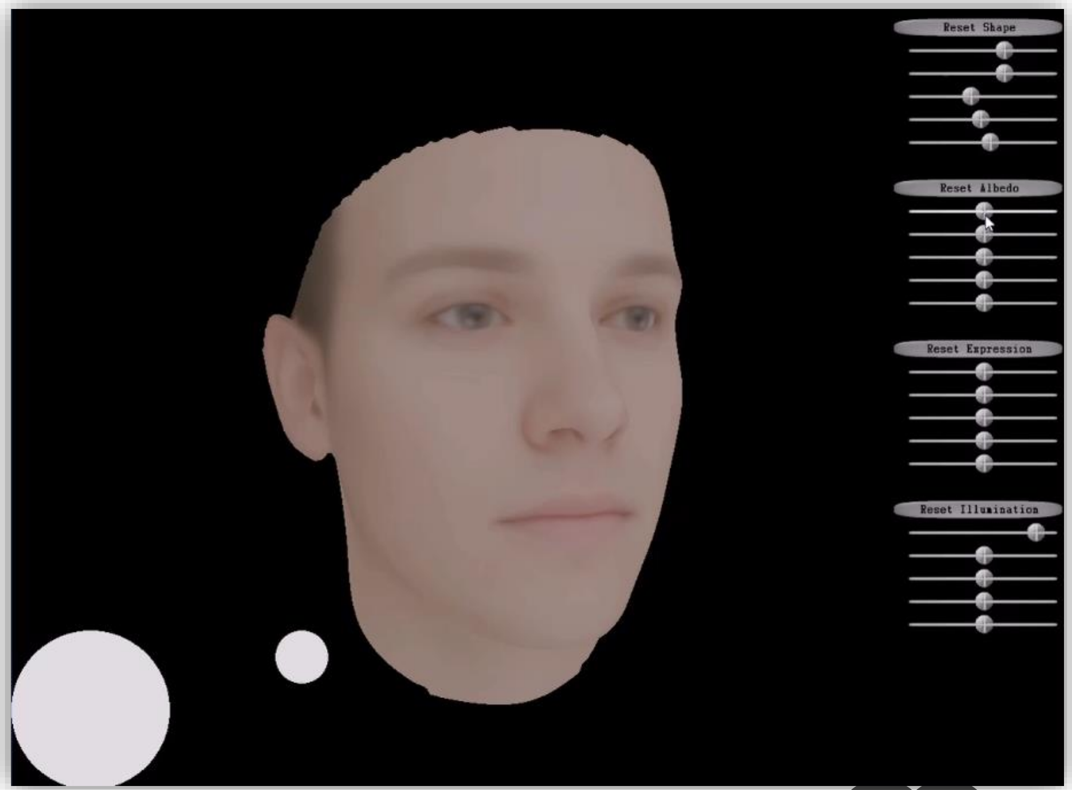
Parametric Model

- Morphable model [Blanz & Vetter '99]
  - Derived from 200 neutral scans (100 male + 100 female)
  - Two independent PCA models:
    - Identity



Parametric Model

- Morphable model [Blanz & Vetter '99]
  - Derived from 200 neutral scans (100 male + 100 female)
  - Two independent PCA models:
    - Identity
    - Albedo

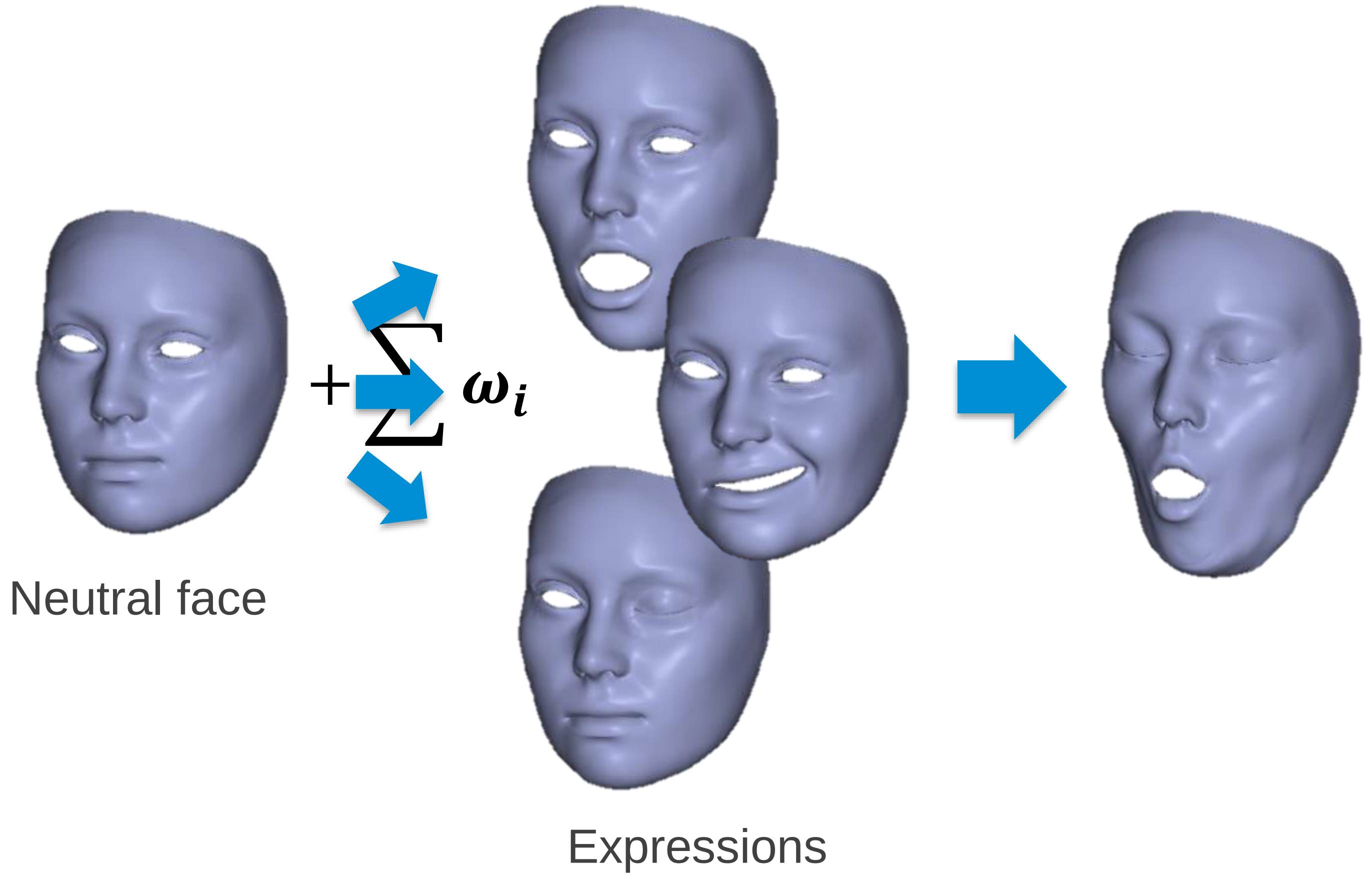


Parametric Model

- Morphable model [Blanz & Vetter '99]
  - Derived from 200 neutral scans (100 male + 100 female)
  - Two independent PCA models:
    - Identity
    - Albedo
- Drawbacks
  - Population bias (mainly Caucasians)
  - Parameters have global influence
  - No semantic deformations

