3D Scanning & Motion Capture

Face Tracking and Reconstruction

Prof. Dr. Nießner

Dynamic 3D Capture

Need more regularization with fewer DoF to make it practical!

Domain-Specific

- Human body
- Hands
- Faces

Free-form Reconstruction

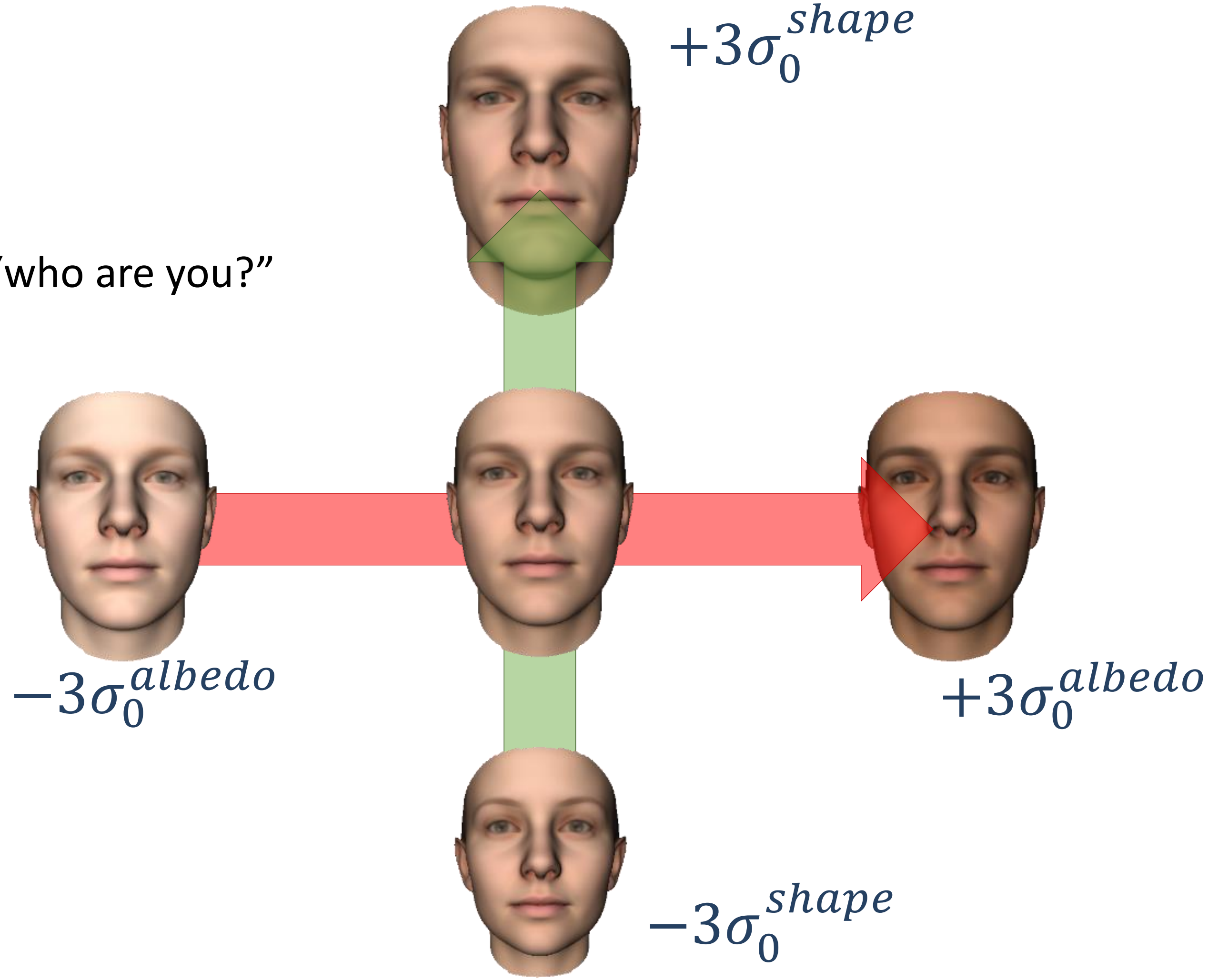
- Joint reconstruction and tracking

Free-form Tracking

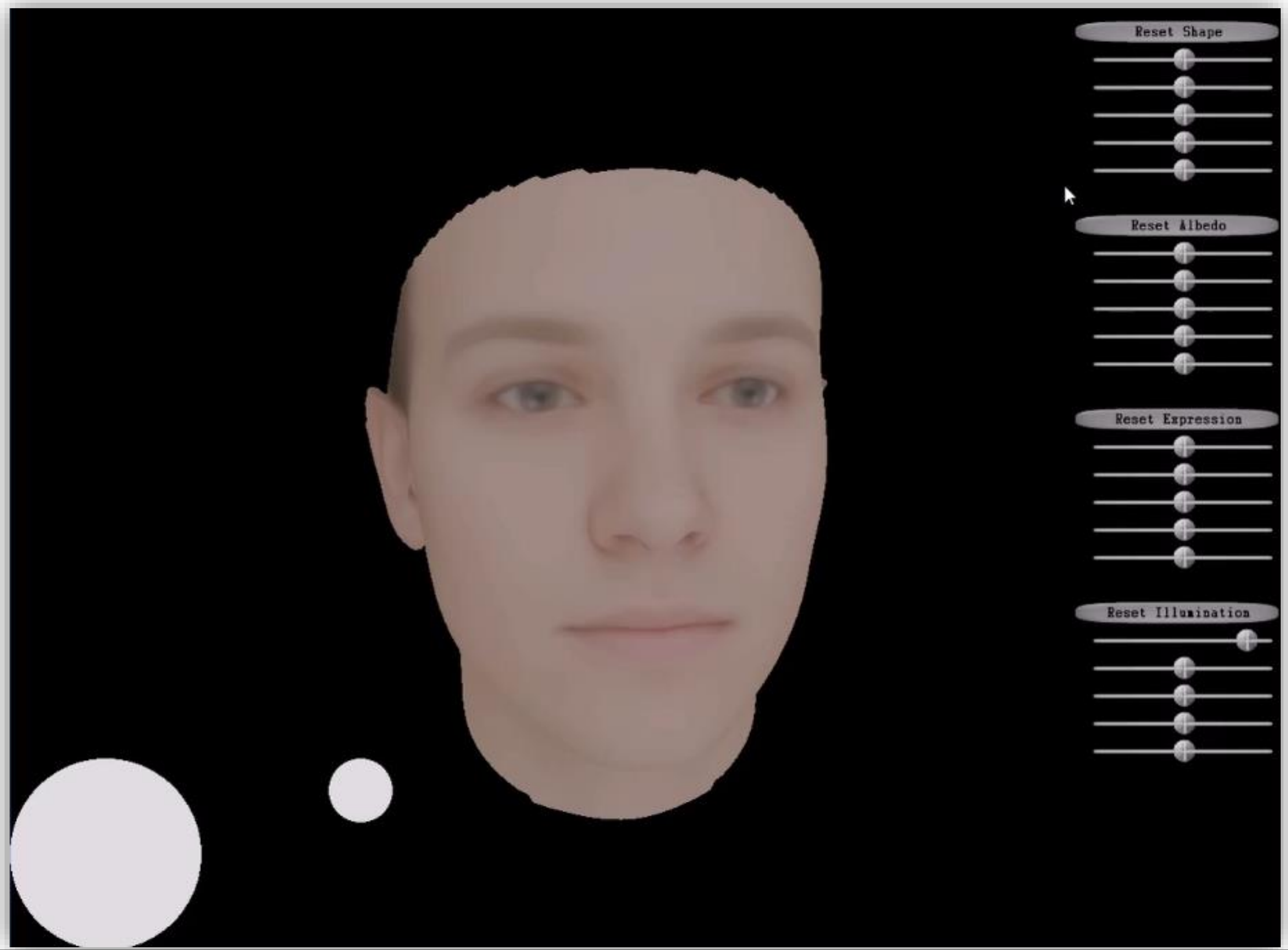
- Offline template reconstruction
- Online template tracking

Parametric Face Model: Shape Identity

Shape Identity defines “who are you?”

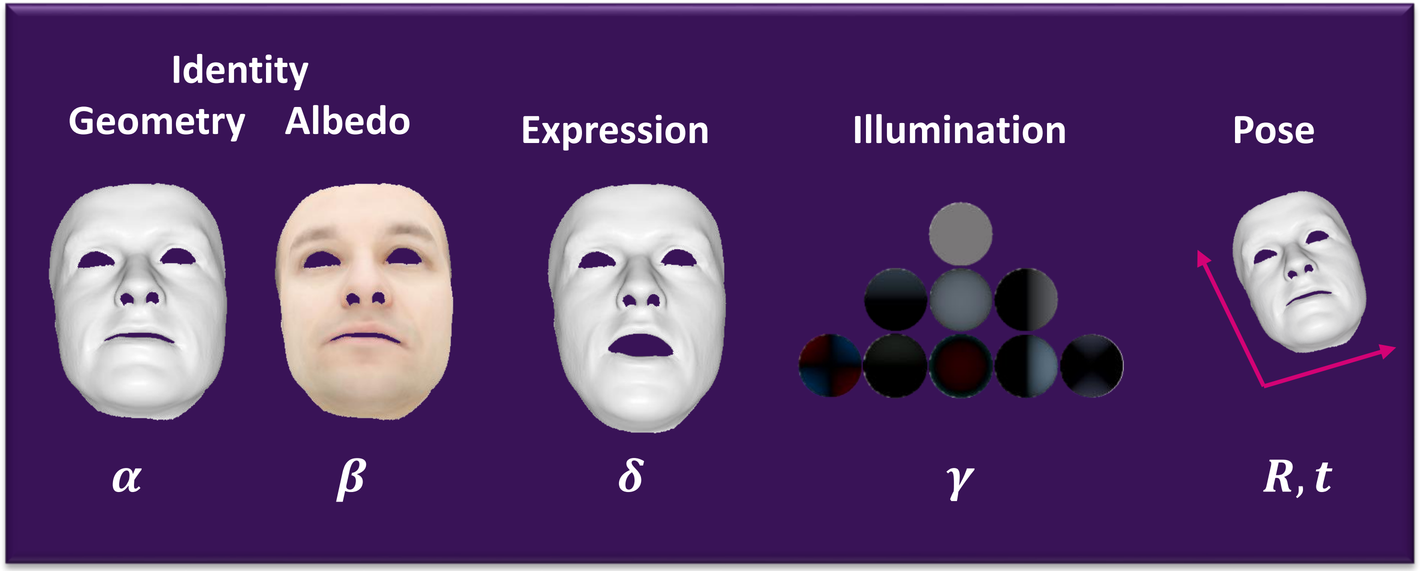


Parametric Model



Parametric Model

$P =$



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

Today: Face Tracking and Reconstruction

Image Formation Model

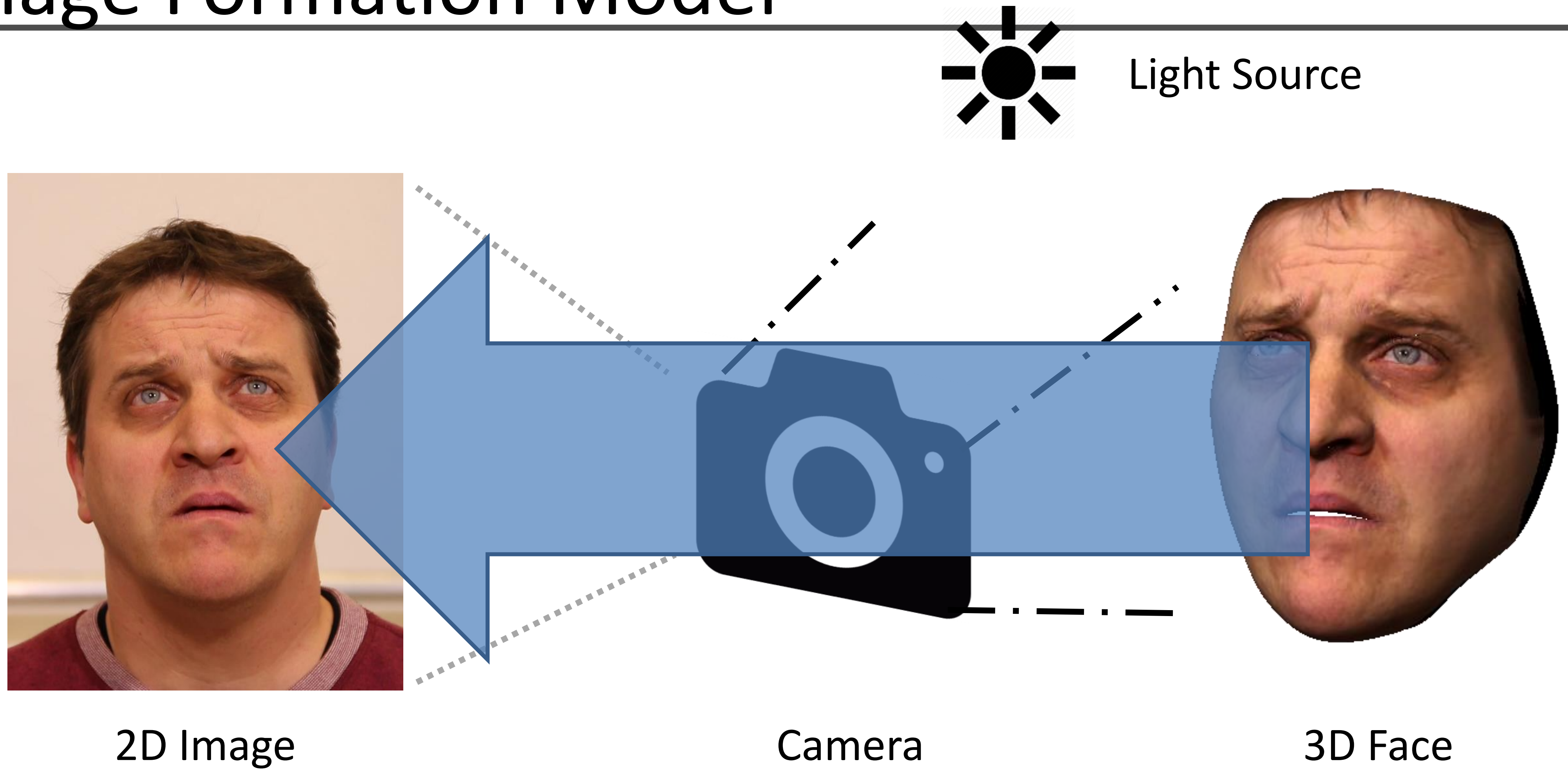
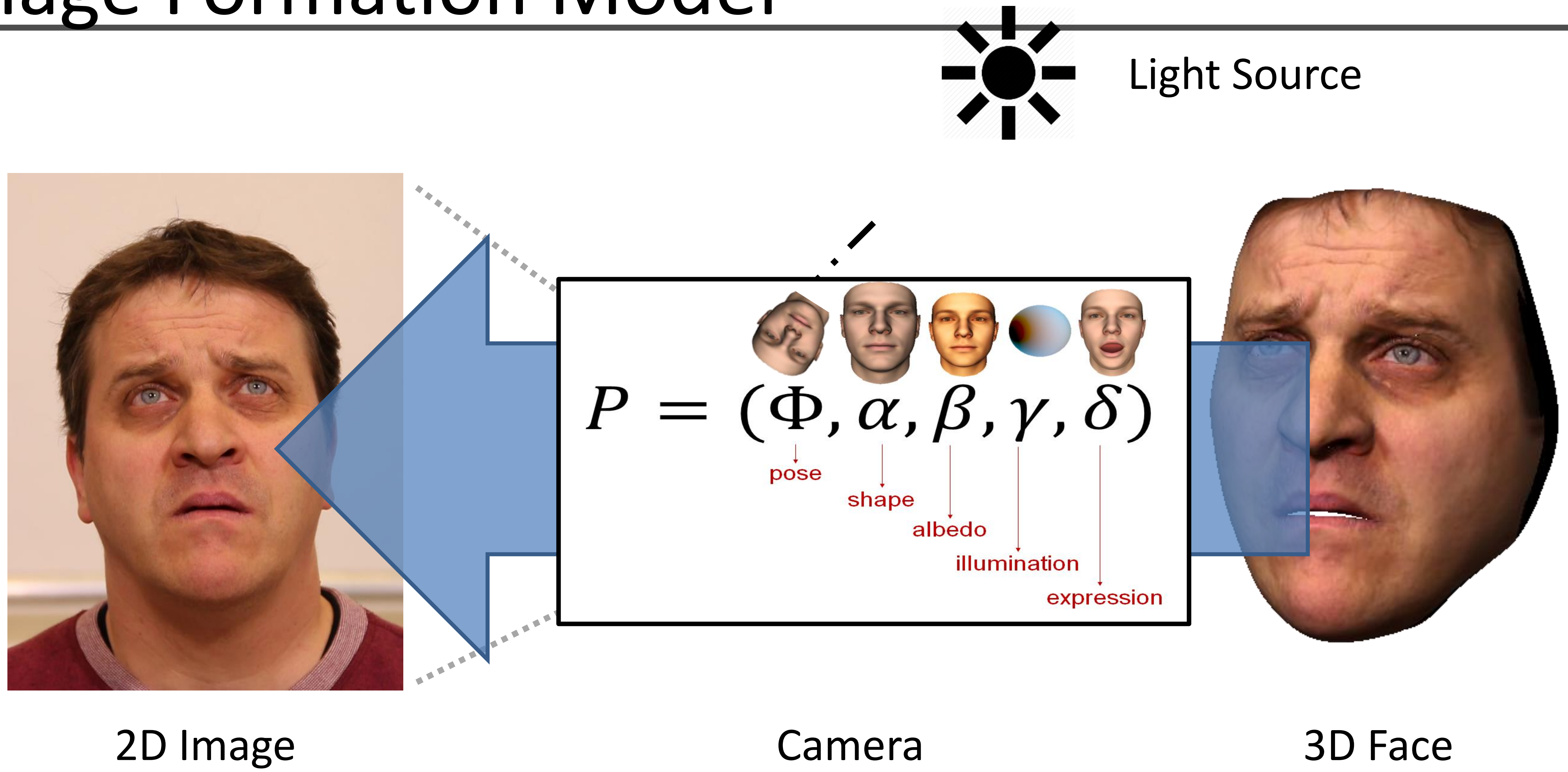
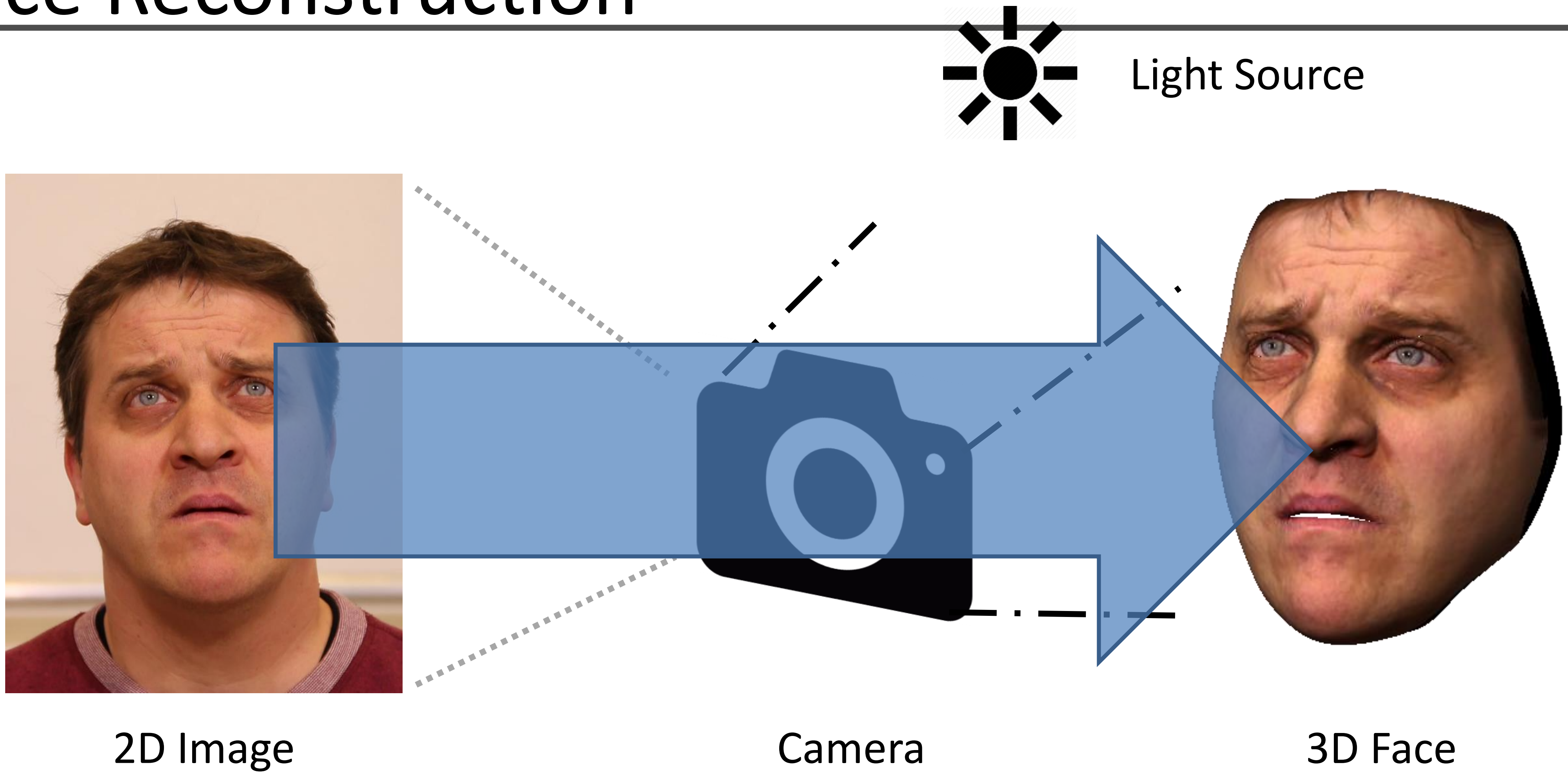