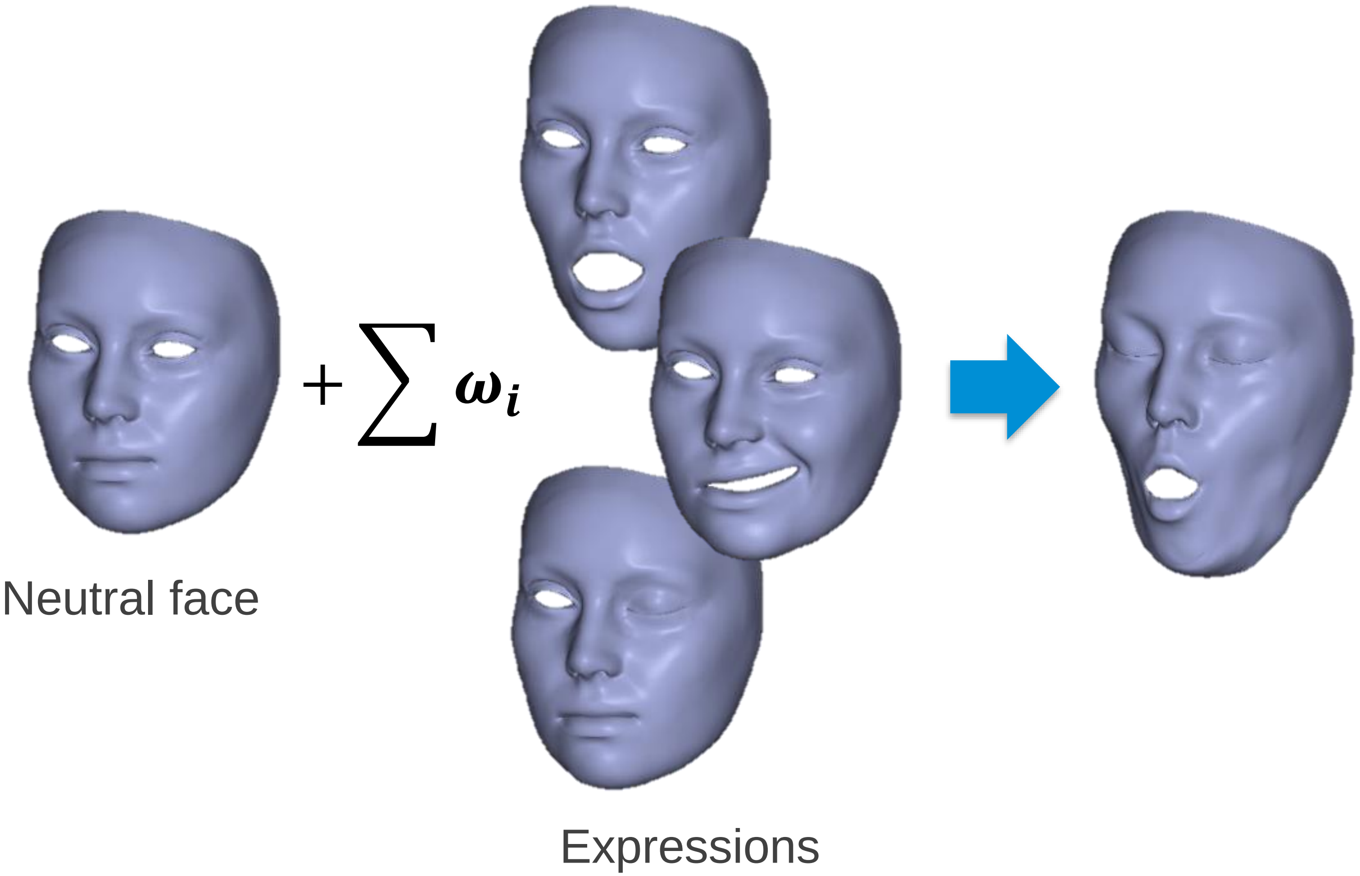
Parametric Model

- Blendshapes [Lewis et al. ‘14]
  - Additive model
  - Controlled via blendshape weights

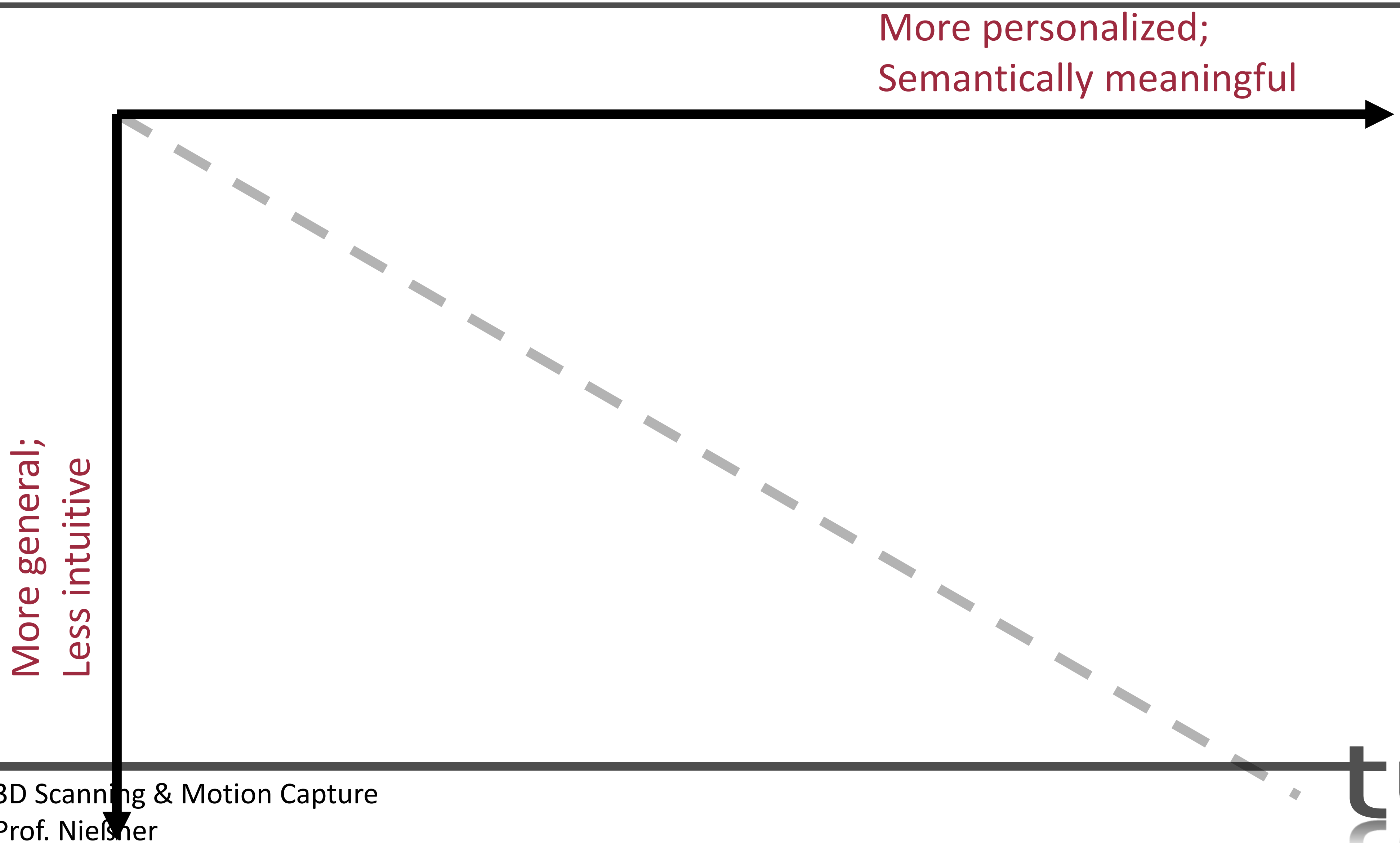


Parametric Model

- Blendshapes [Lewis et al. ‘14]
  - Additive model
  - Controlled via blendshape weights
  - Can be customized
- Popular choice
  - Delta blendshapes



Parametric Model



Parametric Model

More personalized;
Semantically meaningful

More general;
Less intuitive

Face rigs
[Ichim et al. ’15;
Garrido et al. 16]

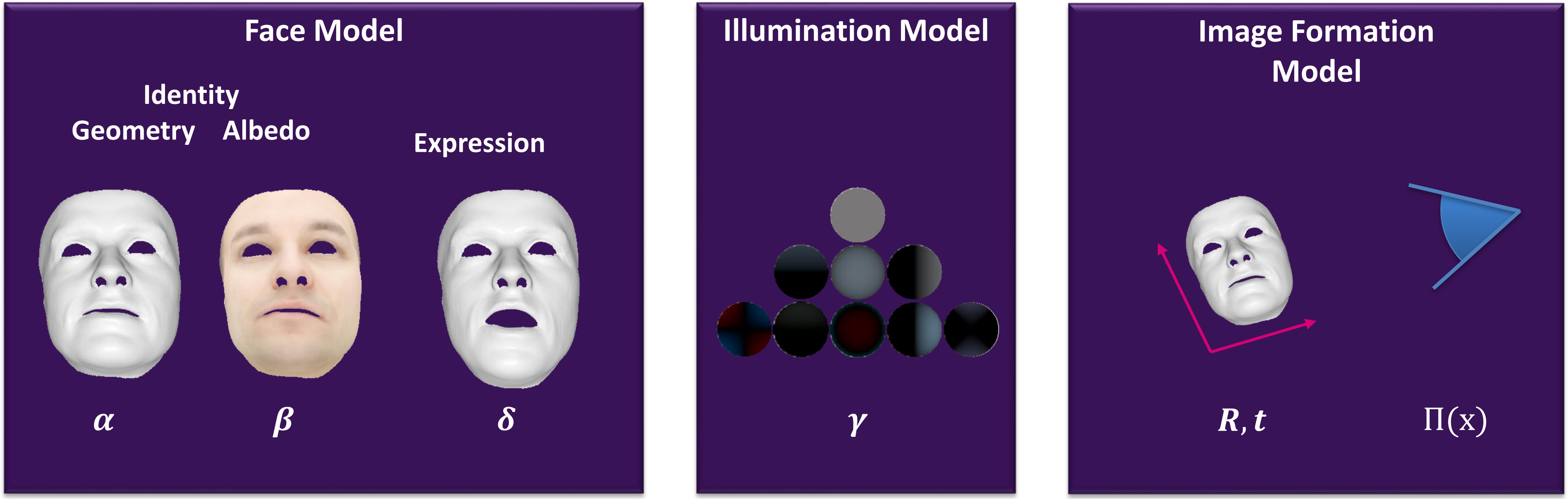
Blendshapes
[Garrido et al. ‘13;
Wu et al. ’16;
Thies et al. ‘16]