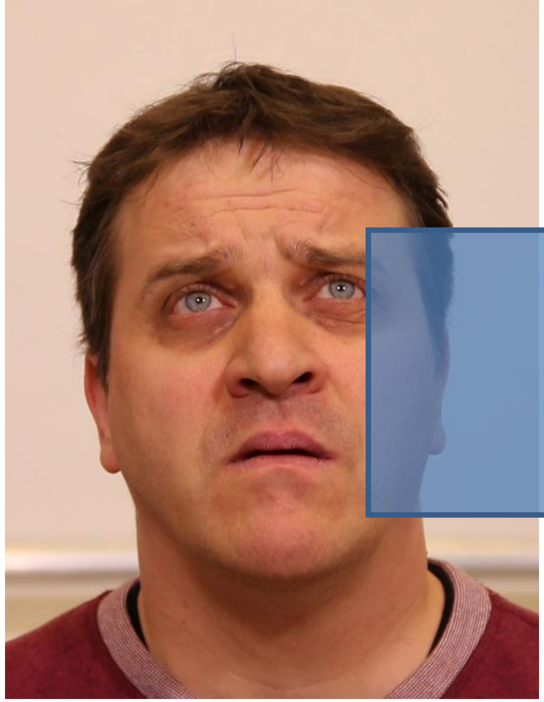
Image Formation Model



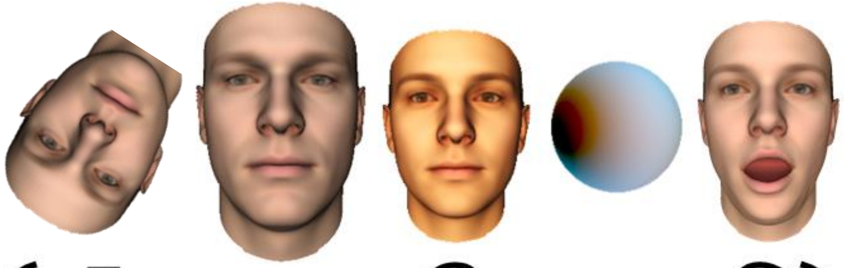
Face Reconstruction



Face Reconstruction



2D Image



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

pose

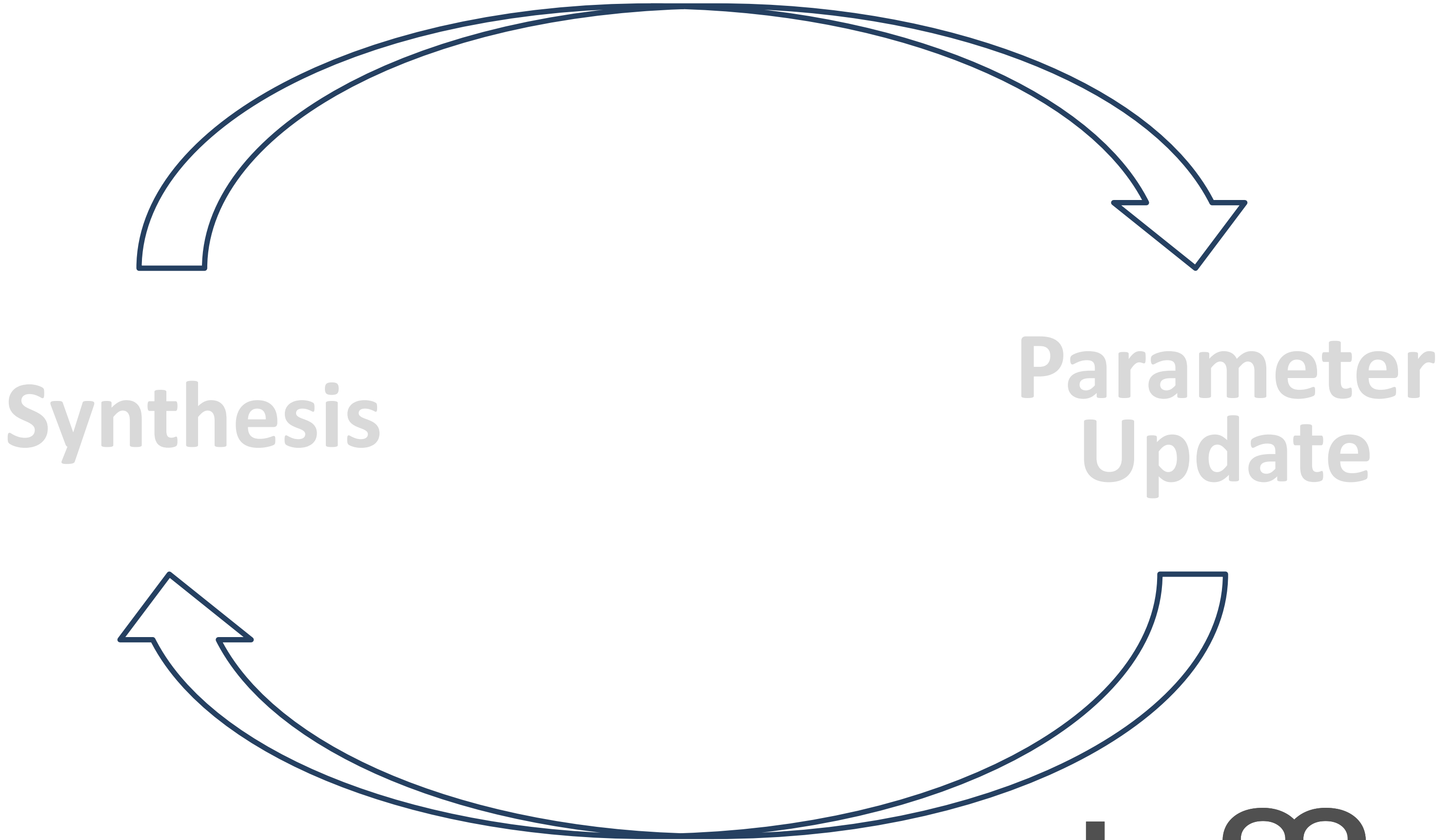
shape

albedo

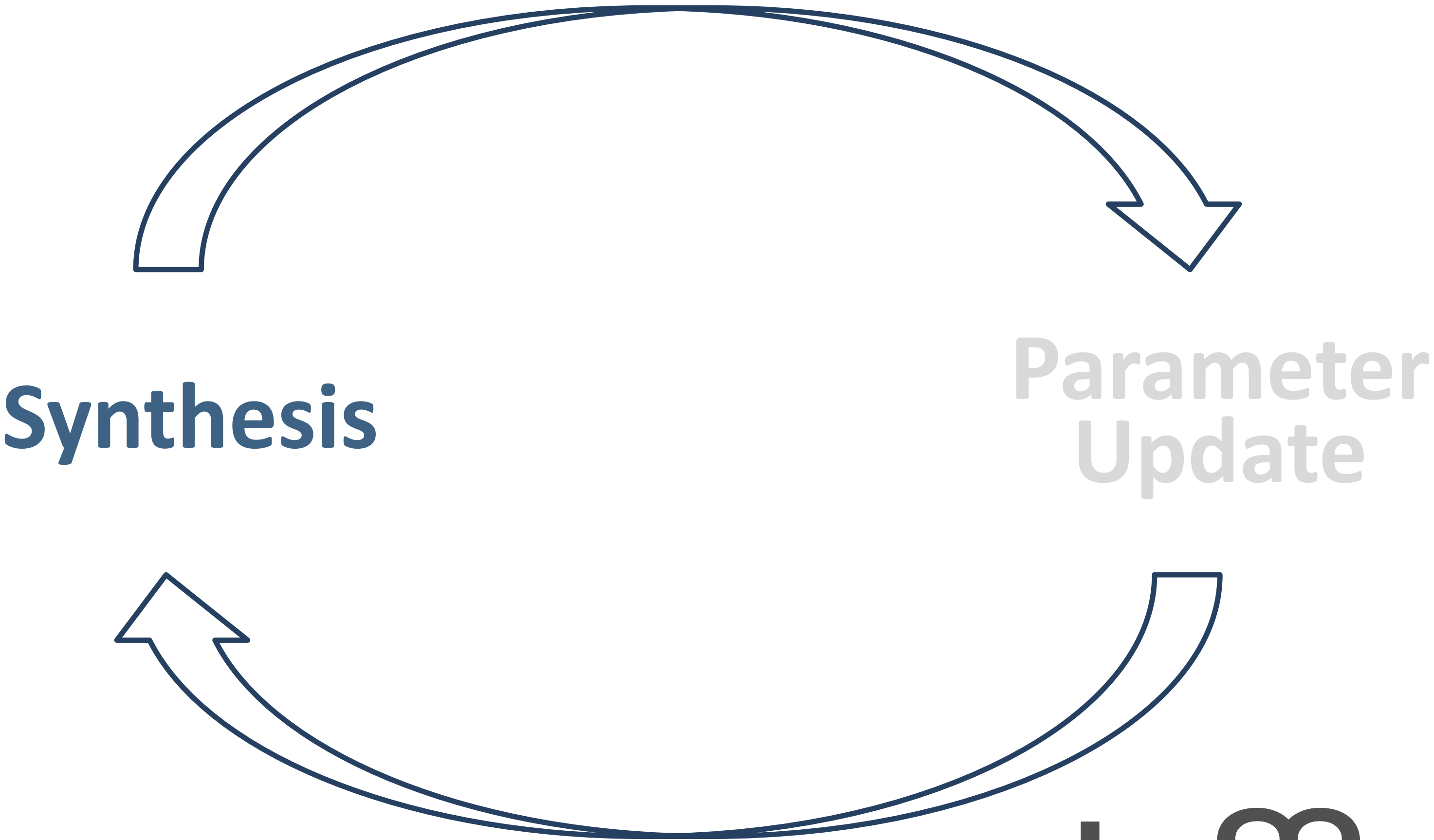
illumination

expression

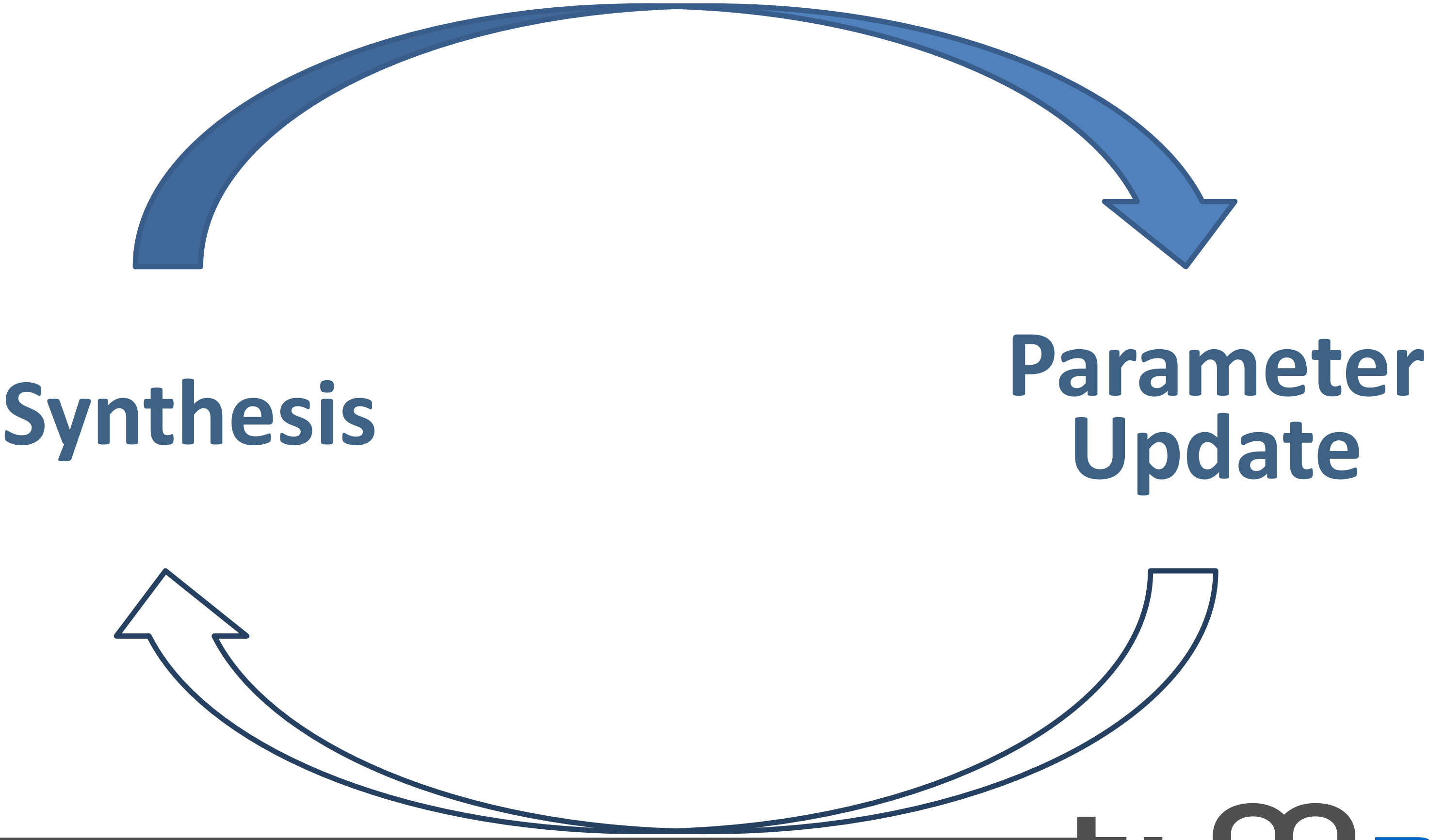
Analysis-by-Synthesis



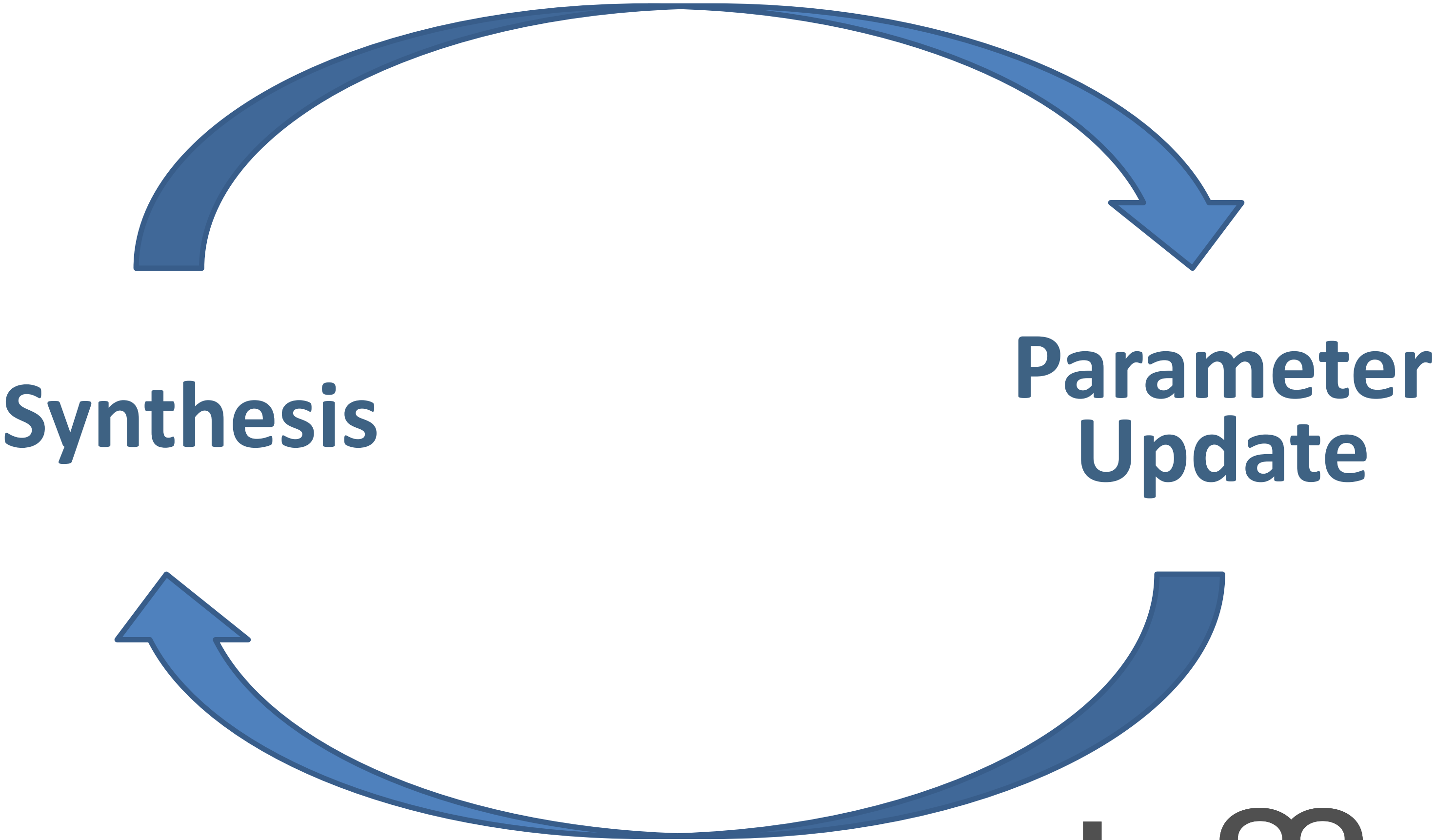
Analysis-by-Synthesis



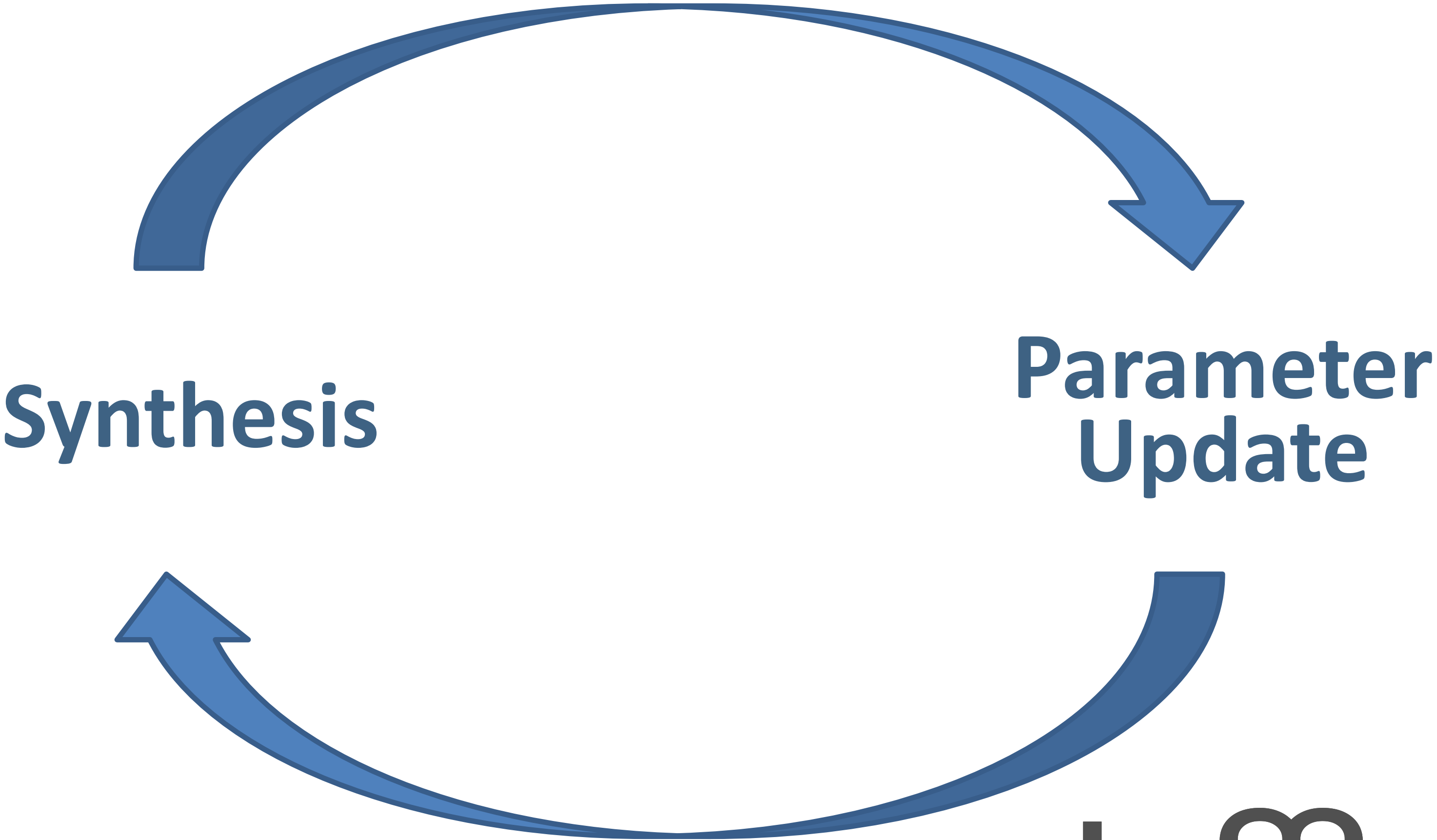
Analysis-by-Synthesis



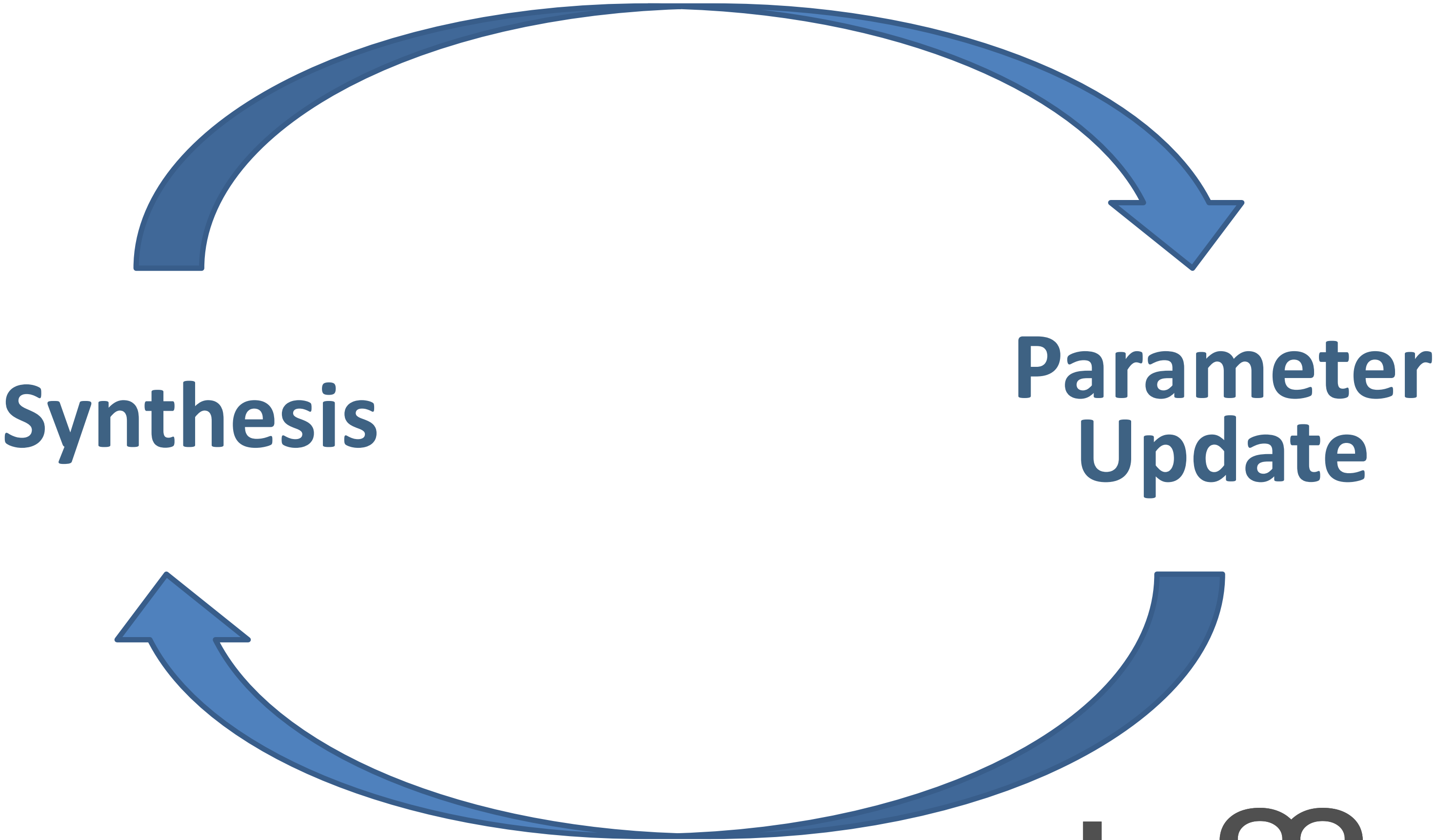
Analysis-by-Synthesis



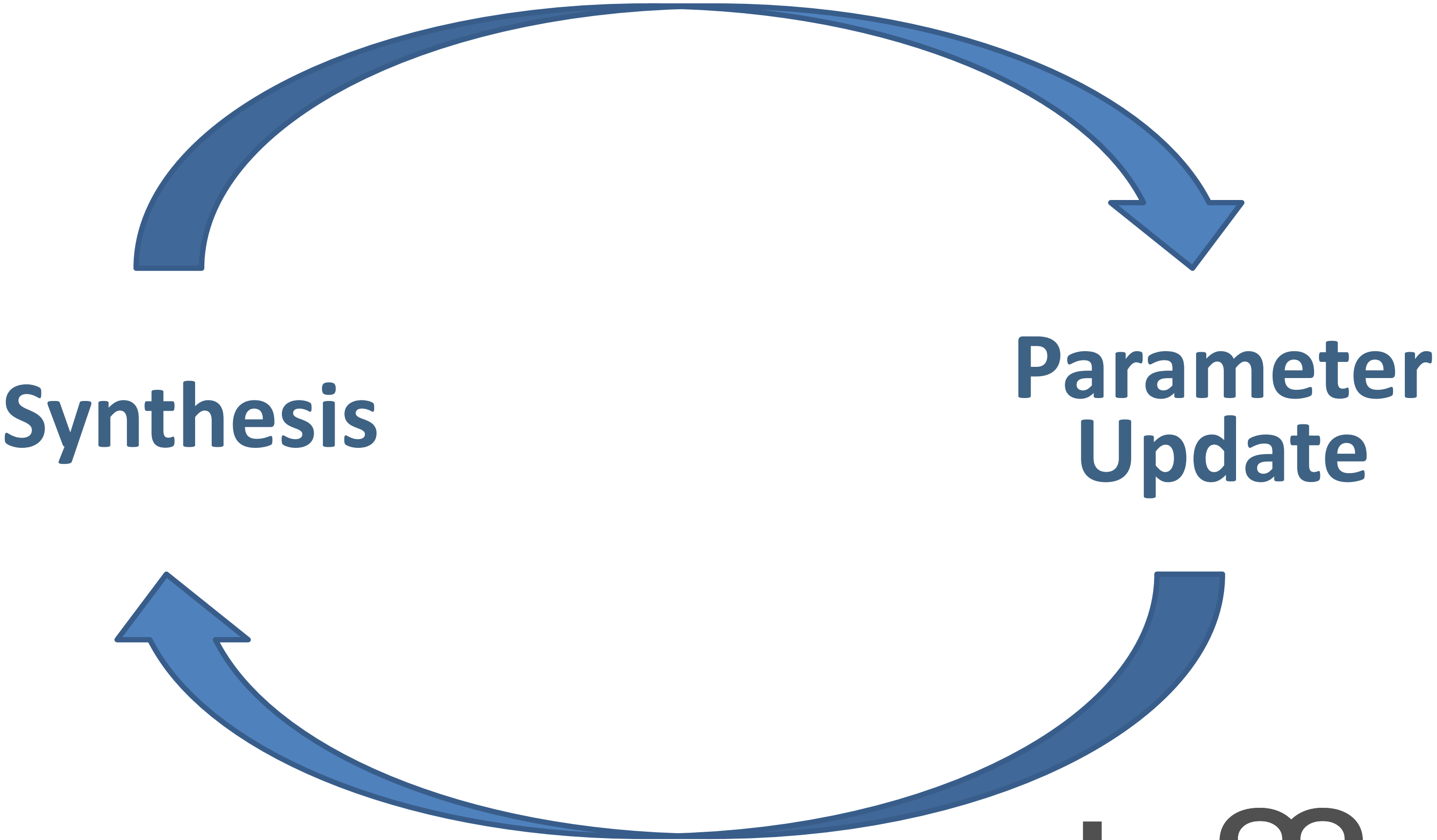
Analysis-by-Synthesis



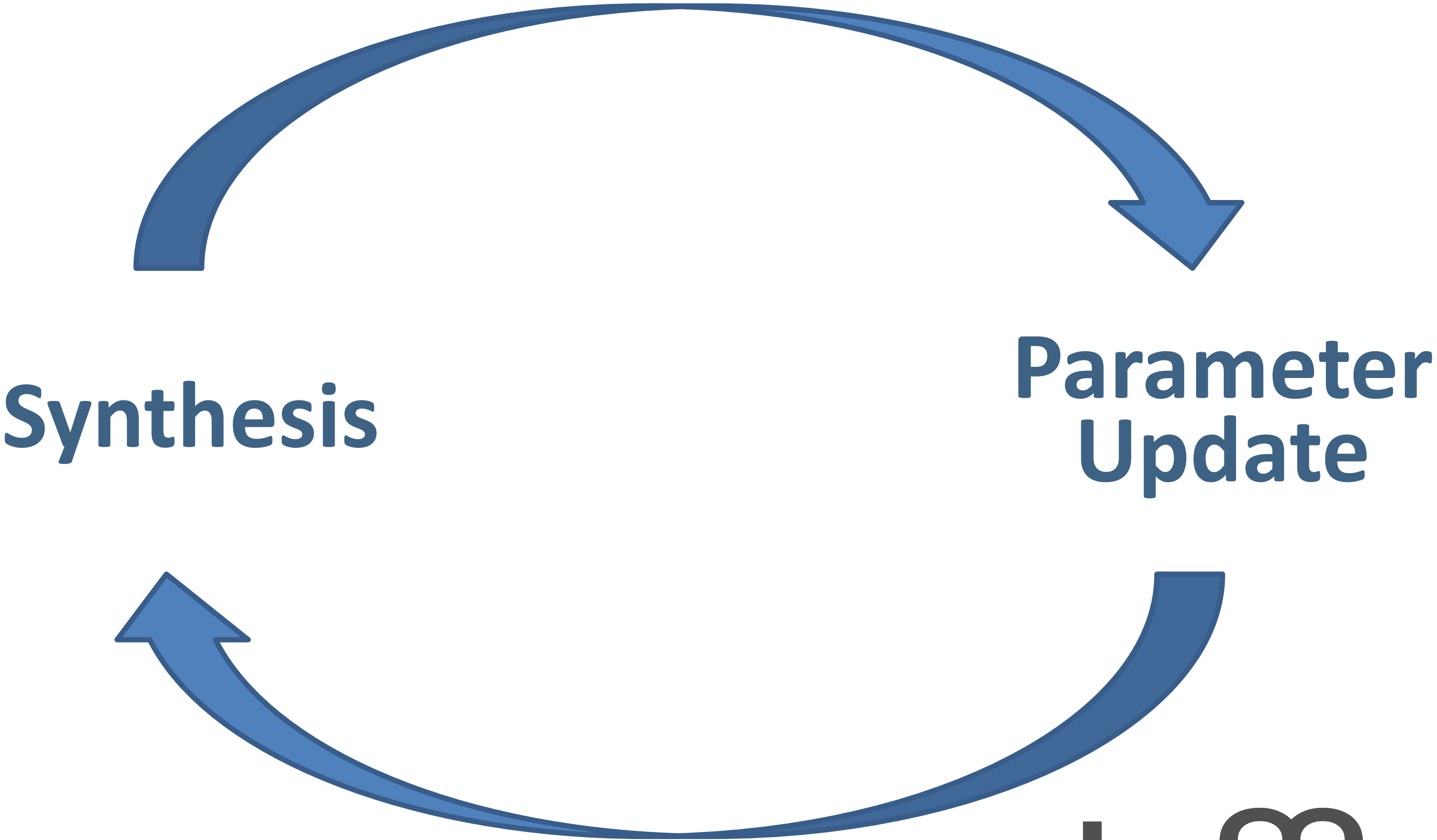
Analysis-by-Synthesis



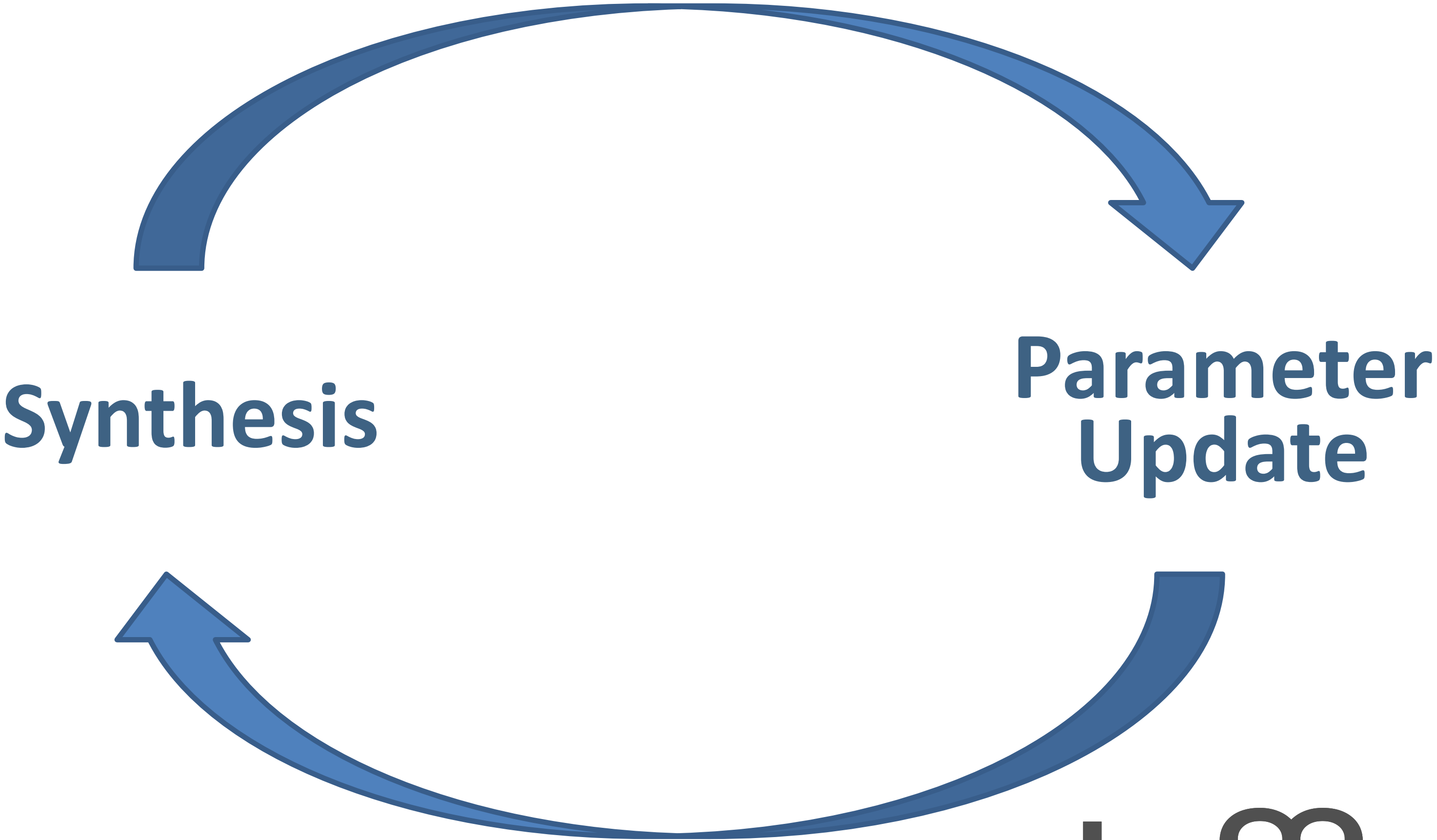
Analysis-by-Synthesis



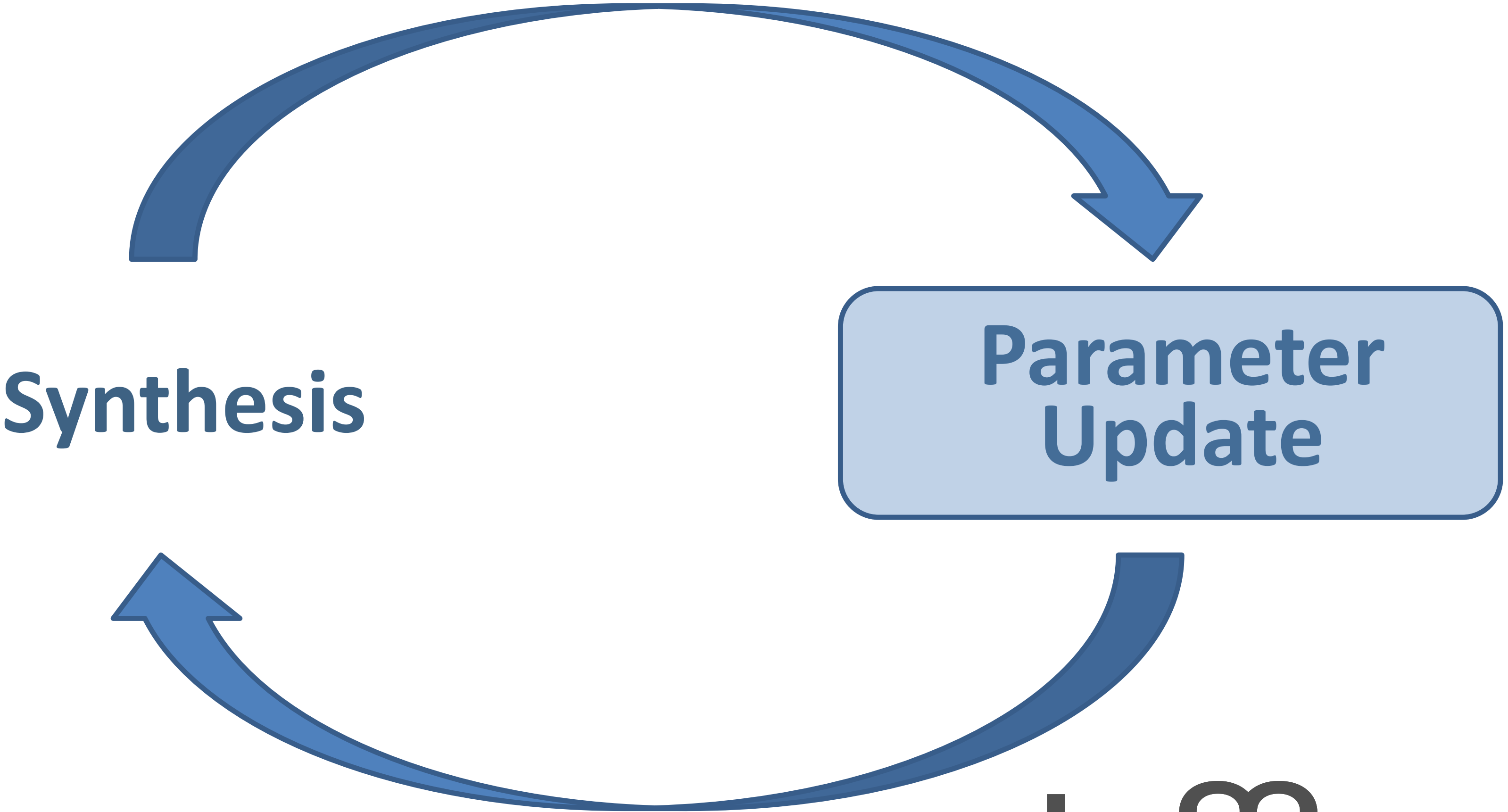
Analysis-by-Synthesis



Analysis-by-Synthesis



Analysis-by-Synthesis

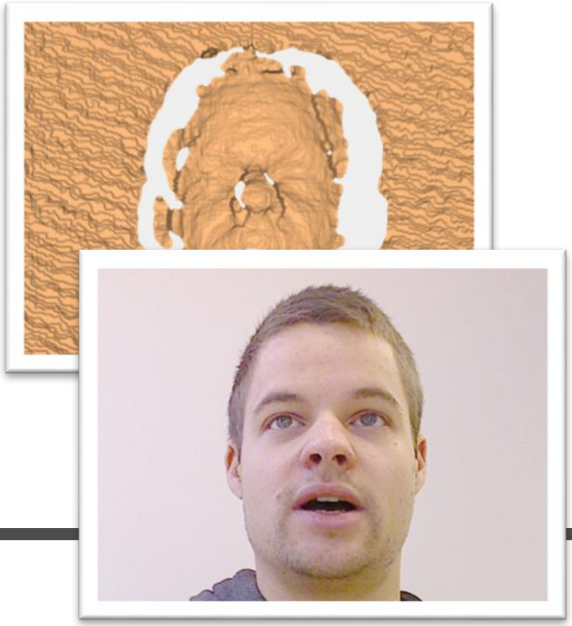


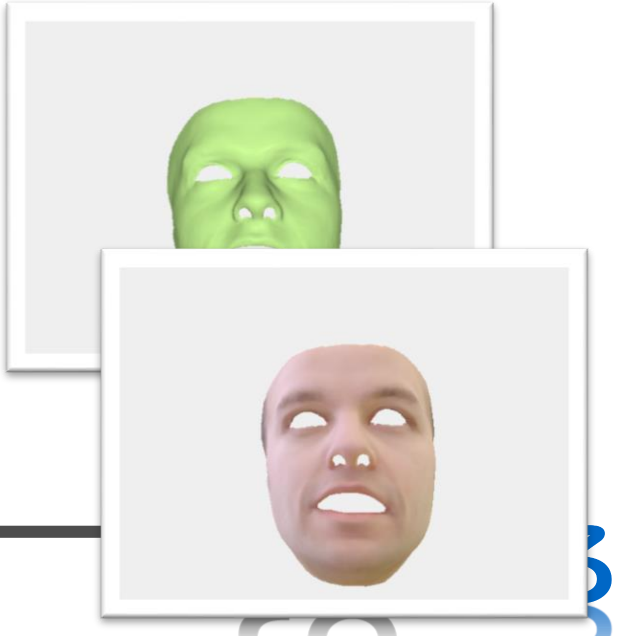
Parameter Estimation as Energy Minimization