Morphable model
[Blanz&Vetter ’99; …]

Adaptive blendshapes
[Bouaziz et al. ‘13;
Li et al. ’13;
Hsieh et al. ‘15]

Multilinear model
[Vlasic et al. ’05;
Cao et al. ’14;
Shi et al. ‘14]

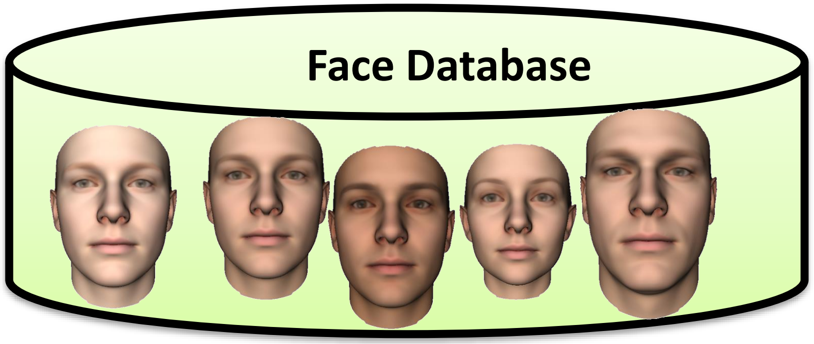
Parametric Face Model



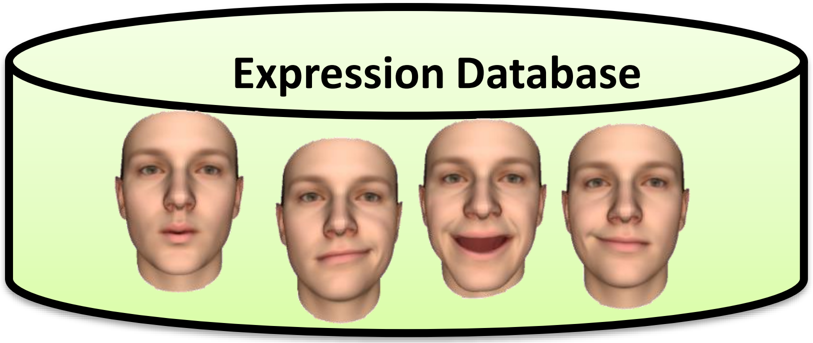
TOG'15 [Thies et al.]: Facical Reenactment

Parametric Face Model

- Offline / “Learning Phase”: Model Construction



- Run time / “Test Phase”: Model Fitting



How to Construct a Parametric Face Model?

- Assume we have a set of meshes of the face with the same topology but different poses
- Then, we have 1:1 correspondences between the vertices
- Basis functions are given through PCA of mesh (linear model)

How to Construct a Parametric Face Model?

- Compute face prior:
  - Scan a few hundred faces/expressions and fit a topologically-consistent template to each scan using non-rigid registration
  - Compute PCA-basis for identity and expressions
- At runtime: fit prior to input data

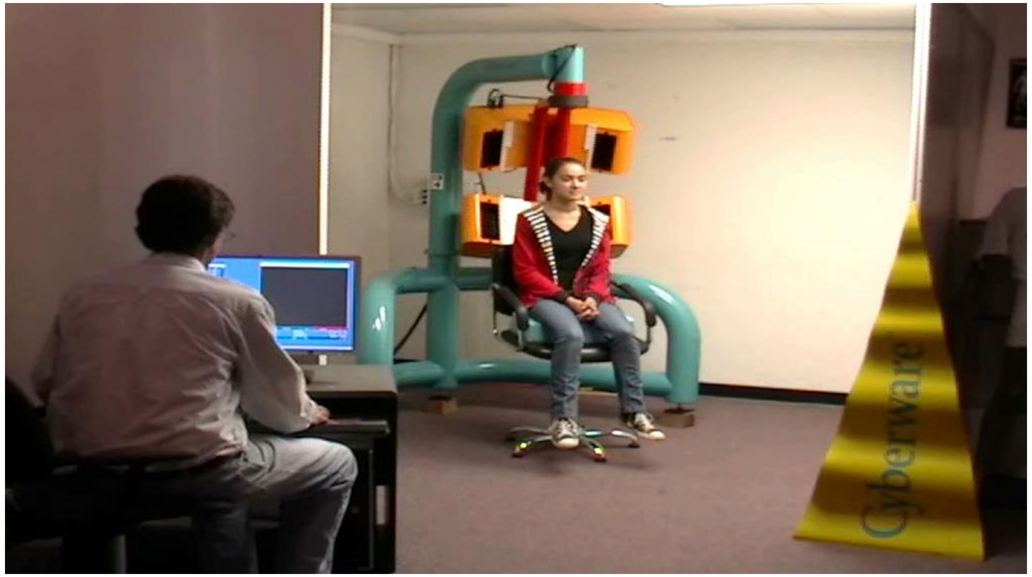
How to Construct a Parametric Face Model?

- Compute face prior:
  - Scan a few hundred faces/expressions and fit a topologically-consistent template to each scan using non-rigid registration
  - Compute PCA-basis for identity and expressions

- At runtime: fit prior to input data
  - E.g., from a sparse set of keypoints
  - E.g., from depth data
  - E.g., from RGB data

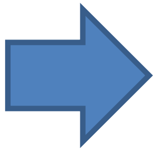
How to Construct a Parametric Face Model?

- Compute face prior:
  - **Scan a few hundred faces/expressions** and fit a topologically-consistent template to each scan using non-rigid registration



How to Construct a Parametric Face Model?

- Compute face prior:
  - **Scan a few hundred faces/expressions** and fit a topologically-consistent template to each scan using non-rigid registration



How to Construct a Parametric Face Model?

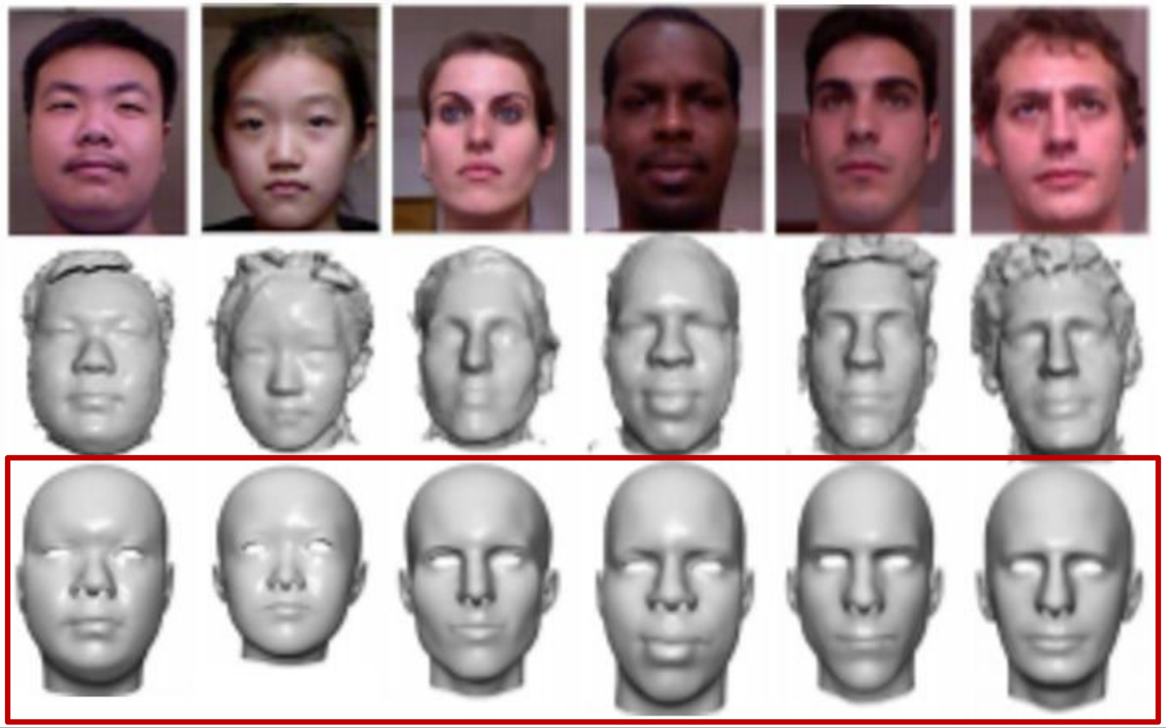
- Compute face prior:
  - **Scan a few hundred faces/expressions** and fit a topologically-consistent template to each scan using non-rigid registration

KinectFusion

Cao et al. “FaceWarehouse”

How to Construct a Parametric Face Model?