Parameter Estimation as Energy Minimization

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

pose
shape
albedo
illumination
expression

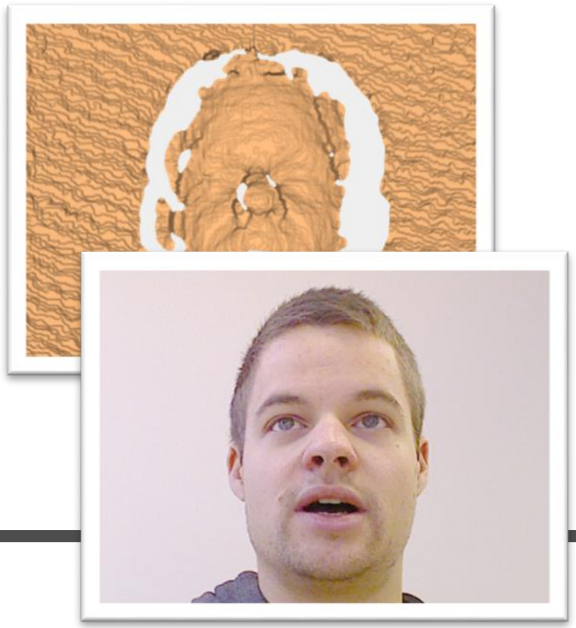




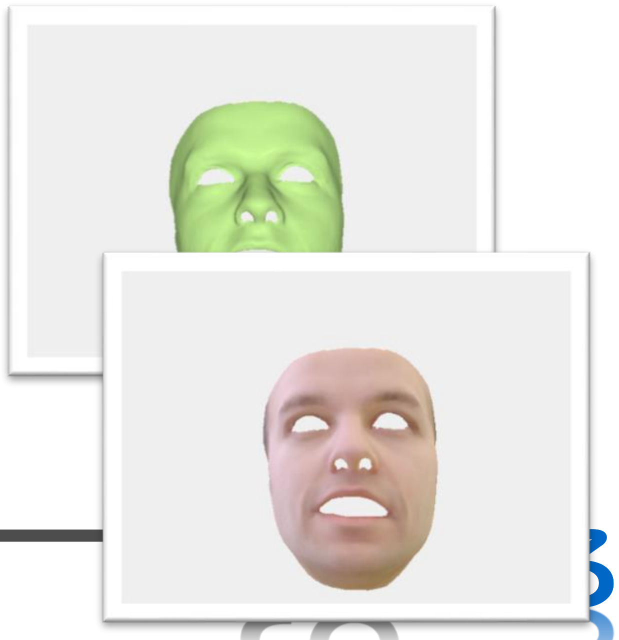
Parameter Estimation as Energy Minimization

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

$E_{dense}(P) =$

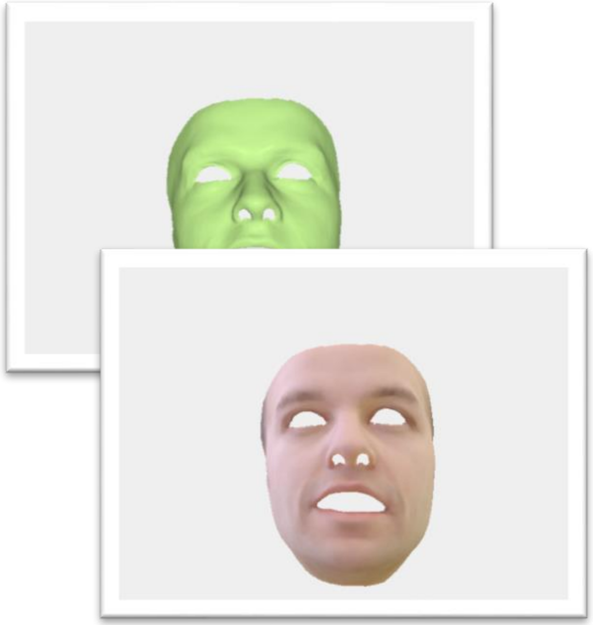


$-$



^{2}

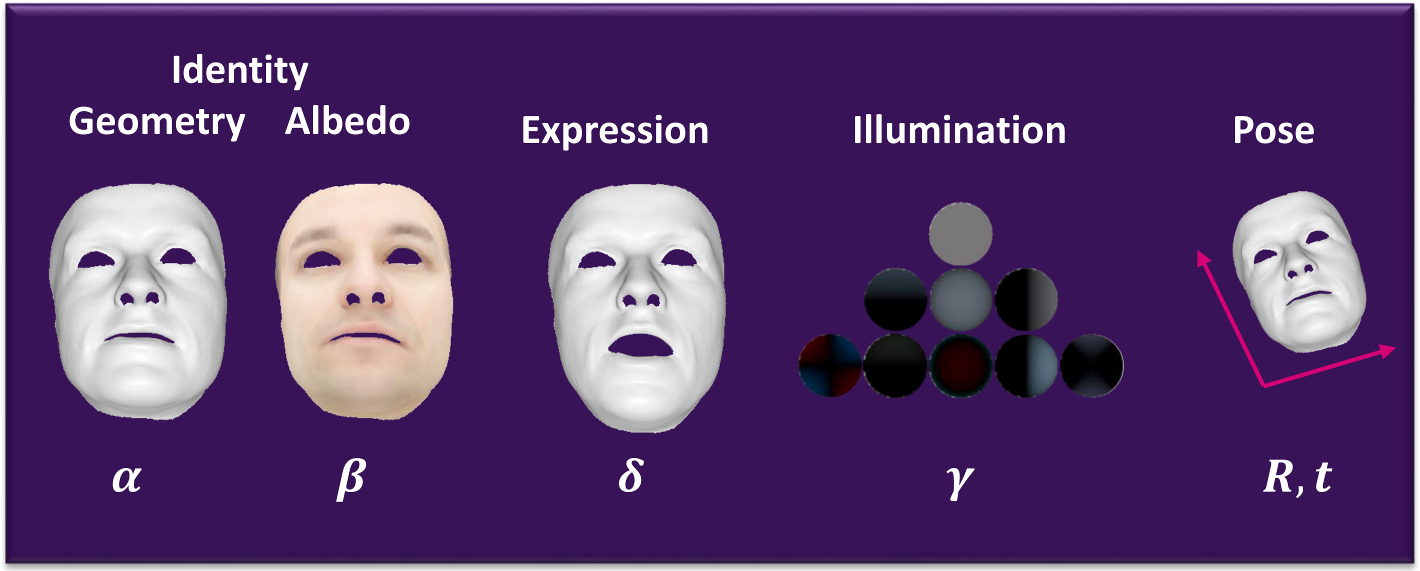
Parameter Estimation as Energy Minimization

$E_{dense}(P) = \Big|$  $-$  $\Big|^{2}$

$$P^{*} = argmin_{P}\, E_{dense}(P)$$

Parameter Estimation as Energy Minimization

$$P =$$



Parameter Estimation as Energy Minimization

$$E(P) = \underbrace{E_{dense}(P) + E_{sparse}(P)}_{\textit{data}} + \underbrace{E_{reg}(P)}_{\textit{prior}}$$

Parameter Estimation as Energy Minimization

$$E(P) = \underbrace{E_{dense}(P) + E_{sparse}(P)}_{data} + \underbrace{E_{reg}(P)}_{prior}$$

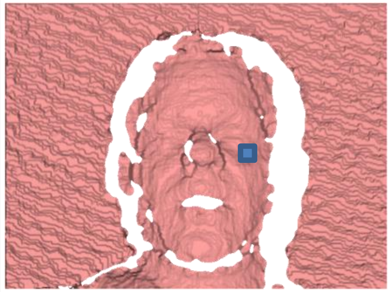
Parameter Estimation as Energy Minimization

$$E(P) = \underbrace{E_{dense}(P) + E_{sparse}(P)}_{data} + \underbrace{E_{reg}(P)}_{prior}$$

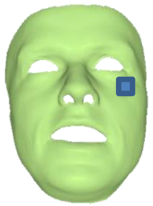
$E_{dense}(P) = \sum_{pixel}$ $\Big|$

$-$

$\Big|^2$ $+$ $\Big|$



$-$

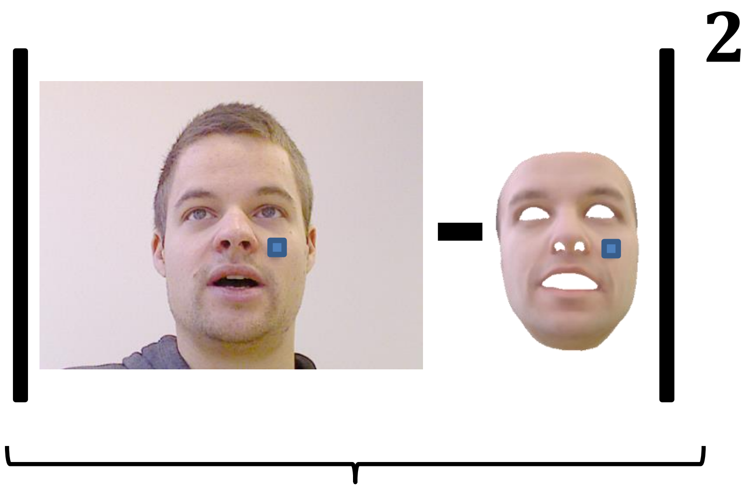


$\Big|^2$

measurement in color space

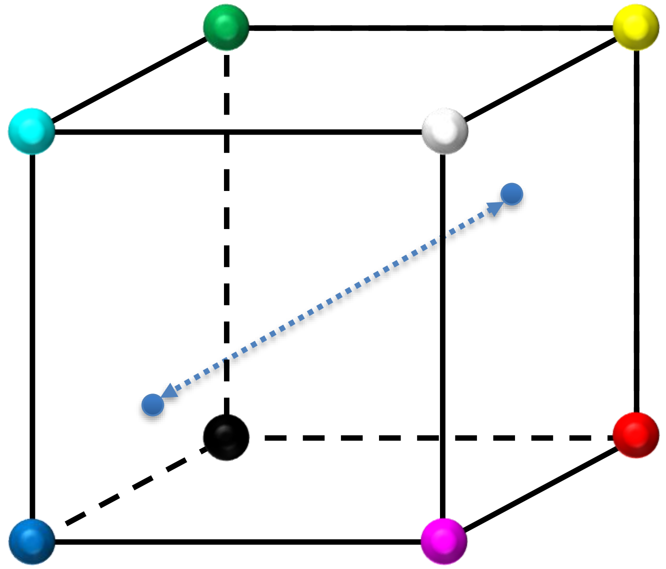
measurement in world space

Parameter Estimation as Energy Minimization

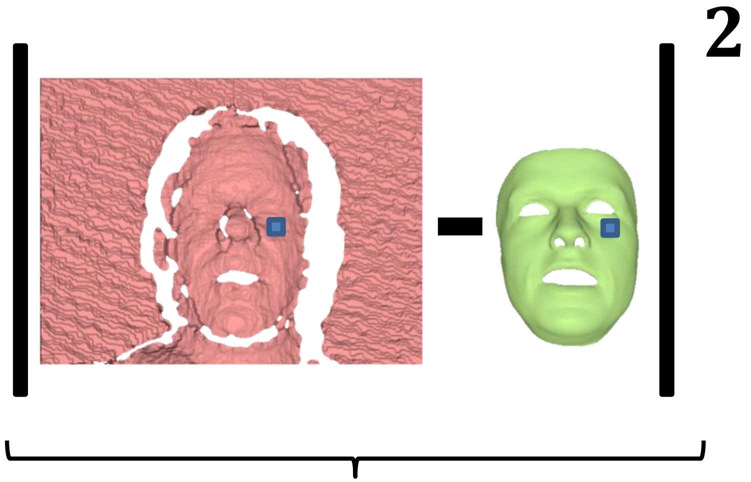


measurement in color space

distance in RGB space

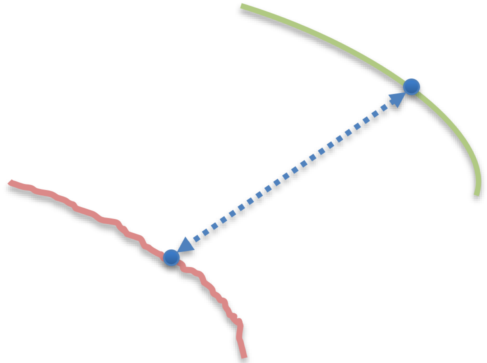


Parameter Estimation as Energy Minimization

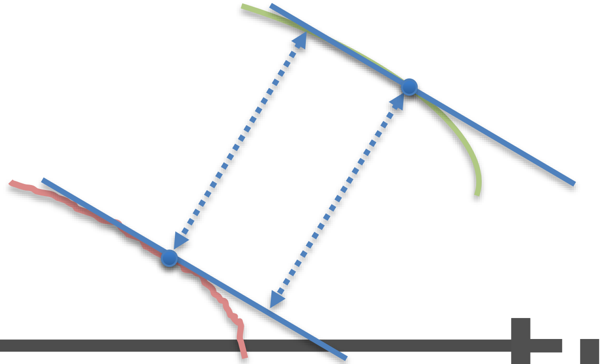


measurement in world space

point-to-point distance

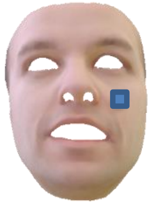
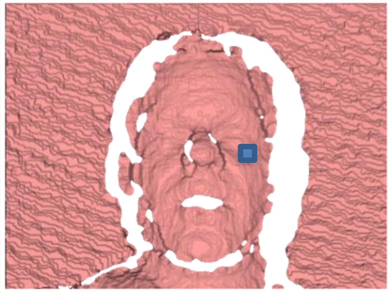
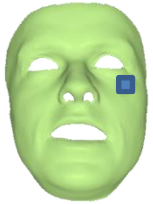


point-to-plane distance



Parameter Estimation as Energy Minimization

$$E(P) = \underbrace{E_{dense}(P) + E_{sparse}(P)}_{data} + \underbrace{E_{reg}(P)}_{prior}$$

$E_{dense}(P) = \sum_{pixel}$ $\Big|$  $-$  $\Big|^2$ $+$ $\Big|$  $-$  $\Big|^2$

measurement in color space

measurement in world space

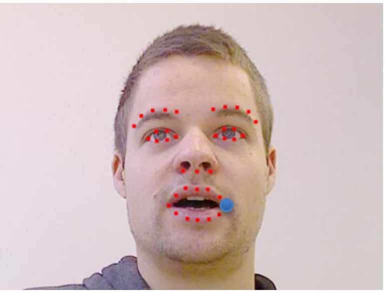
Parameter Estimation as Energy Minimization

$$E(P) = \underbrace{E_{dense}(P) + E_{sparse}(P)}_{data} + \underbrace{E_{reg}(P)}_{prior}$$

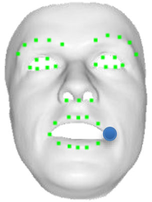
Parameter Estimation as Energy Minimization

$$E(P) = \underbrace{E_{dense}(P) + E_{sparse}(P)}_{data} + \underbrace{E_{reg}(P)}_{prior}$$

$E_{sparse}(P) = \sum_{feature}$ $\Big|$

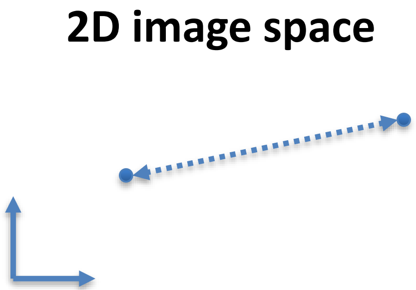


$-$



$\Big|^{2}$

measurement in image space



Parameter Estimation as Energy Minimization

$$E(P) = \underbrace{E_{dense}(P) + E_{sparse}(P)}_{data} + \underbrace{E_{reg}(P)}_{prior}$$

Parameter Estimation as Energy Minimization

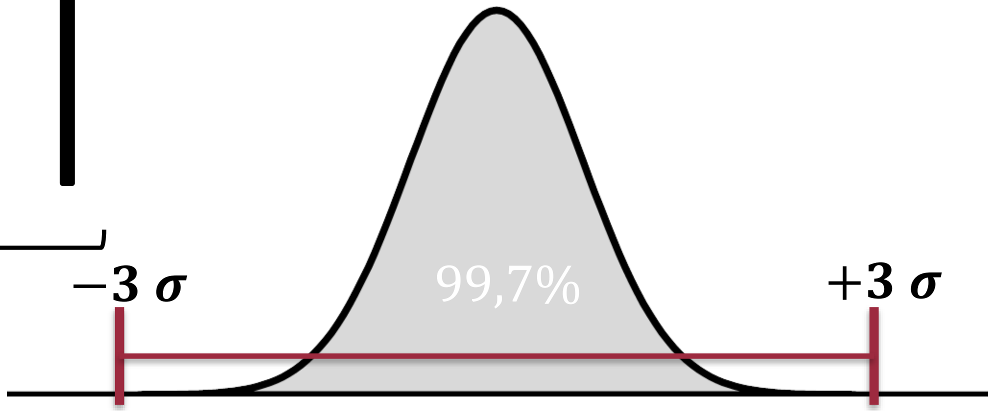
$$E(P) = \underbrace{E_{dense}(P) + E_{sparse}(P)}_{data} + \underbrace{E_{reg}(P)}_{prior}$$

$$E_{reg}(P) = \sum \underbrace{\left| \frac{P_i}{\sigma_{P_i}} \right|^2}_{\textit{measurement in parameter space}}$$

Parameter Estimation as Energy Minimization

$$E(P) = \underbrace{E_{dense}(P) + E_{sparse}(P)}_{\textit{data}} + \underbrace{E_{reg}(P)}_{\textit{prior}}$$

$$E_{reg}(P) = \sum \underbrace{\left| \frac{P_i}{\sigma_{P_i}} \right|^{2}}_{\substack{\textit{measurement in} \\ \textit{parameter space}}}$$



Parameter Estimation as Energy Minimization

$$E(P) = \underbrace{E_{dense}(P) + E_{sparse}(P)}_{\textit{data}} + \underbrace{E_{reg}(P)}_{\textit{prior}}$$

Parameter Estimation as Energy Minimization

- Given model $M_i(P)$ (i-th vertex)
- $E_{geom}(P) = \sum_i \underbrace{||M_i(P) - D(\pi(M_i(P))||_2^2}_{\text{Point-to-point}} + \underbrace{[(M_i(P) - D(\pi(M_i(P))) \cdot n_i]^2}_{\text{Point-to-plane}}$
- $E_{col}(P) = \sum_i ||M_i(P)_c - I(\pi(M_i(P)))||_2^2$

  color at surface point i
- $E_{mrk}(P) = \sum_i ||\underbrace{\pi(M_i(P)_m)}_{\text{2D location of projected marker pos.}} - C_i||_2^2$
- $E_{reg}(P) = \sum_i \left[\left(\frac{\alpha_i}{\sigma_{id,i}}\right)^2 + \left(\frac{\beta_i}{\sigma_{alb,i}}\right)^2\right] + \sum_i \left(\frac{\delta_i}{\sigma_{exp,i}}\right)^2$

Std dev. $\alpha, \beta, \gamma \in P$ are id, albedo, and expr. params

Parameter Estimation as Energy Minimization

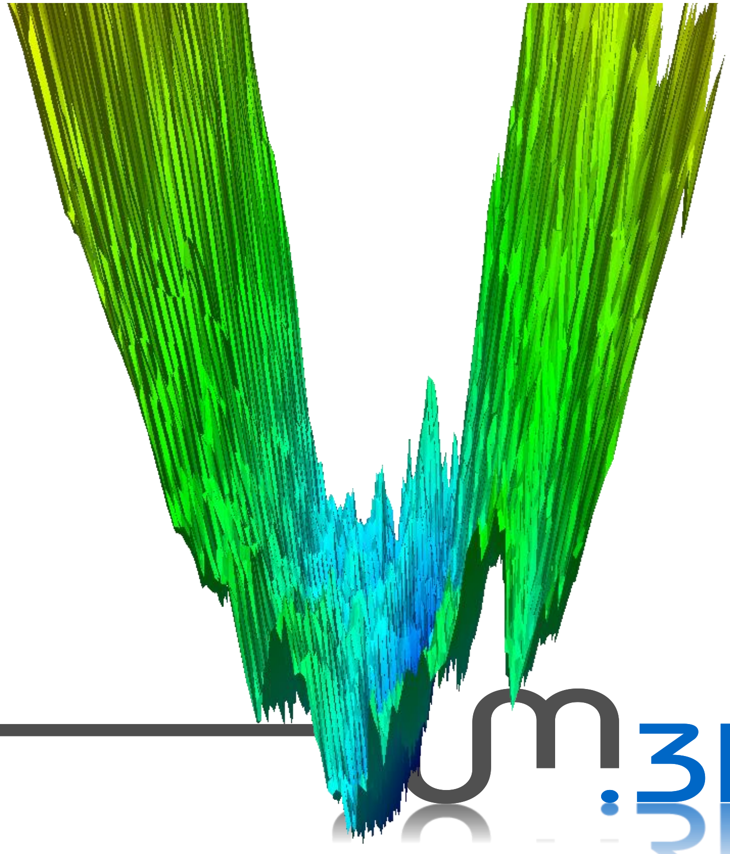
$$E(P) = \underbrace{E_{dense}(P) + E_{sparse}(P)}_{data} + \underbrace{E_{reg}(P)}_{prior}$$

$$P^* = argmin_P\, E(P)$$

Parameter Estimation as Energy Minimization

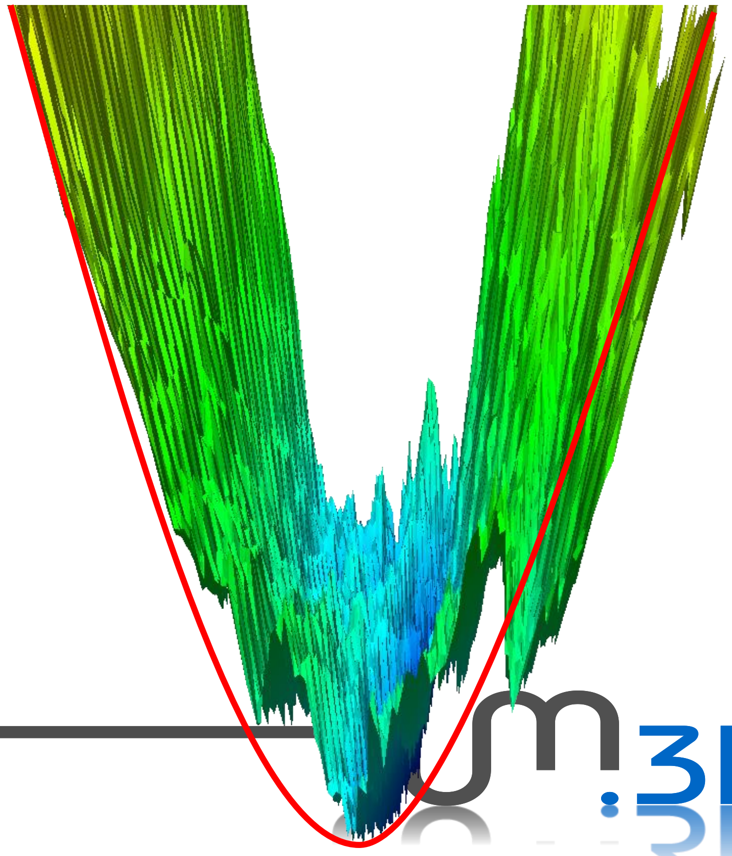
$$E(P) = \underbrace{E_{dense}(P) + E_{sparse}(P)}_{data} + \underbrace{E_{reg}(P)}_{prior}$$

$$P^* = argmin_P\, E(P)$$



Parameter Estimation as Energy Minimization

$$E(P) = \underbrace{E_{dense}(P) + E_{sparse}(P)}_{data} + \underbrace{E_{reg}(P)}_{prior}$$

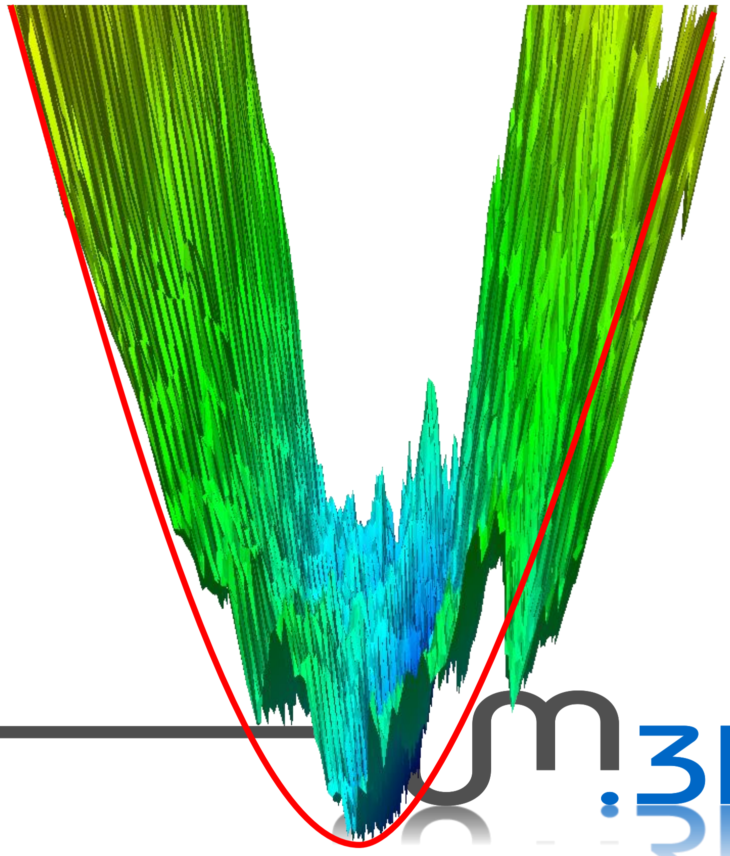


Non-linear Least Squares

$$E\,(P) = |F\,(P)|_2^2$$

Parameter Estimation as Energy Minimization

$$E(P) = \underbrace{E_{dense}(P) + E_{sparse}(P)}_{data} + \underbrace{E_{reg}(P)}_{prior}$$



Non-linear Least Squares

$$E(P) = |F(P)|_2^2$$

Gauss-Newton-Method

Parameter Estimation as Energy Minimization

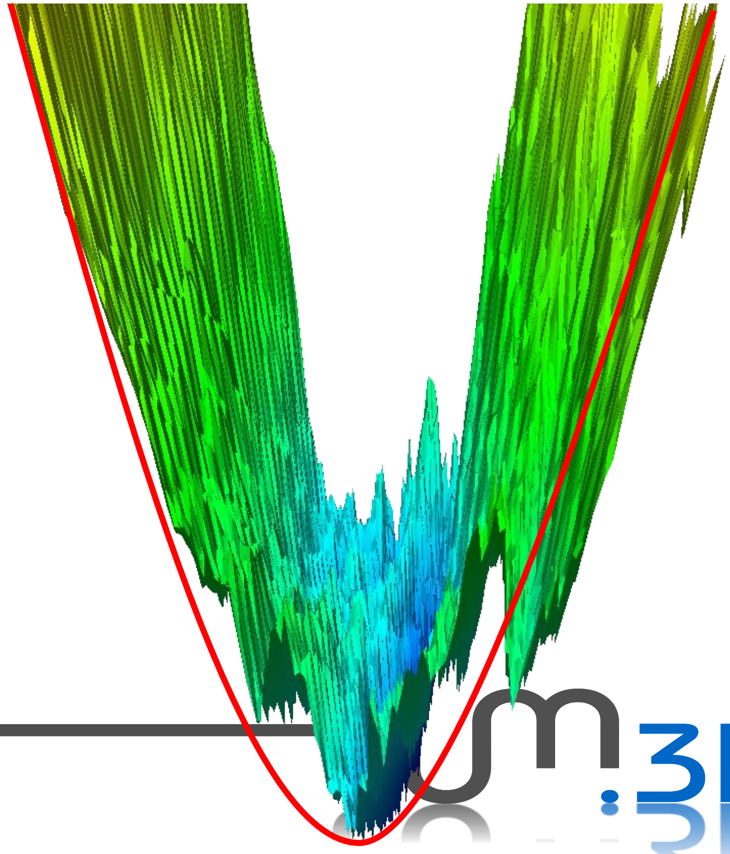
Gauss-Newton-Method

$$\boldsymbol{E(P) = |F(P)|_2^2}$$

Iterative Update Rule:

$$\boldsymbol{P_{n+1} = P_n - \underbrace{\left(J_F^T \cdot J_F\right)^{-1} \cdot J_F^T \cdot F}_{=\Delta}}$$

$$\Rightarrow \quad \boldsymbol{\left(J_F^T \cdot J_F\right) \cdot \Delta = J_F^T \cdot F}$$



Parameter Estimation as Energy Minimization

Gauss-Newton-Method

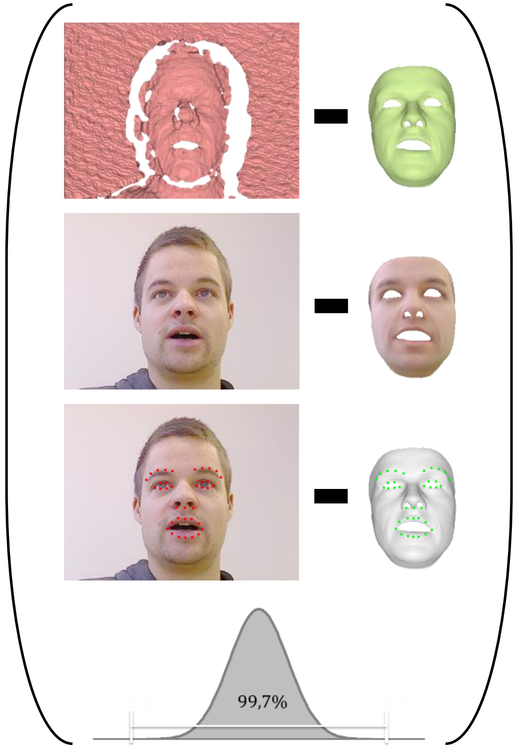
$$\boldsymbol{E(P) = |F(P)|_2^2}$$

Iterative Update Rule:

$$\boldsymbol{P_{n+1} = P_n - \underbrace{\left(J_F^T \cdot J_F\right)^{-1} \cdot J_F^T \cdot F}_{= \Delta}}$$

$$\Rightarrow \quad \boldsymbol{\left(J_F^T \cdot J_F\right) \cdot \Delta = J_F^T \cdot F}$$

$F(P) =$



$$\in \mathbb{R}^{>7 \cdot \#pixels}$$

Parameter Estimation as Energy Minimization

Gauss-Newton-Method

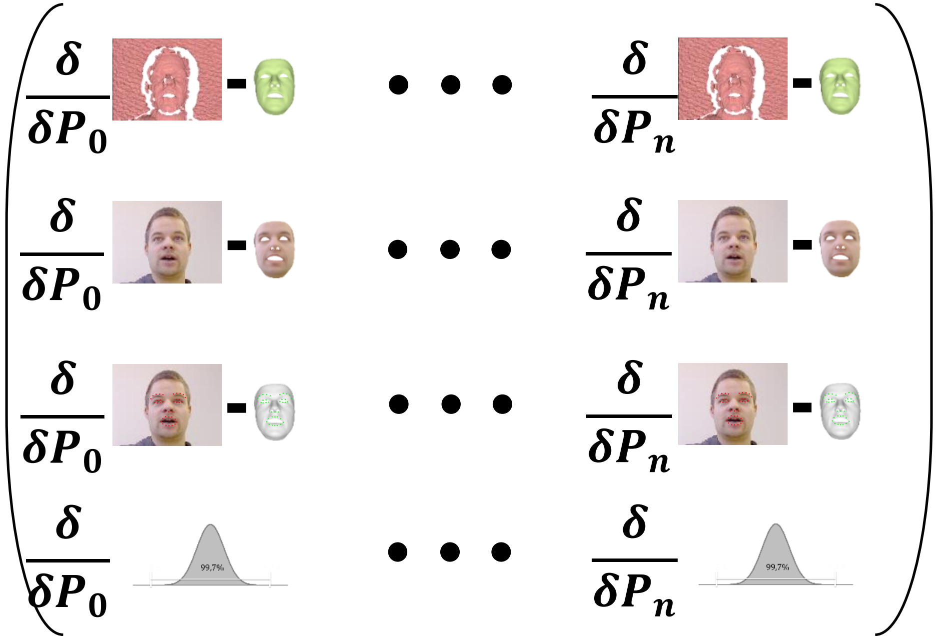
$$\boldsymbol{E(P) = |F(P)|_2^2}$$

Iterative Update Rule:

$$\boldsymbol{P_{n+1} = P_n - \underbrace{\left(J_F^T \cdot J_F\right)^{-1} \cdot J_F^T \cdot F}_{= \Delta}}$$

$$\Rightarrow \quad \boldsymbol{\left(J_F^T \cdot J_F\right) \cdot \Delta = J_F^T \cdot F}$$

$J_F(P) =$



Parameter Estimation as Energy Minimization

Gauss-Newton-Method

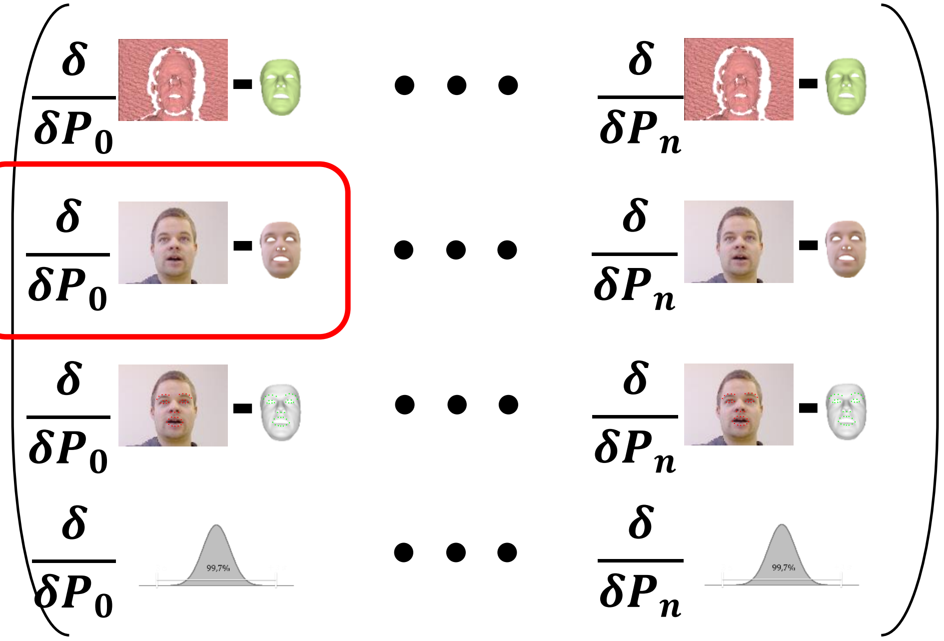
$$E(P) = |F(P)|_2^2$$

Iterative Update Rule:

$$P_{n+1} = P_n - \underbrace{\left(J_F^T \cdot J_F\right)^{-1} \cdot J_F^T \cdot F}_{= \Delta}$$

$$\Rightarrow \quad \left(J_F^T \cdot J_F\right) \cdot \Delta = J_F^T \cdot F$$

$J_F(P) =$



Parameter Estimation as Energy Minimization

Gauss-Newton-Method

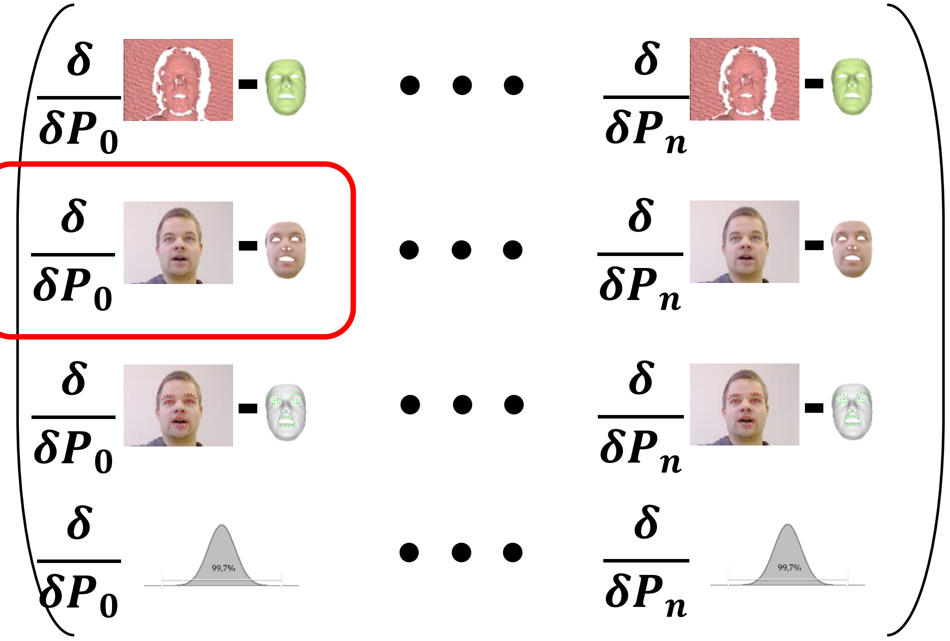
$$\boldsymbol{E(P) = |F(P)|_2^2}$$

Iterative Update Rule:

$$\boldsymbol{P_{n+1} = P_n - \underbrace{\left(J_F^T \cdot J_F\right)^{-1} \cdot J_F^T \cdot F}_{= \Delta}}$$

$$\Rightarrow \quad \boldsymbol{\left(J_F^T \cdot J_F\right) \cdot \Delta = J_F^T \cdot F}$$

$J_F(P) =$



requires partial derivatives of the synthetic rendering

Parameter Estimation as Energy Minimization

Gauss-Newton-Method

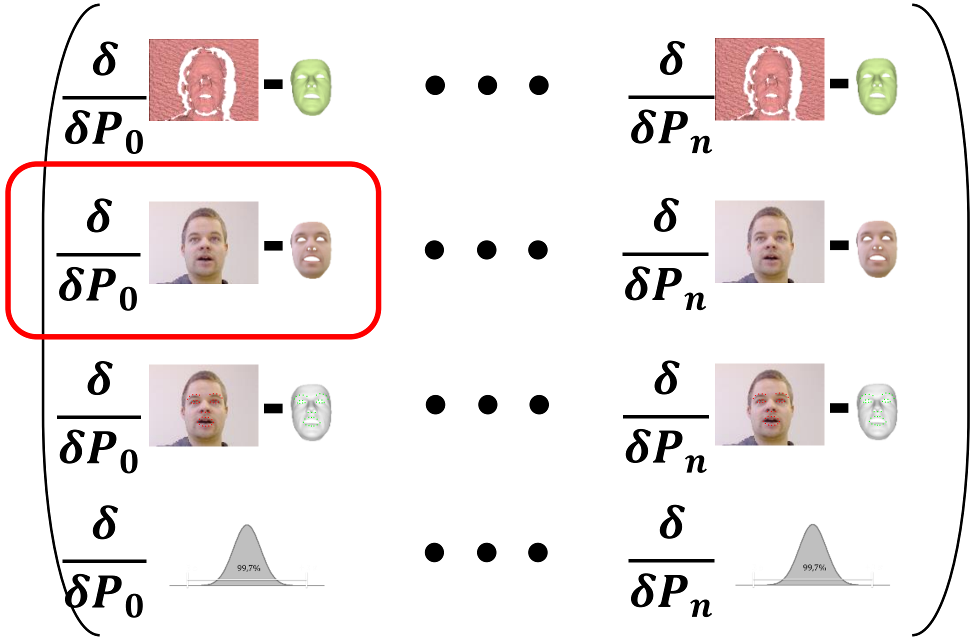
$$E(P) = |F(P)|_2^2$$

Iterative Update Rule:

$$P_{n+1} = P_n - \underbrace{\left(J_F^T \cdot J_F\right)^{-1} \cdot J_F^T \cdot F}_{= \Delta}$$

$$\Rightarrow \quad \left(J_F^T \cdot J_F\right) \cdot \Delta = J_F^T \cdot F$$

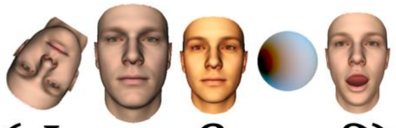
$J_F(P) =$



requires partial derivatives of the synthetic rendering

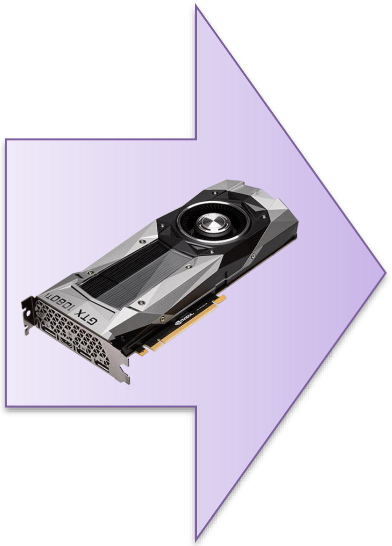
→ Differentiable Rendering

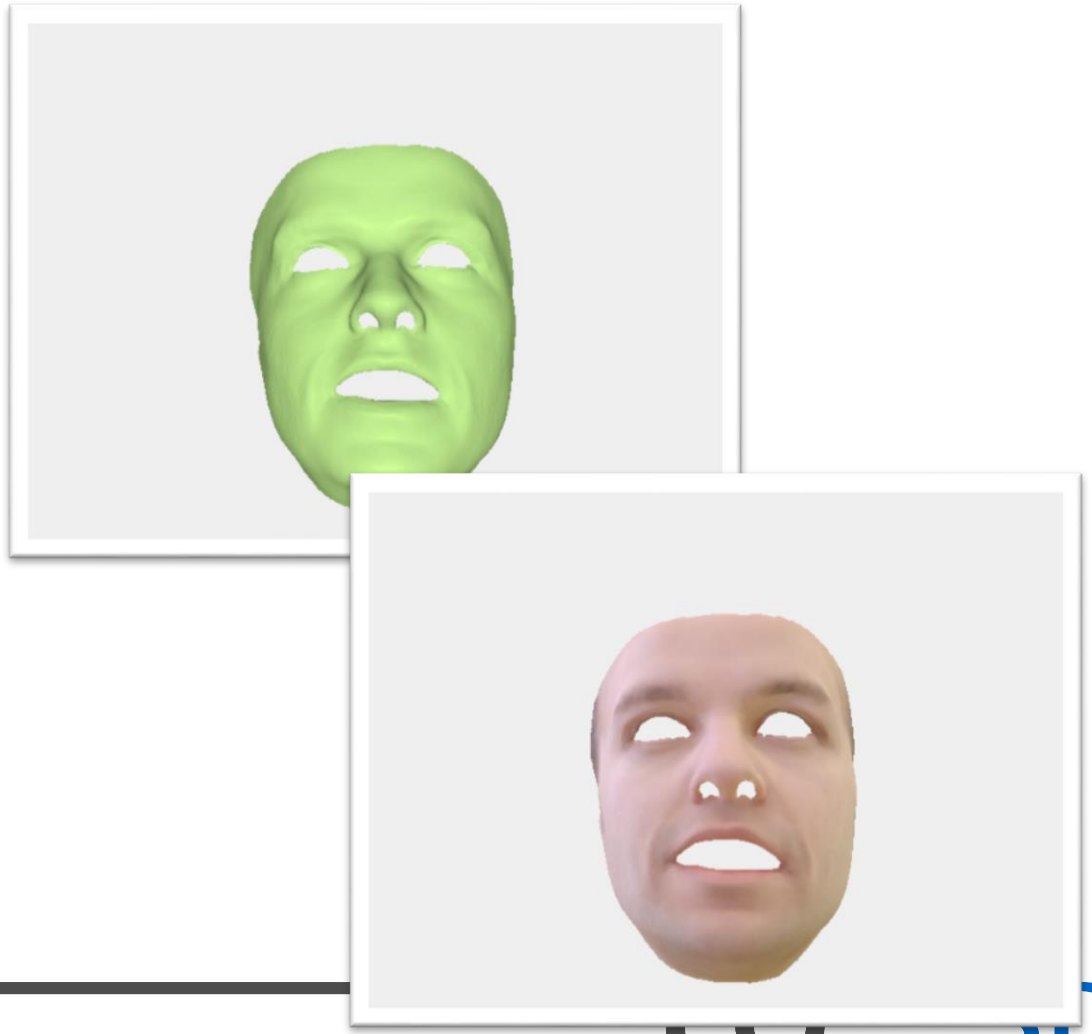
Differentiable Rendering



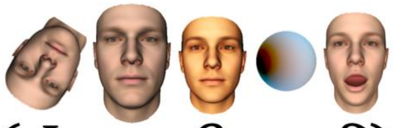
$$P = (\Phi, \alpha, \beta, \gamma, \delta)$$