- Compute face prior:
  - **Scan a few hundred faces/expressions** and **fit a topologically-consistent template** to each scan using non-rigid registration



Cao et al. “FaceWarehouse”

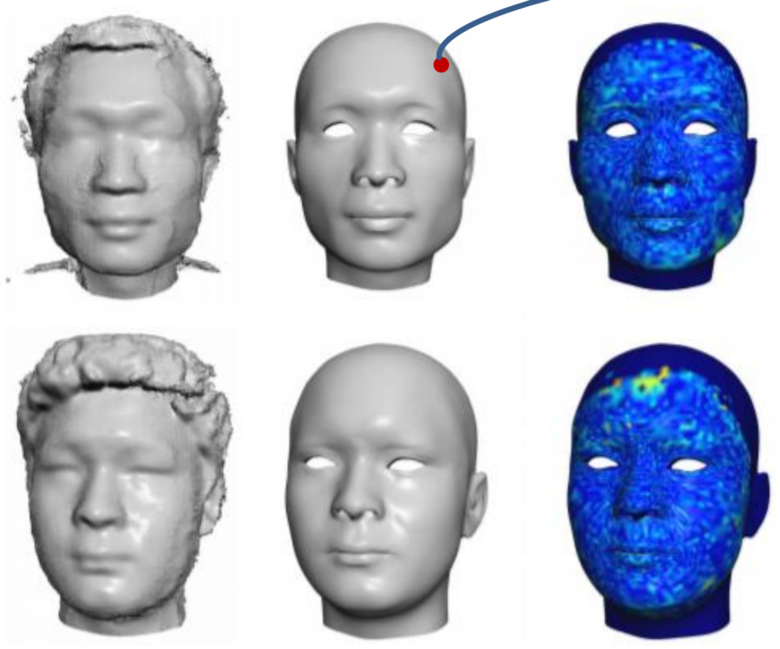
How to Construct a Parametric Face Model?

- Compute face prior:
  – **Scan a few hundred faces/expressions** and **fit a topologically-consistent template** to each scan using **non-rigid registration**

Cao et al. “FaceWarehouse”

How to Construct a Parametric Face Model?

- Compute face prior:
  - **Scan a few hundred faces/expressions** and **fit a topologically-consistent template** to each scan using **non-rigid registration**



$$V_0 = \begin{pmatrix} x_0 \\ y_0 \\ z_0 \\ \vdots \\ x_{n-1} \\ y_{n-1} \\ z_{n-1} \end{pmatrix}$$

vertices of the deformed template mesh

$$V_1 = \begin{pmatrix} x_0 \\ y_0 \\ z_0 \\ \vdots \\ x_{n-1} \\ y_{n-1} \\ z_{n-1} \end{pmatrix}$$

Cao et al. “FaceWarehouse”

How to Construct a Parametric Face Model?

- Compute face prior:
  - Scan a few hundred faces/expressions and fit a topologically-consistent template to each scan using non-rigid registration
  - **Compute PCA-basis** for identity [and expressions]

$$V_0 = \begin{pmatrix} x_0 \\ y_0 \\ z_0 \\ \vdots \\ x_{n-1} \\ y_{n-1} \\ z_{n-1} \end{pmatrix} \quad \cdots \quad V_{m-1} = \begin{pmatrix} x_0 \\ y_0 \\ z_0 \\ \vdots \\ x_{n-1} \\ y_{n-1} \\ z_{n-1} \end{pmatrix} \xrightarrow{\textbf{PCA}}$$

Average: $\bar{V}$

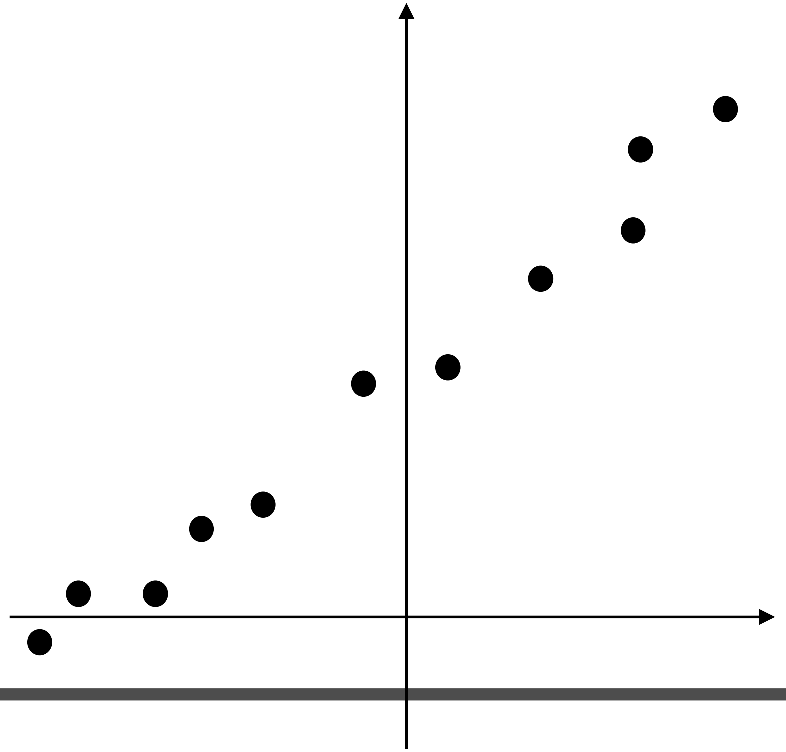
Principle Components: $\vec{E}_0 \cdots \vec{E}_{m-2}$

Std. Deviation: $\sigma_0 \cdots \sigma_{m-2}$

Principle Component Analysis (PCA)

- Compute mean:

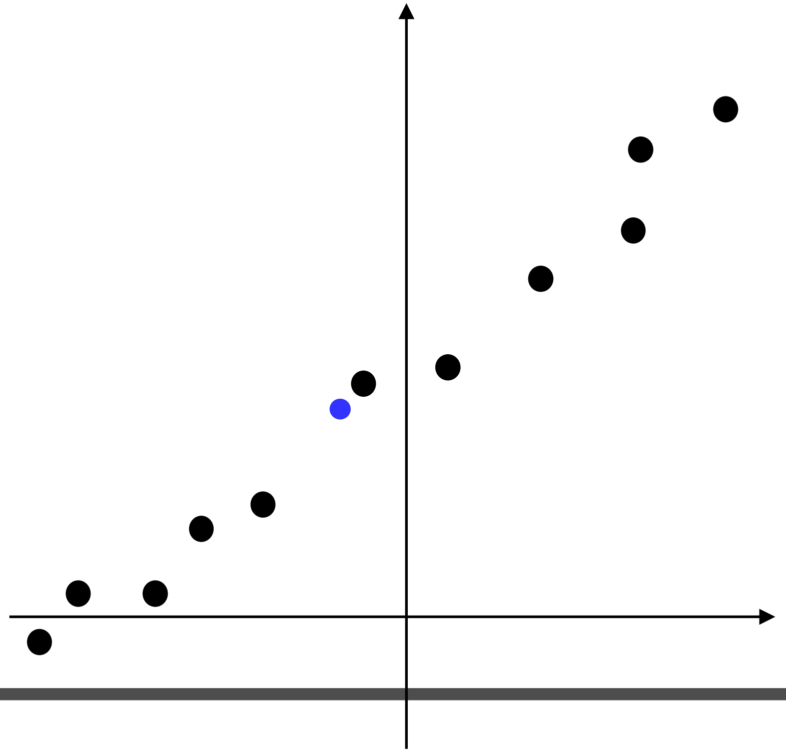
$$\overline{\mathbf{X}} = \frac{1}{N}\sum_i \mathbf{X}_i \,, \quad \overline{\mathbf{X}}_i = \mathbf{X}_i - \overline{\mathbf{X}}$$



Principle Component Analysis (PCA)

- Compute mean:

$$\overline{\mathbf{X}} = \frac{1}{N}\sum_i \mathbf{X}_i \,, \quad \overline{\mathbf{X}}_i = \mathbf{X}_i - \overline{\mathbf{X}}$$

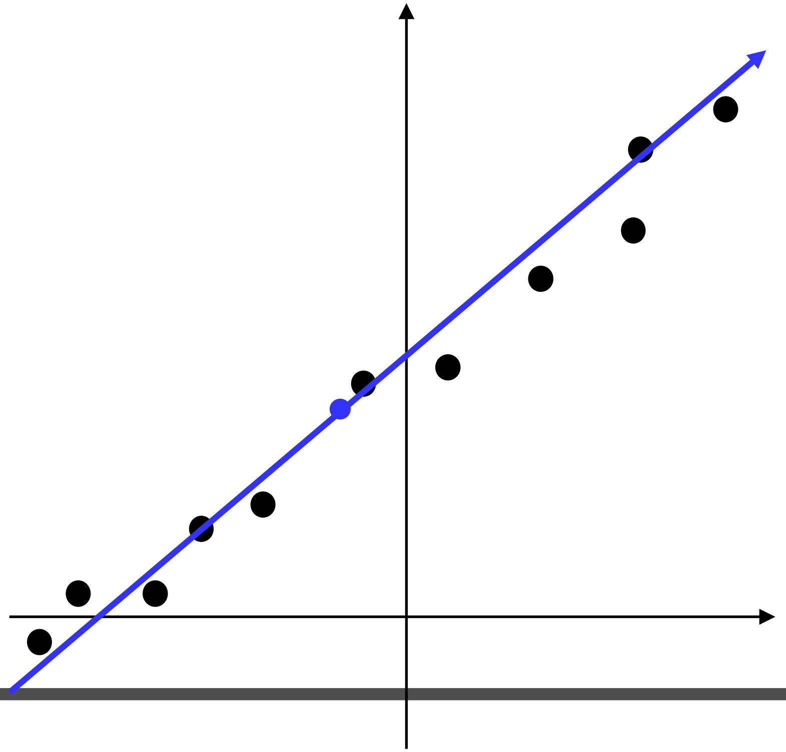


Principle Component Analysis (PCA)

- Compute mean:

$$\overline{\mathbf{x}} = \frac{1}{N}\sum_i \mathbf{x}_i \,, \quad \overline{\mathbf{x}}_i = \mathbf{x}_i - \overline{\mathbf{x}}$$

- Compute principle components of the dataset $\bar{X}_i$

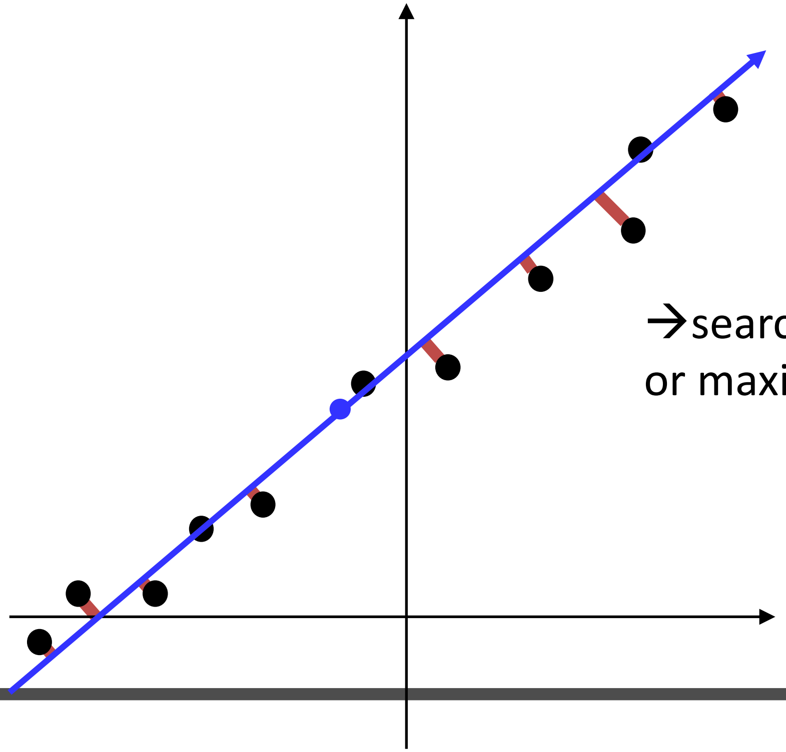


Principle Component Analysis (PCA)

- Compute mean:

$$\overline{\mathbf{X}} = \frac{1}{N}\sum_i \mathbf{X}_i \,, \quad \overline{\mathbf{X}}_i = \mathbf{X}_i - \overline{\mathbf{X}}$$

- Compute principle components of the dataset $\overline{X_i}$



→search direction that minimizes the distance, or maximizes the projection

$$argmax_{\vec{\omega}} \sum_i \left| |(X_i - \bar{X}) \cdot \vec{\omega}| \right|^2$$

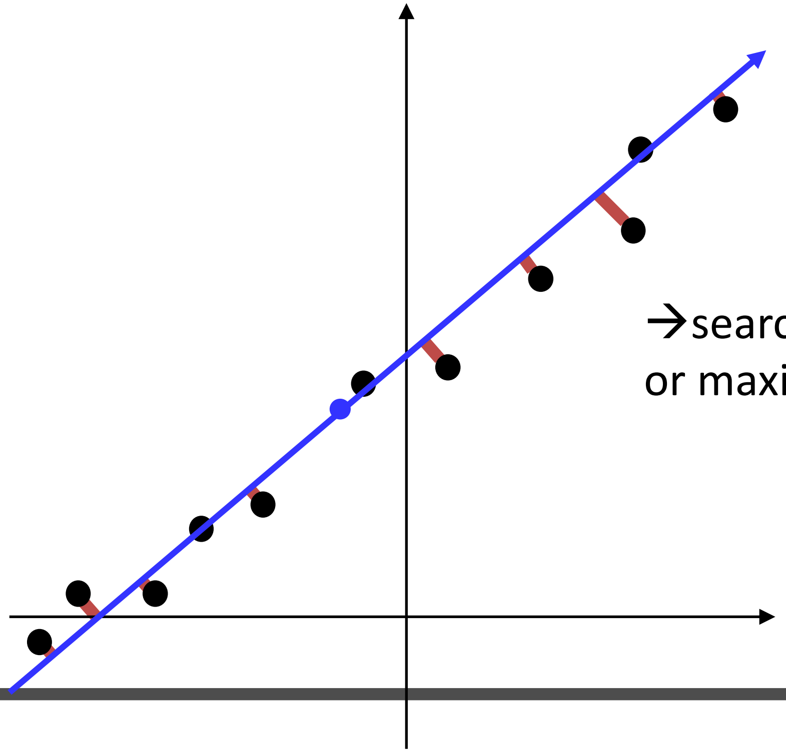
$$s.t. \left| |\vec{\omega}| \right|^2 = 1$$

Principle Component Analysis (PCA)

- Compute mean:

$$\overline{\mathbf{X}} = \frac{1}{N}\sum_i \mathbf{X}_i\,,\quad \overline{\mathbf{X}}_i = \mathbf{X}_i - \overline{\mathbf{X}}$$

- Compute principle components of the dataset $\overline{X}_i$



→search direction that minimizes the distance, or maximizes the projection

$$argmax_{\vec{\omega}} \sum_i \left|\left|(X_i - \bar{X}) \cdot \vec{\omega}\right|\right|^2$$

$$s.t. \left|\left|\vec{\omega}\right|\right|^2 = 1$$

Eigenvectors of the Covariance Matrix

Principle Component Analysis (PCA)

- Compute mean:

$$\overline{\mathbf{X}} = \frac{1}{N}\sum_i \mathbf{X}_i \,, \quad \overline{\mathbf{X}}_i = \mathbf{X}_i - \overline{\mathbf{X}}$$

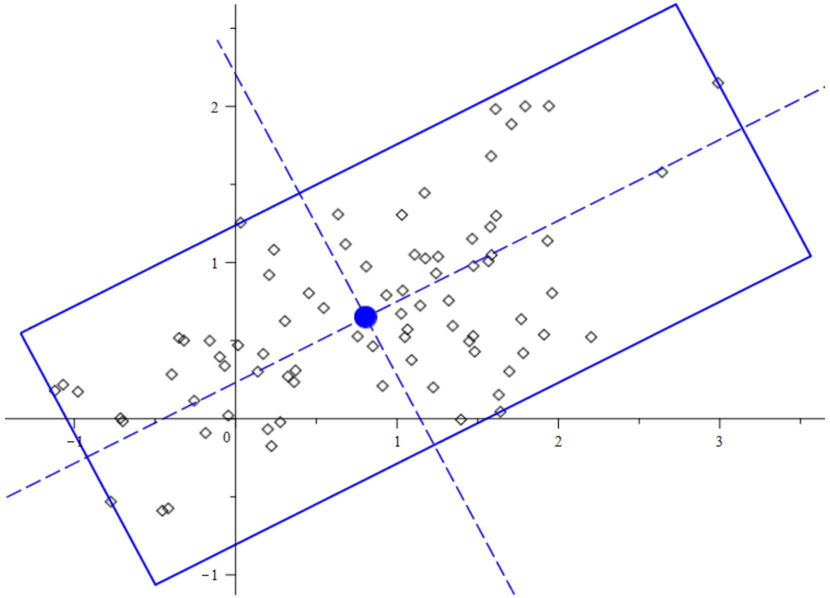
- Compute principle components of the dataset $\overline{X}_i$
  - Compute covariance matrix:

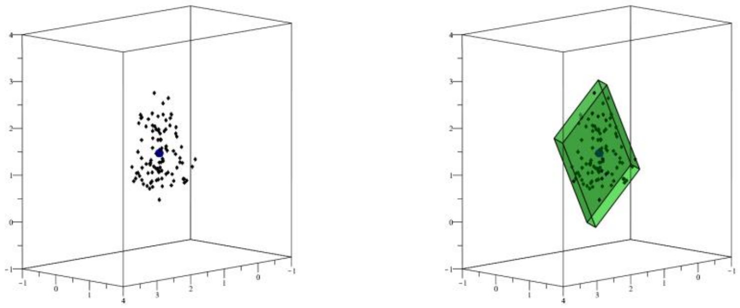
$$\boldsymbol{C} = \operatorname{cov}(\boldsymbol{X}, \boldsymbol{X}) = \frac{1}{N-1}\sum_i \overline{\boldsymbol{X}}_i \cdot \overline{\boldsymbol{X}}_i^{\,T}$$

  - Compute Eigenvectors and Eigenvalues
    - Eigenvectors: Principle components
    - Eigenvalues: Variance

Principle Component Analysis (PCA)

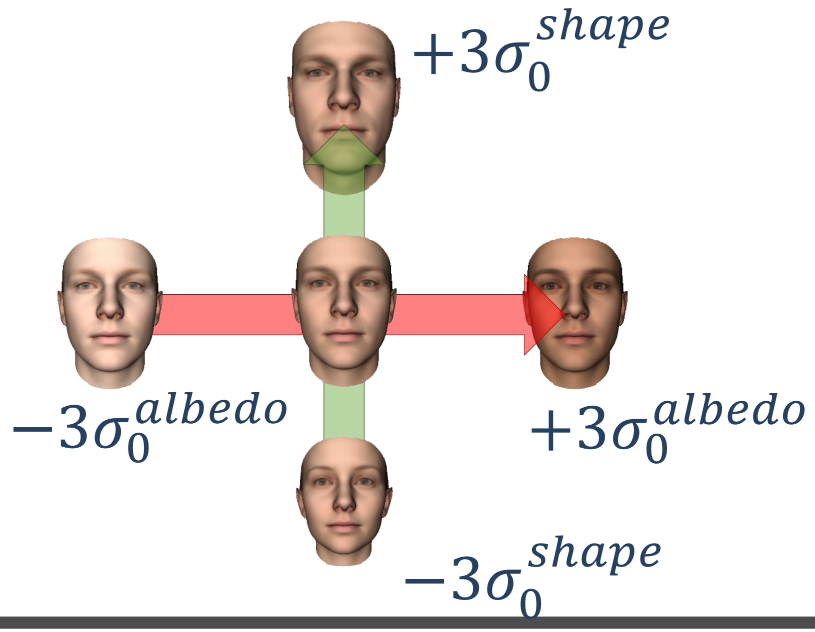
- Applications:
  - Reduction of dimensionality → approximation
  - Object oriented bounding boxes (OOBB)





How to Construct a Parametric Face Model?

- Compute face prior:
  - Scan a few hundred faces/expressions and fit a topologically-consistent template to each scan using non-rigid registration
  - **Compute PCA-basis** for identity [and expressions]



Note: the PCA is independently applied to shape and albedo

How to Construct a Parametric Face Model?

- Compute face prior:
  - Scan a few hundred faces/expressions and fit a topologically-consistent template to each scan using non-rigid registration
  - **Compute PCA-basis** for identity [and expressions]
    - New faces can be computed using a linear combination of the basis vectors

$$
\begin{aligned}
\mathcal{M}_{\text{geo}}(\boldsymbol{\alpha}, \boldsymbol{\delta}) &= \boldsymbol{a}_{\text{id}} + E_{\text{id}} \cdot \boldsymbol{\alpha} + E_{\text{exp}} \cdot \boldsymbol{\delta} \\
\mathcal{M}_{\text{alb}}(\boldsymbol{\beta}) &= \boldsymbol{a}_{\text{alb}} + E_{\text{alb}} \cdot \boldsymbol{\beta}
\end{aligned}
$$

a_{id}: Shape Average
a_{alb}: Albedo Average

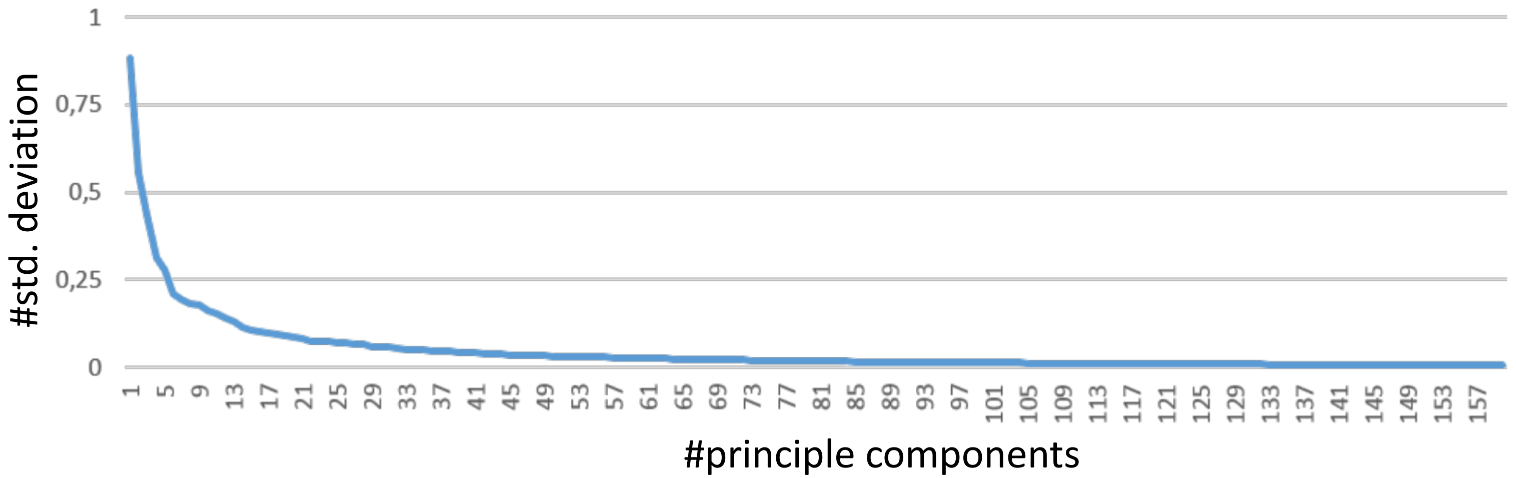
E_{id}: Shape PCA basis

E_{exp}: Expression PCA basis

E_{alb}: Albedo PCA basis

How to Construct a Parametric Face Model?

- Compute face prior:
  - Scan a few hundred faces/expressions and fit a topologically-consistent template to each scan using non-rigid registration
  - **Compute PCA-basis** for identity [and expressions]
    - New faces can be computed using a linear combination of the PCA basis