pose
shape
albedo
illumination
expression



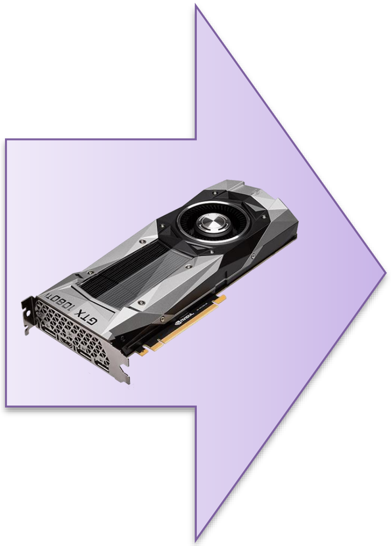


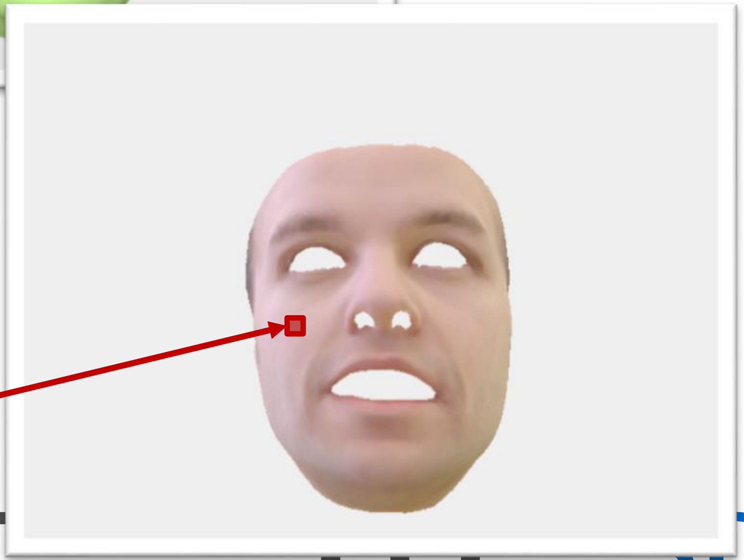
Differentiable Rendering



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

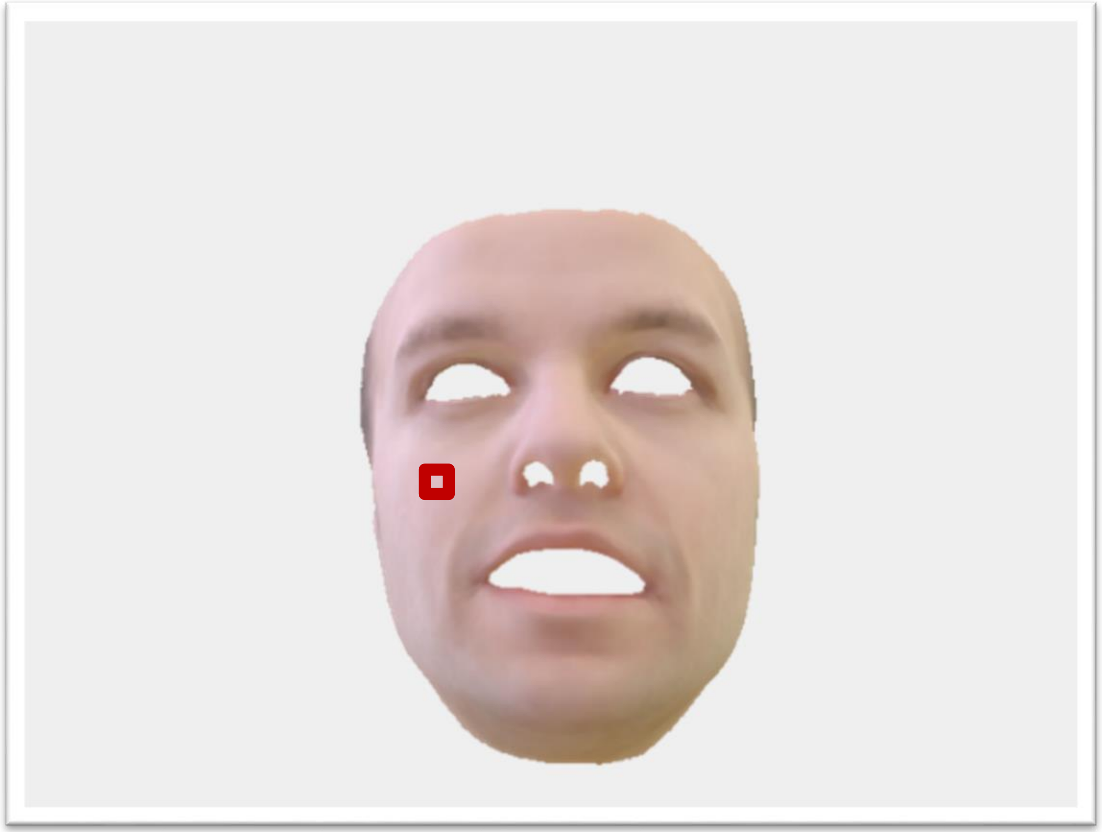
pose
shape
albedo
illumination
expression



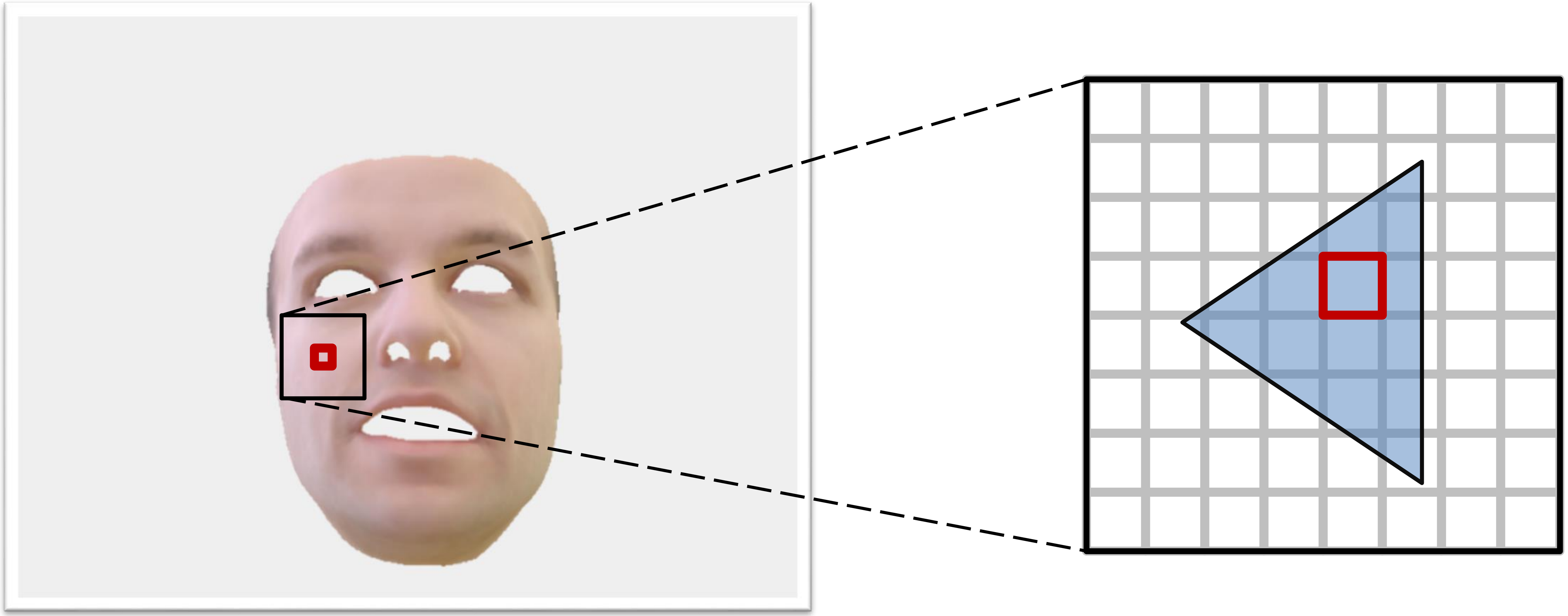


How to compute the partial derivatives of a pixel?

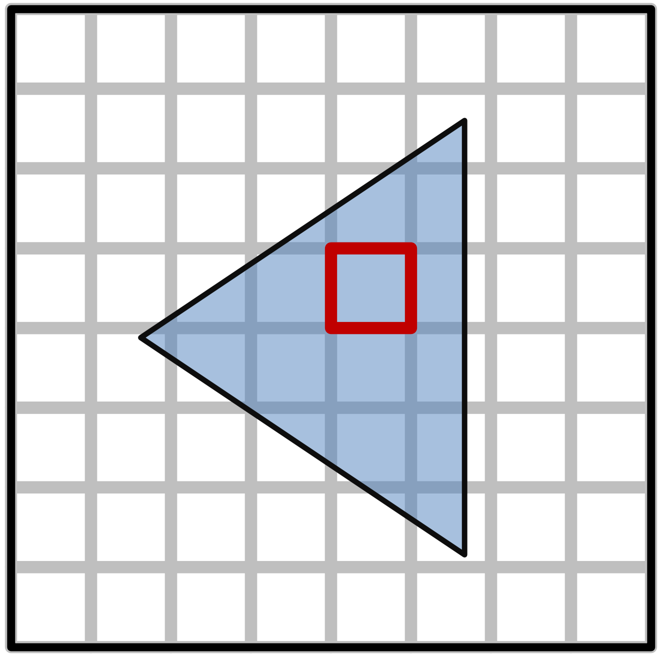
Differentiable Rendering



Differentiable Rendering

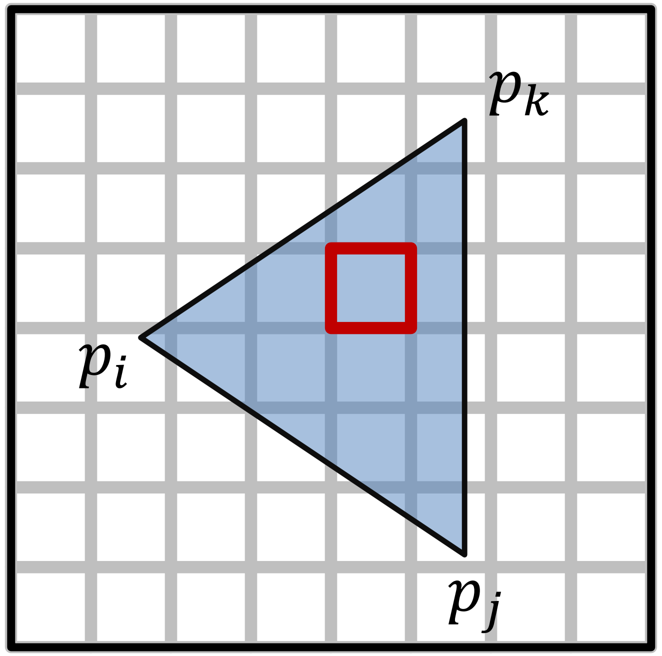


Differentiable Rendering



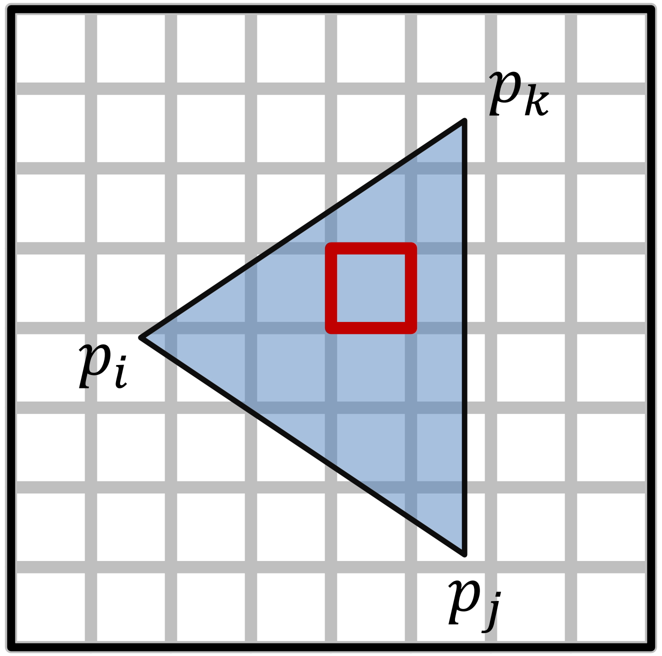
$C_{pixel} =$

Differentiable Rendering



$$C_{pixel} = \alpha \cdot C_{p_i} + \beta \cdot C_{p_j} + \gamma \cdot C_{p_k}$$

Differentiable Rendering

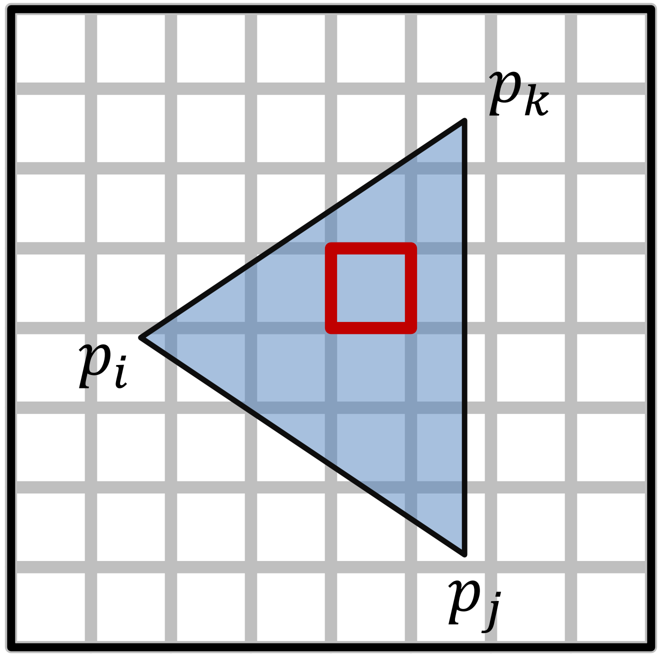


$$C_{pixel} = \alpha \cdot C_{p_i} + \beta \cdot C_{p_j} + \gamma \cdot C_{p_k}$$

Linear combination of vertex colors

→ Barycentric coordinates

Differentiable Rendering



$$C_{pixel} = \alpha \cdot C_{p_i} + \beta \cdot C_{p_j} + \gamma \cdot C_{p_k}$$

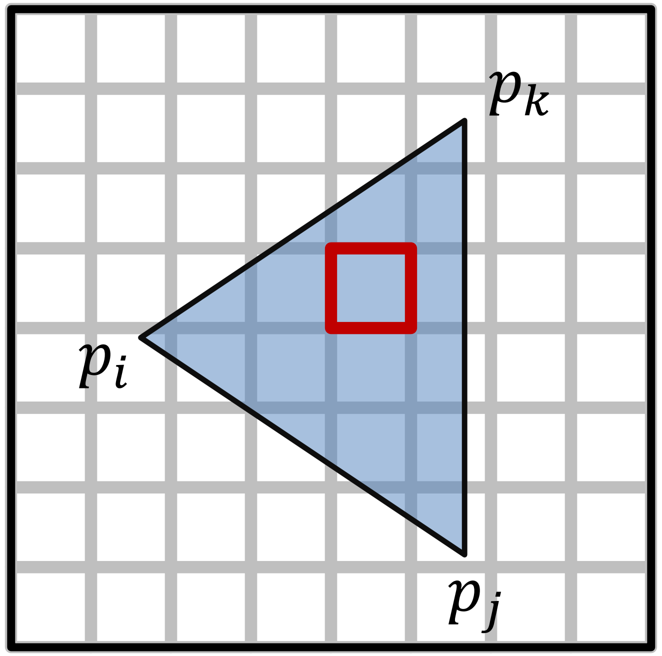
Linear combination of vertex colors

→ **Barycentric coordinates**

Vertex indices

→ **Triangle index**

Differentiable Rendering



$$C_{pixel} = \alpha \cdot C_{p_i} + \beta \cdot C_{p_j} + \gamma \cdot C_{p_k}$$

Linear combination of vertex colors

→ **Barycentric coordinates**

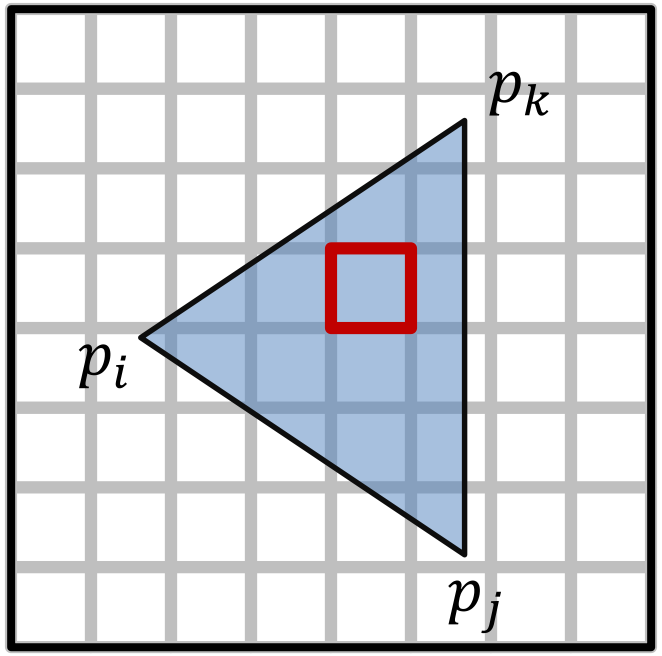
Vertex indices

→ **Triangle index**

Store the information during the forward rendering

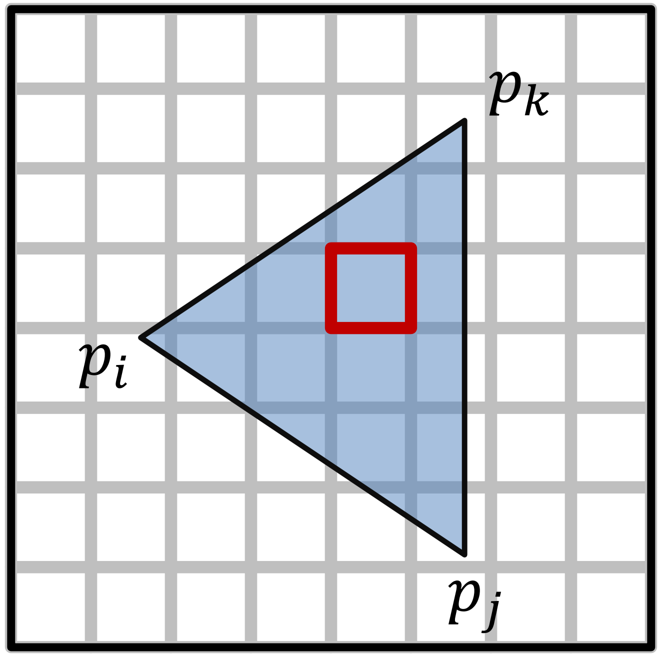
→ **Use additional render targets**

Differentiable Rendering



$$C_{pixel} = \alpha \cdot C_{p_i} + \beta \cdot C_{p_j} + \gamma \cdot C_{p_k}$$