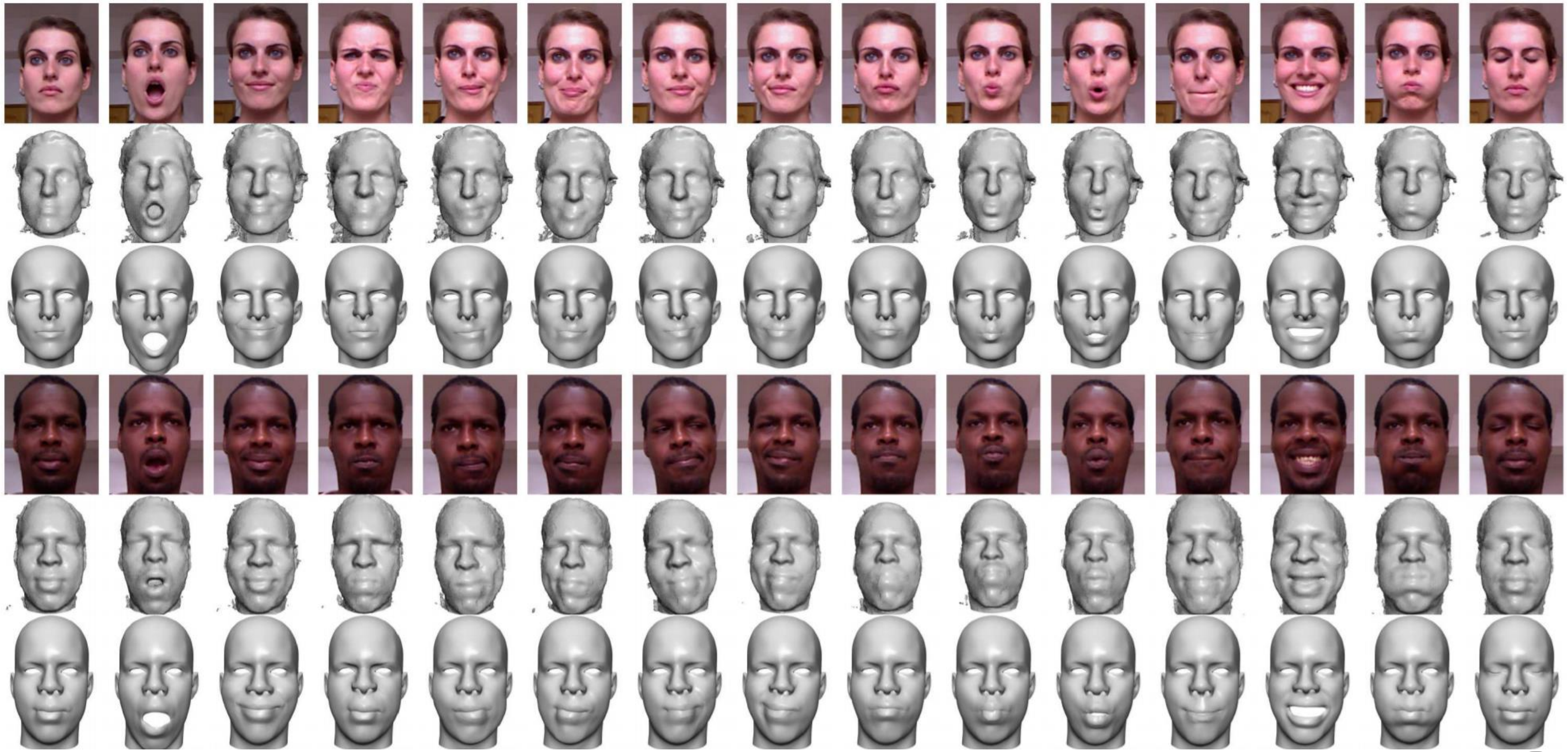


Std. deviation drops quickly

➢ Use this property to reduce computation time (e.g., only use the first 80 principle components)

How to Construct a Parametric Face Model?

- Expressions:



How to Construct a Parametric Face Model?

- Expressions:
  - PCA basis representation:
    - Loss of semantic meaning
    - Can be used to compress the data

  - Delta-Blendshape basis:
    - Delta-Blendshape: vertex displacement of a specific expression to the neutral face
    - Use linear combination to interpolate between expressions (same as for PCA)
    - Keeps semantic meaning
      - ➢ can be used to transfer expressions to another Blendshape model

How to Construct a Parametric Face Model?

- Compute face prior:
  - Scan a few hundred faces/expressions and fit a topologically-consistent template to each scan using non-rigid registration
  - **Compute PCA-basis** for identity [and expressions]
    - New faces can be computed using a linear combination of the basis vectors

$$\begin{aligned} \mathcal{M}_{\text{geo}}(\boldsymbol{\alpha}, \boldsymbol{\delta}) &= \boldsymbol{a}_{\text{id}} + E_{\text{id}} \cdot \boldsymbol{\alpha} + E_{\text{exp}} \cdot \boldsymbol{\delta} \\ \mathcal{M}_{\text{alb}}(\boldsymbol{\beta}) &= \boldsymbol{a}_{\text{alb}} + E_{\text{alb}} \cdot \boldsymbol{\beta} \end{aligned}$$

- At runtime: fit prior to input data
  - Estimate coefficients of the linear combination ($\alpha, \beta, and\ \delta$)