Differentiable Rendering

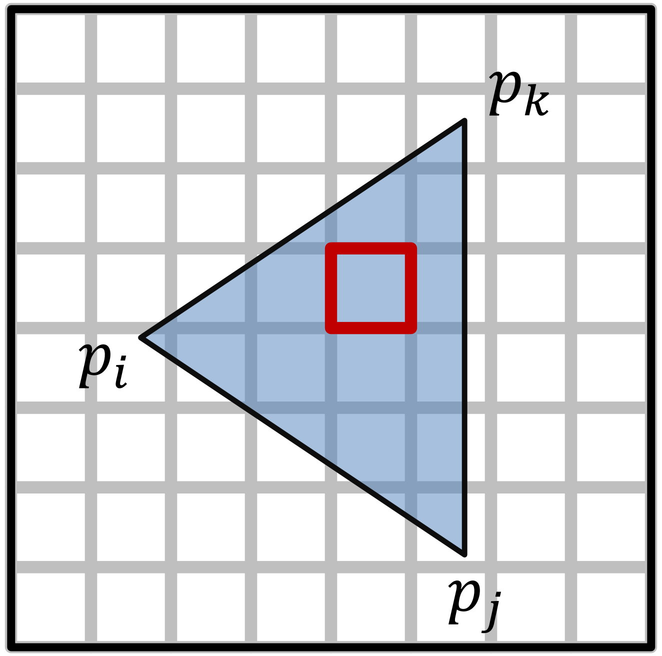


$$C_{pixel}(P) = \alpha \cdot C_{p_i}(P) + \beta \cdot C_{p_j}(P) + \gamma \cdot C_{p_k}(P)$$

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

pose
shape
albedo
illumination
expression

Differentiable Rendering



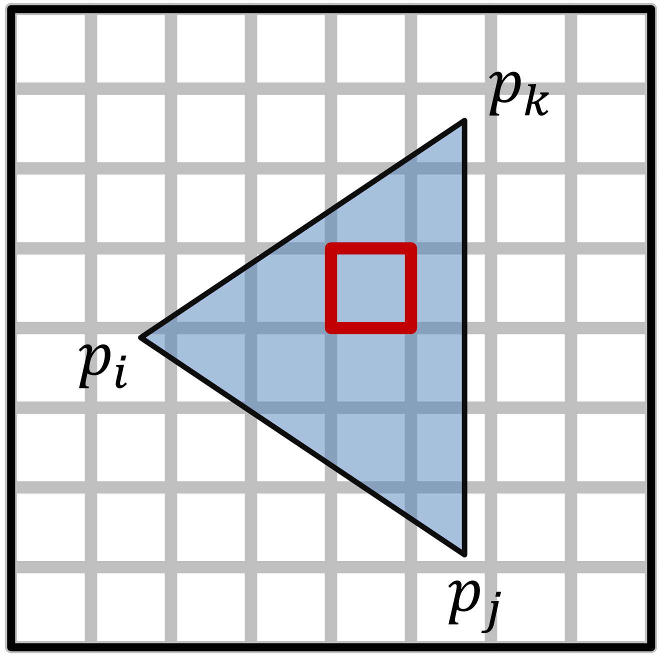
$$C_{pixel}(P) = \alpha \cdot C_{p_i}(P) + \beta \cdot C_{p_j}(P) + \gamma \cdot C_{p_k}(P)$$

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

pose
shape
albedo
illumination
expression

$$\frac{\partial C_{pixel}}{\partial P_i}(P) = \alpha \cdot \frac{\partial C_{p_i}}{\partial P_i}(P) + \beta \cdot \frac{\partial C_{p_j}}{\partial P_i}(P) + \gamma \cdot \frac{\partial C_{p_k}}{\partial P_i}(P)$$

Differentiable Rendering



$$C_{pixel}(P) = \alpha \cdot C_{p_i}(P) + \beta \cdot C_{p_j}(P) + \gamma \cdot C_{p_k}(P)$$

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

pose
shape
albedo
illumination
expression

$$\frac{\partial C_{pixel}}{\partial P_i}(P) = \alpha \cdot \underbrace{\frac{\partial C_{p_i}}{\partial P_i}(P)} + \beta \cdot \frac{\partial C_{p_j}}{\partial P_i}(P) + \gamma \cdot \frac{\partial C_{p_k}}{\partial P_i}(P)$$

partial derivatives of the parametric face model

Parameter Estimation as Energy Minimization

Gauss-Newton-Method

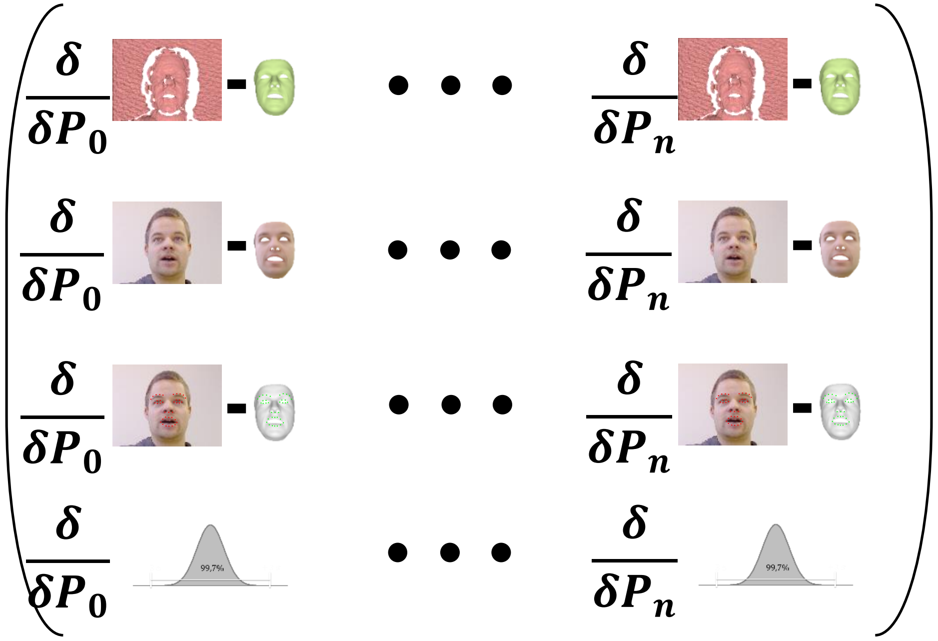
$$\boldsymbol{E(P) = |F(P)|_2^2}$$

Iterative Update Rule:

$$\boldsymbol{P_{n+1} = P_n - \underbrace{\left(J_F^T \cdot J_F\right)^{-1} \cdot J_F^T \cdot F}_{= \Delta}}$$

$$\Rightarrow \quad \boldsymbol{\left(J_F^T \cdot J_F\right) \cdot \Delta = J_F^T \cdot F}$$

$J_F(P) =$



Parameter Estimation as Energy Minimization

Gauss-Newton-Method

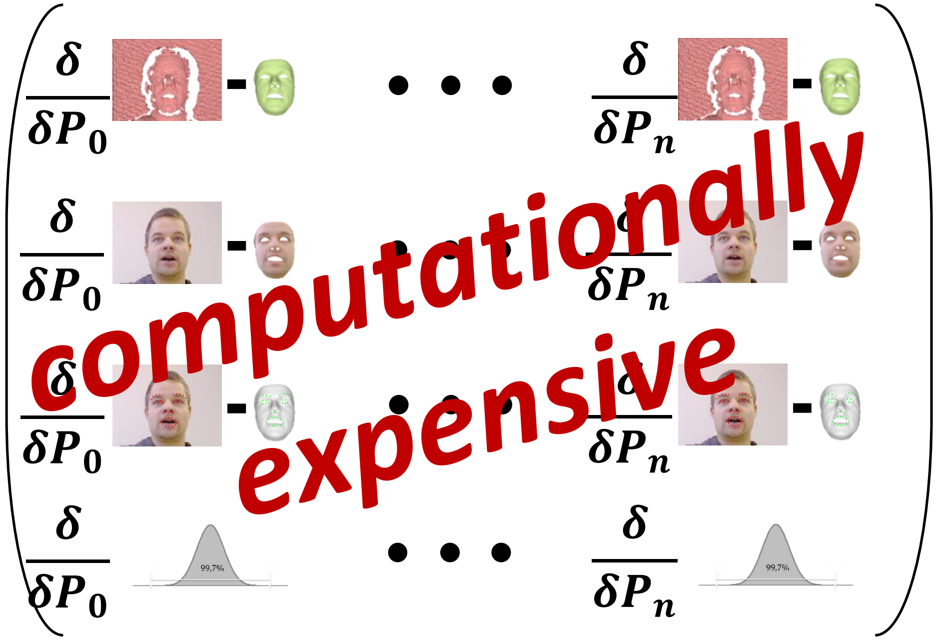
$$\boldsymbol{E(P) = |F(P)|_2^2}$$

Iterative Update Rule:

$$\boldsymbol{P_{n+1} = P_n - \underbrace{\left(J_F^T \cdot J_F\right)^{-1} \cdot J_F^T \cdot F}_{= \Delta}}$$

$$\Rightarrow \quad \boldsymbol{\left(J_F^T \cdot J_F\right) \cdot \Delta = J_F^T \cdot F}$$

$J_F(P) =$



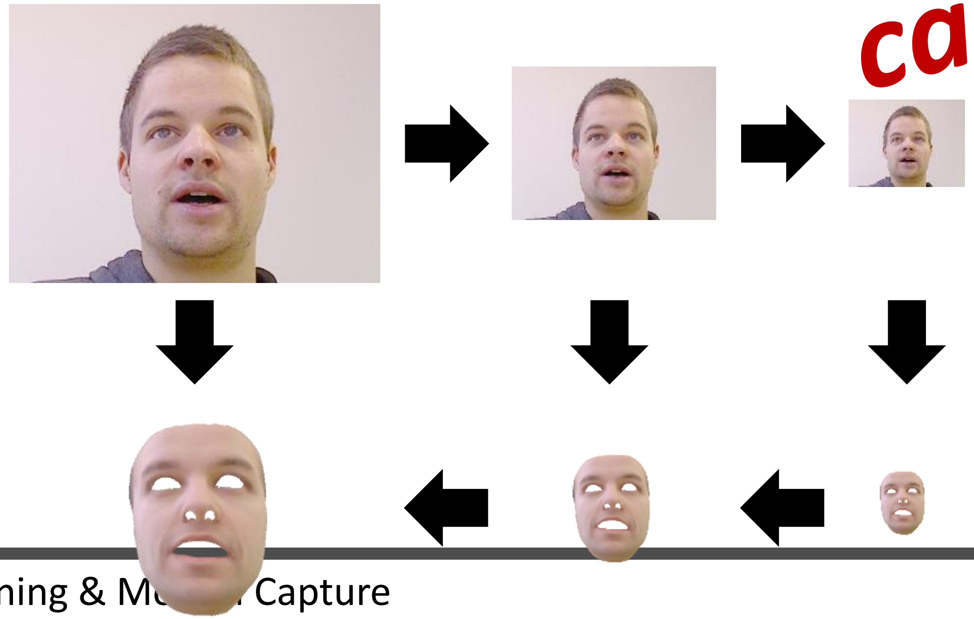
Parameter Estimation as Energy Minimization

Gauss-Newton-Method

$$\left(J_F^T \cdot J_F\right) \cdot \Delta = J_F^T \cdot F$$

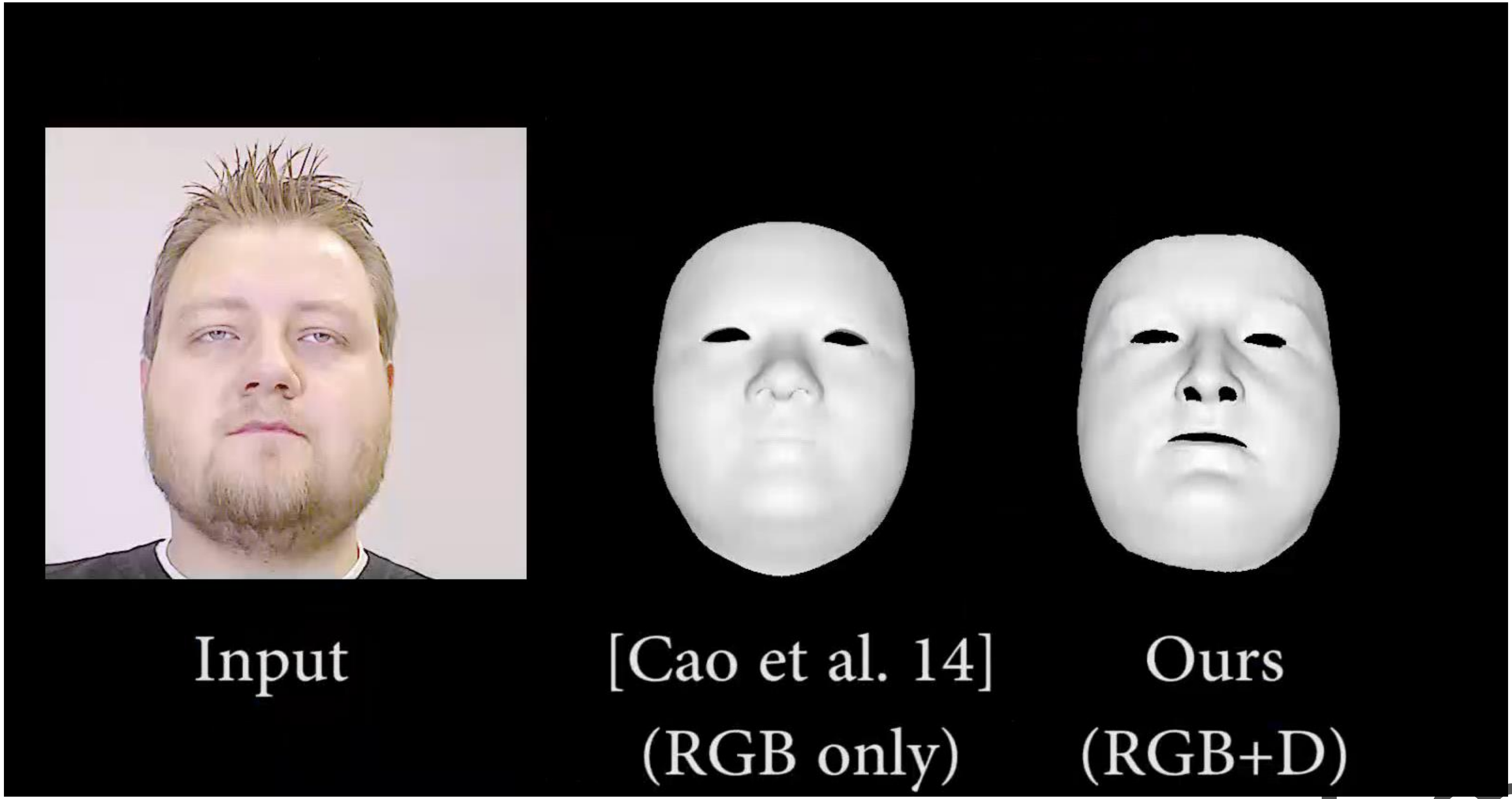
real-time capable

reduce workload



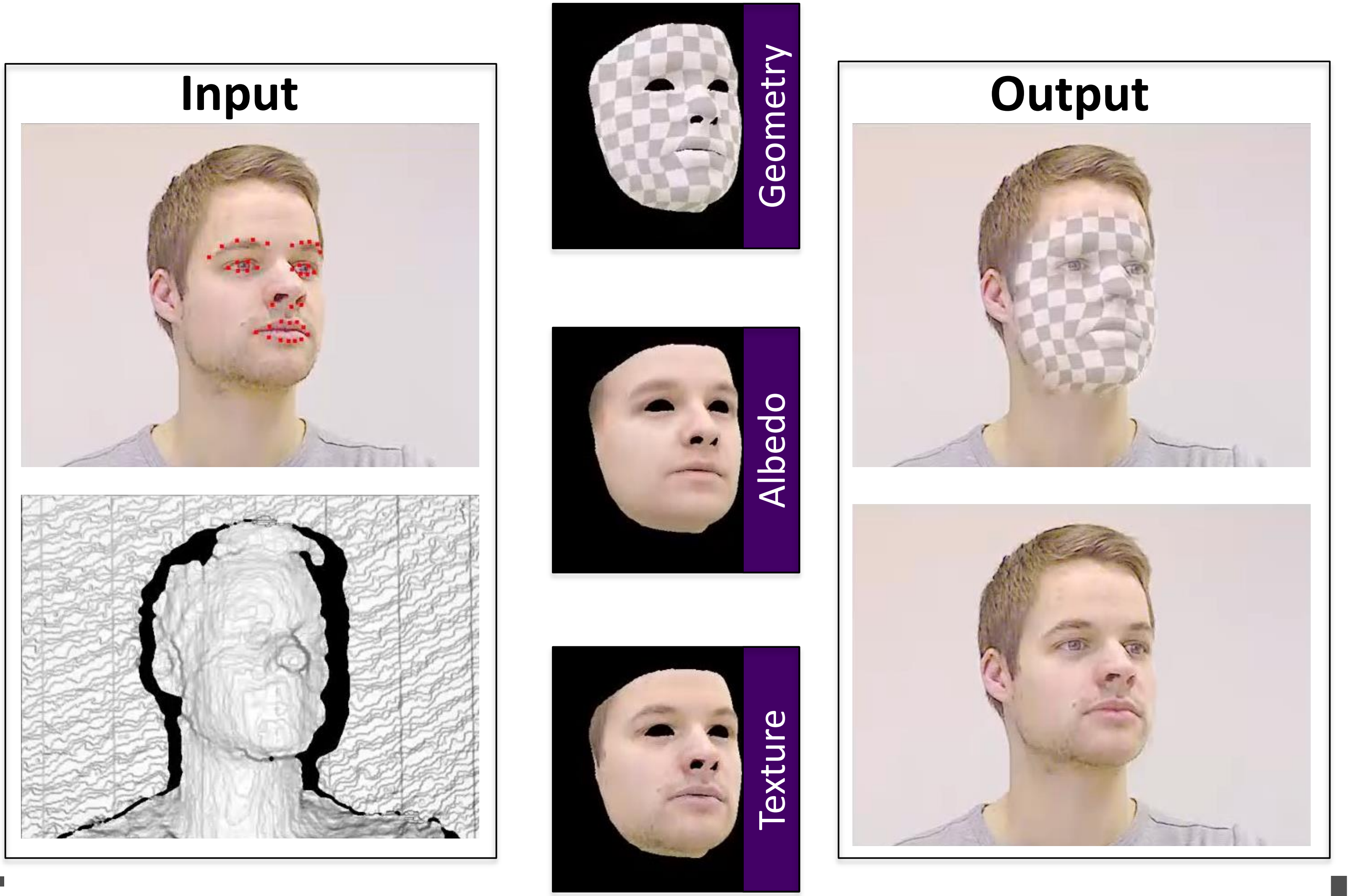
distribute workload

Facial Performance Capture



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

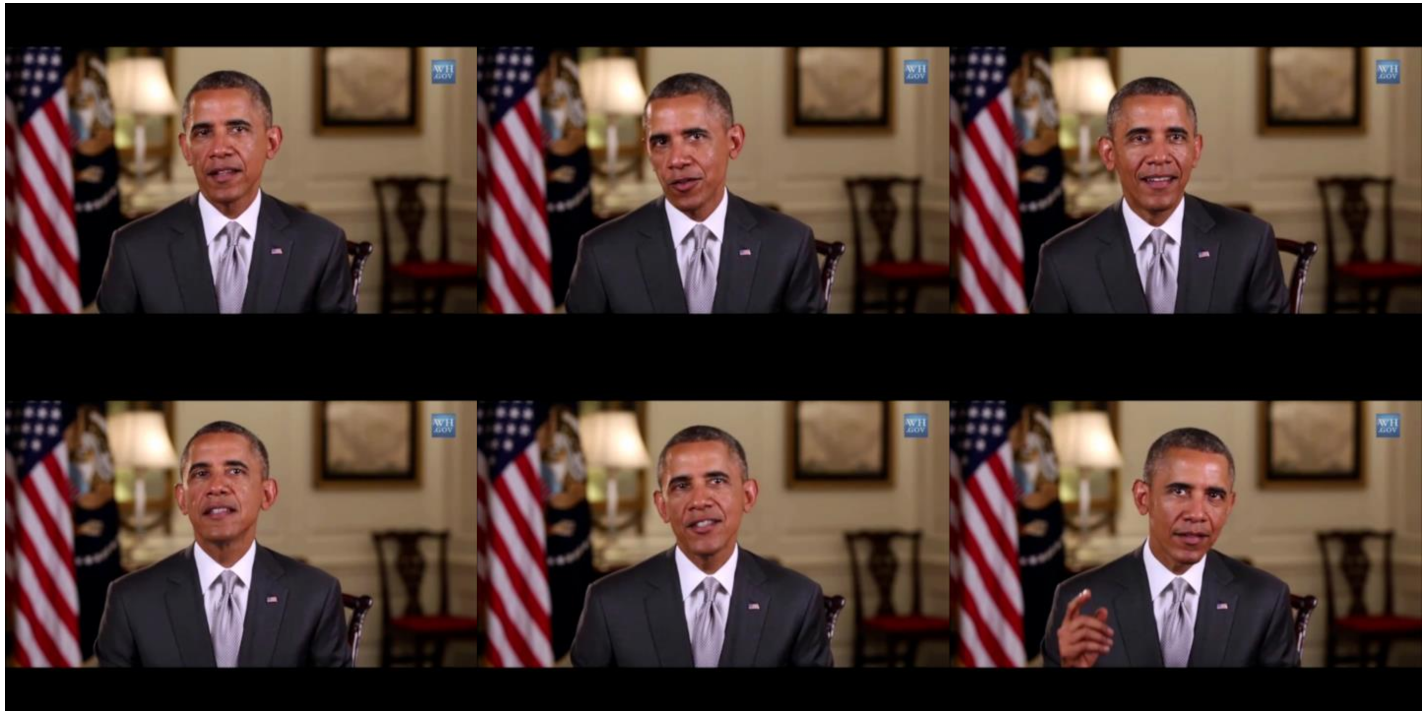
Facial Performance Capture



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

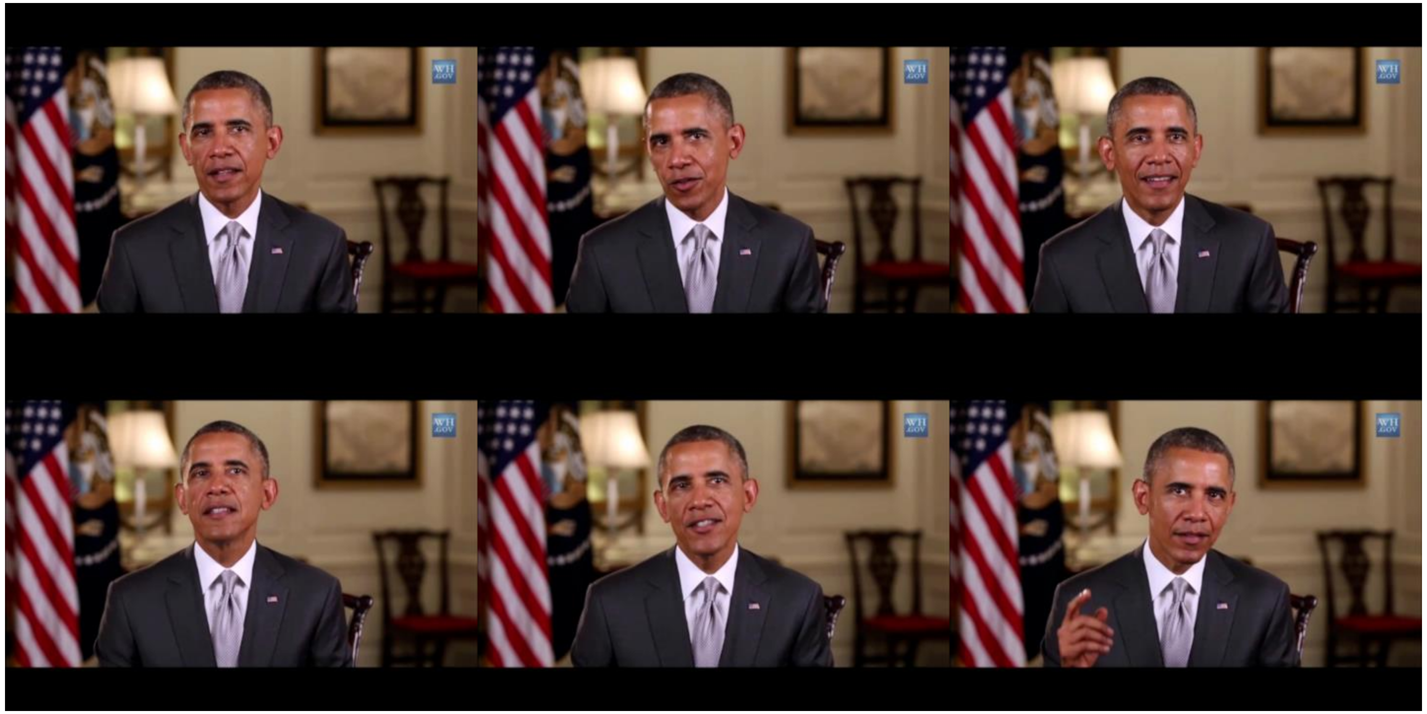
RGB-only Face Capture

- Use multiple views to estimate identity
  - “non-rigid model-based bundle adjustment”