Image Formation Model

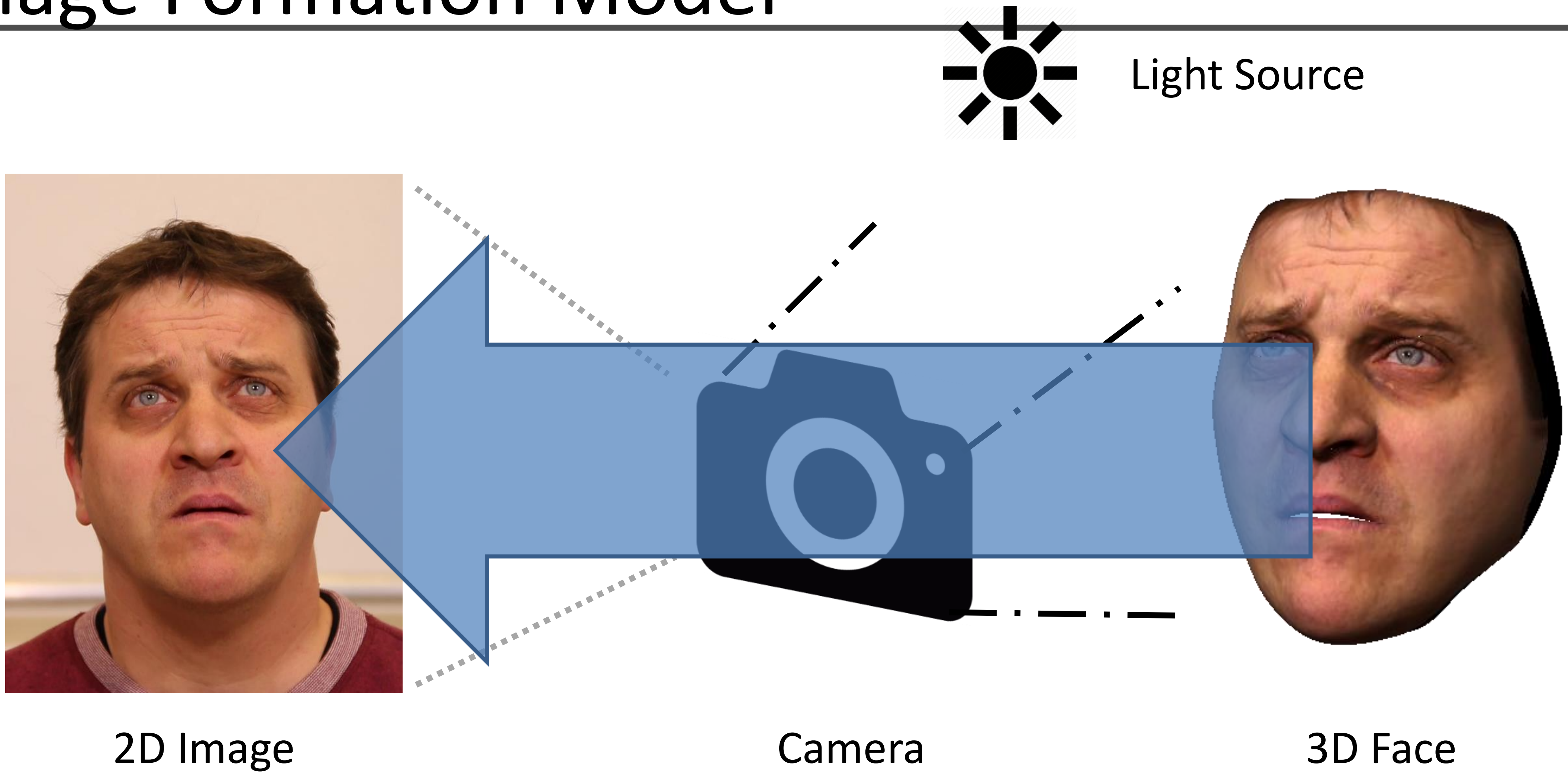
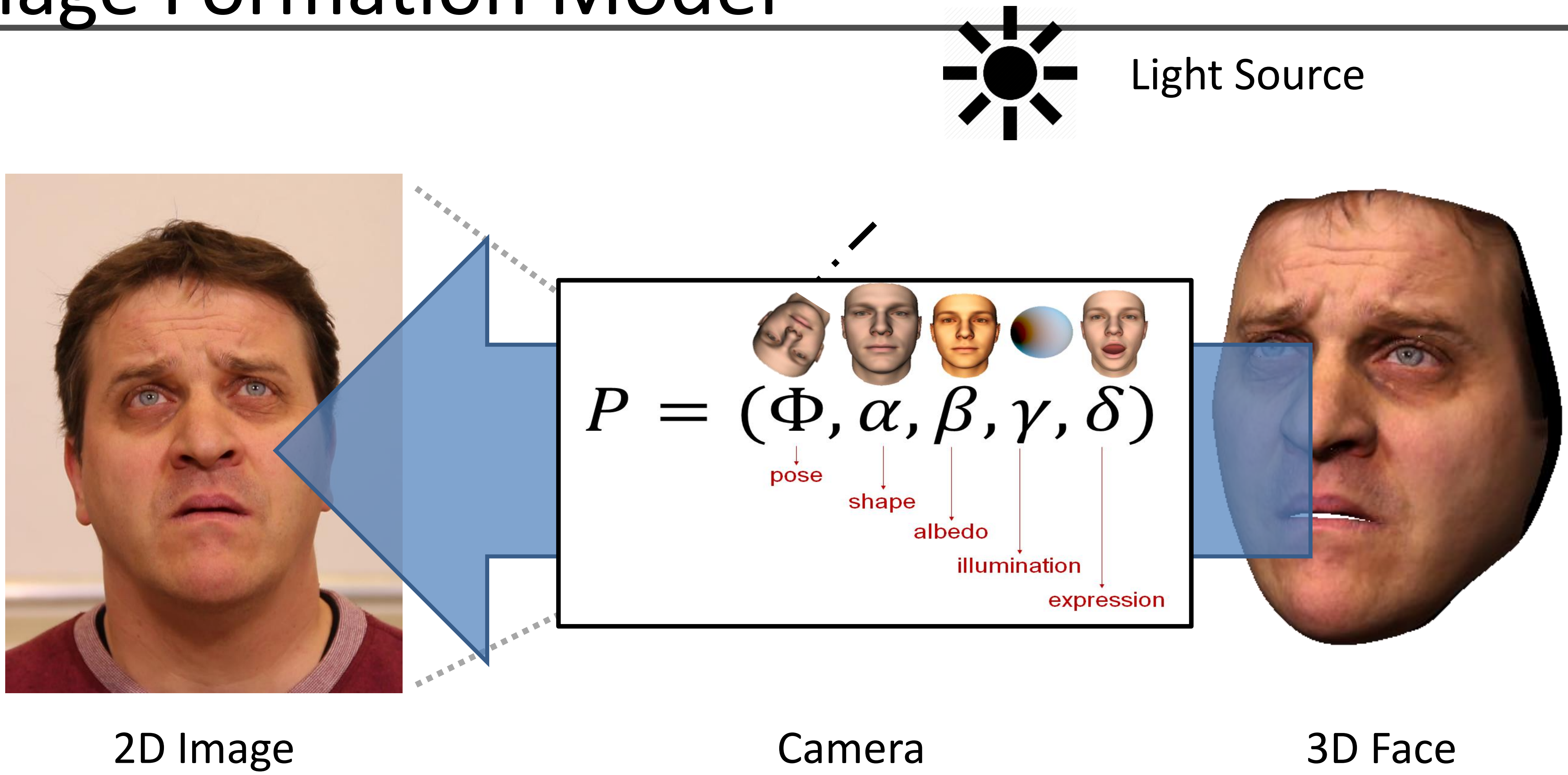
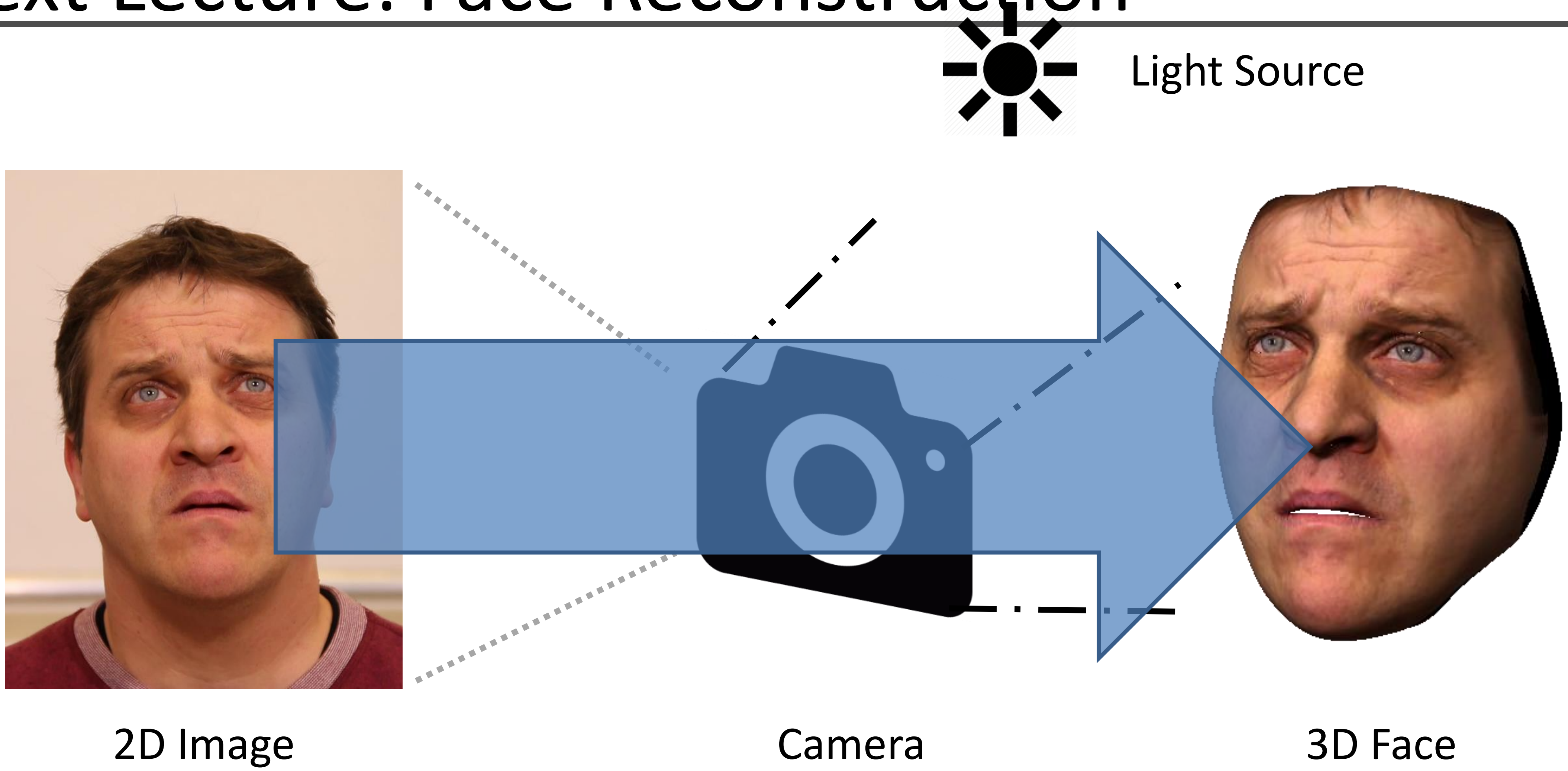


Image Formation Model

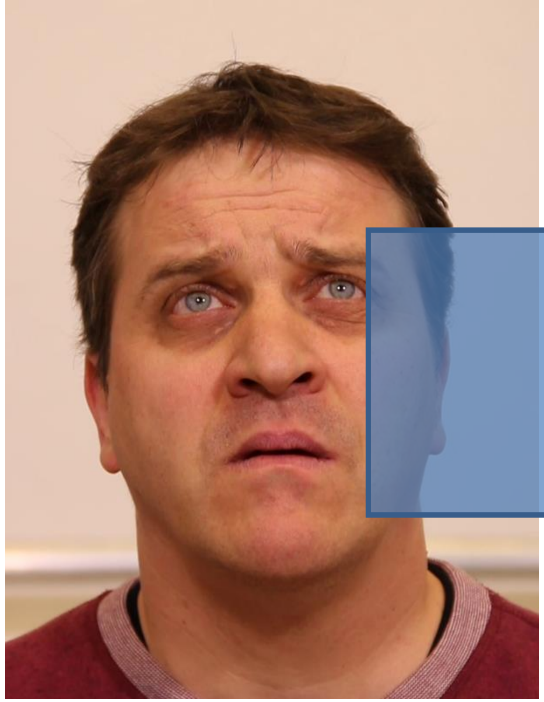


$$P = (\Phi, \alpha, \beta, \gamma, \delta)$$

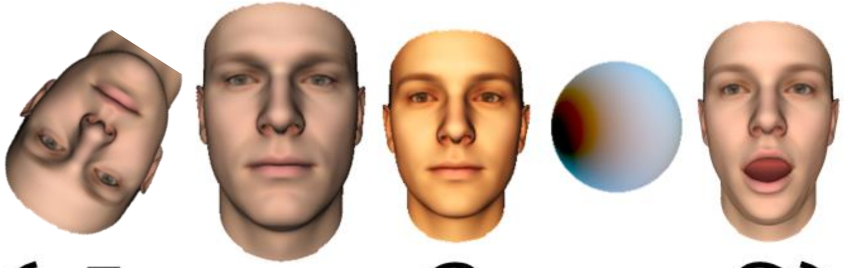
Next Lecture: Face Reconstruction



Next Lecture: Face Reconstruction



2D Image



$$P = (\underbrace{\Phi}_{\text{pose}}, \underbrace{\alpha}_{\text{shape}}, \underbrace{\beta}_{\text{albedo}}, \underbrace{\gamma}_{\text{illumination}}, \underbrace{\delta}_{\text{expression}})$$

Administrative

- Reading Homework:
  - [Blanz and Vetter 99] A Morphable Model For The Synthesis Of 3D Faces
    https://dl.acm.org/doi/pdf/10.1145/311535.311556
  - Read up on PCA – understand the correlations to SVD and Covariance Matrices
- Next week:
  - Face Tracking & Reconstruction

Administrative

See you next week!