CVPR’16 [Thies et al.]: Face2Face

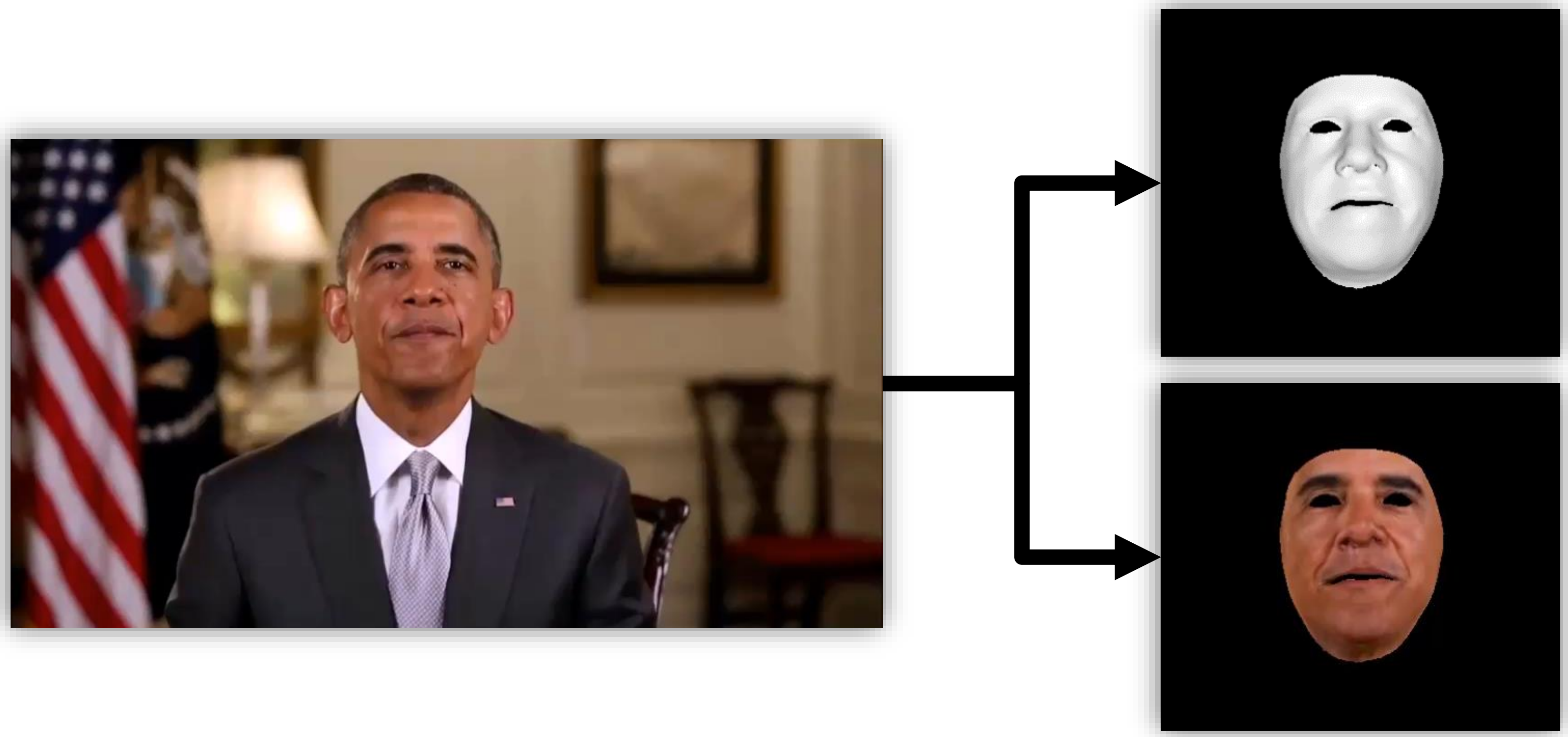
Face Identity Estimation



$$E_{total}(\boldsymbol{P}) = \sum_{i=0}^{n} E_i(\boldsymbol{P}) \quad \rightarrow min$$

CVPR'16 [Thies et al.]: Face2Face

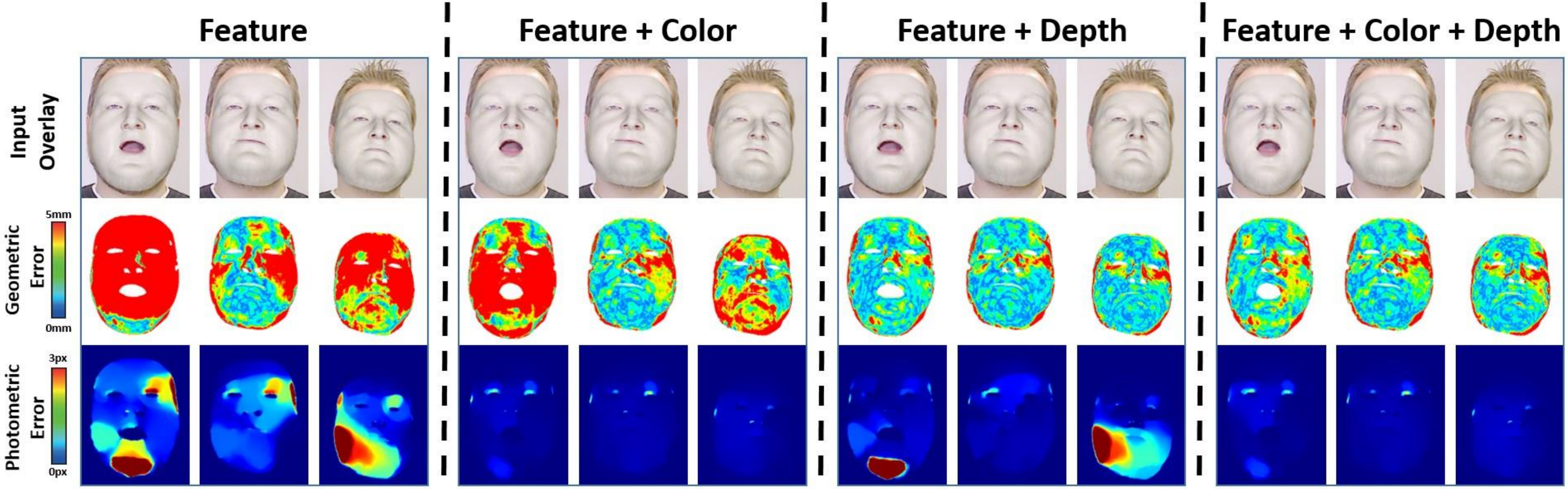
RGB-only Face Tracking



CVPR'16 [Thies et al.]: Face2Face

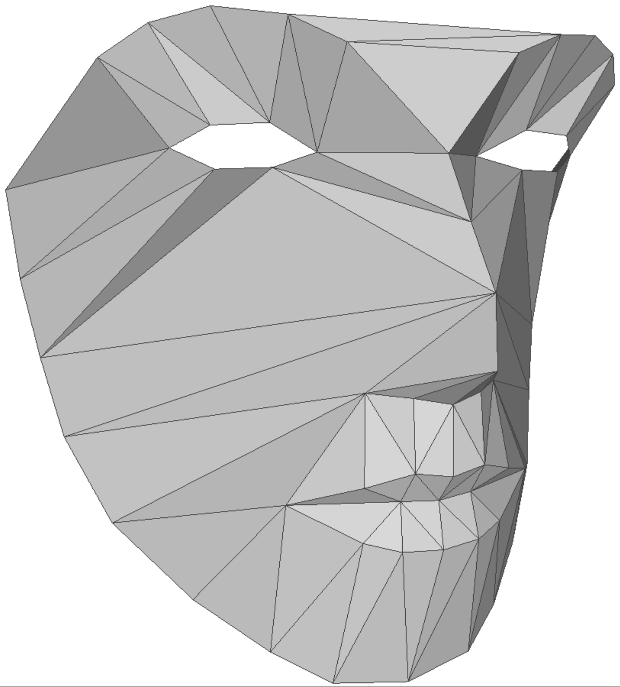
Summary - Parameter Estimation

- Energy Formulation & Minimization is dependent on
  - Input Data (RGB vs RGB-D)

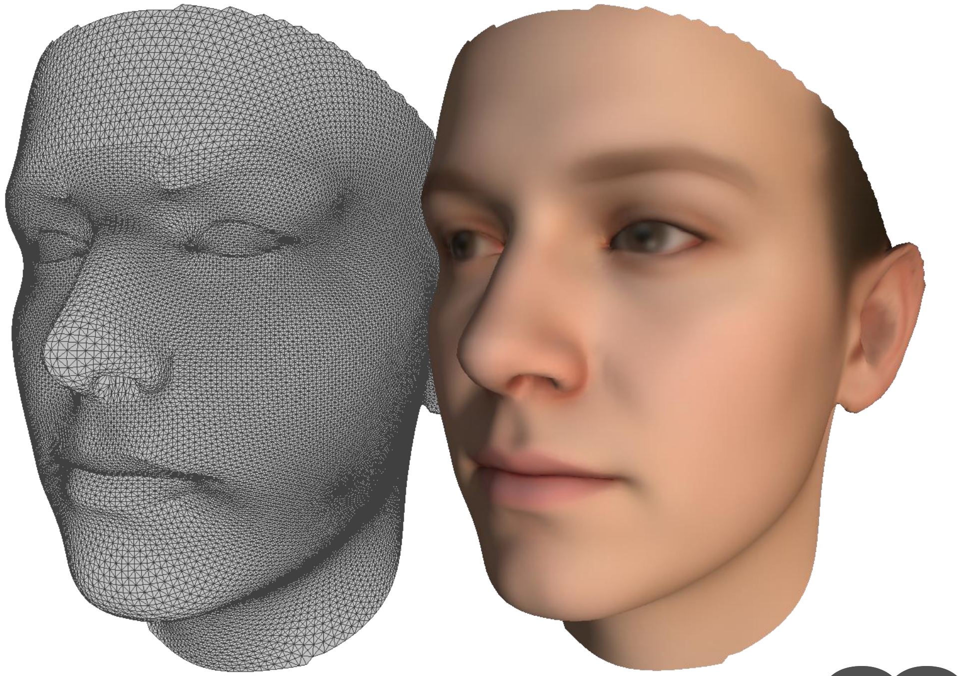


Summary - Parameter Estimation

- Energy Formulation & Minimization is dependent on
  – Input Data (RGB vs RGB-D)
  – Face Model, Illumination Model (coarse model vs dense model)



→ **Sparse Energy Terms**



→ **Dense Energy Terms**

Summary - Parameter Estimation

- Energy Formulation & Minimization is dependent on
  - Input Data (RGB vs RGB-D)
  - Face Model, Illumination Model (coarse model vs dense model)
  - Run-time requirements, hardware (Online/Offline)

→ **Sparse Energy Terms**

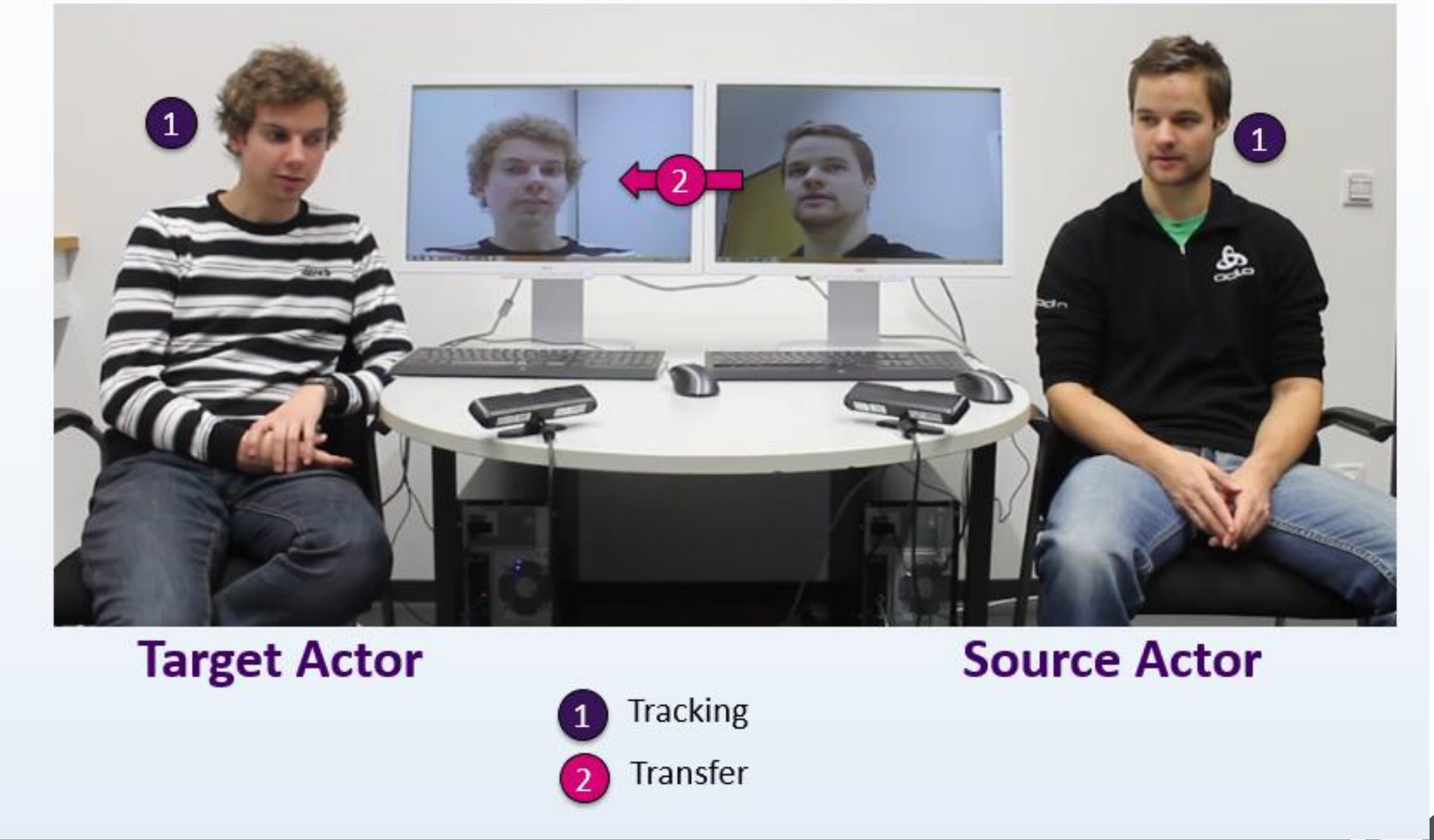
→ **Dense Energy Terms**

Summary - Parameter Estimation

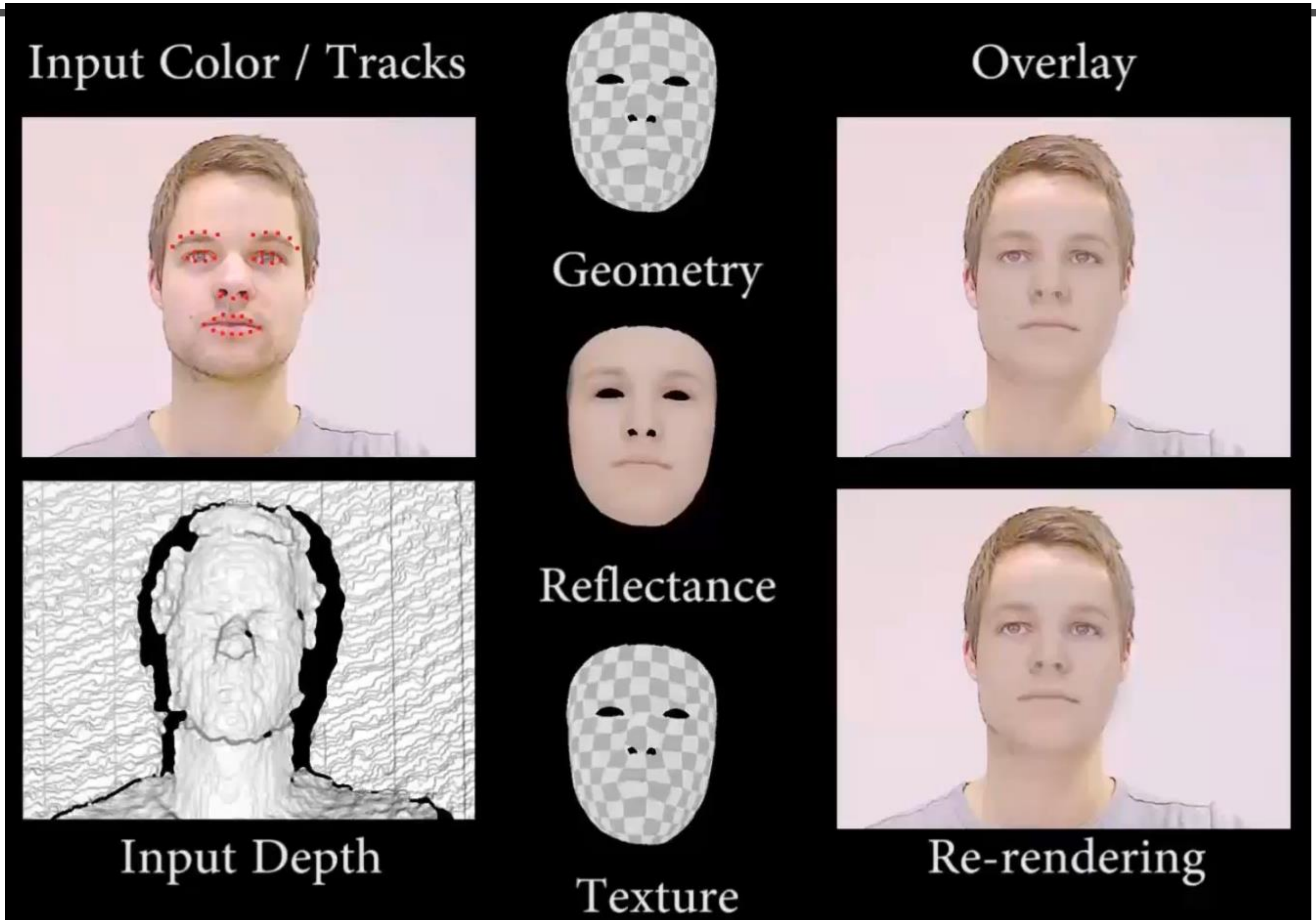
- Energy Formulation & Minimization is dependent on
  – Input Data (RGB vs RGB-D)
  – Face Model, Illumination Model (coarse model vs dense model)
  – Run-time requirements, hardware (Online/Offline)
  – Application (Do you need a dense reconstruction?)

Applications of Face Tracking & Reconstruction

Facial Reenactment

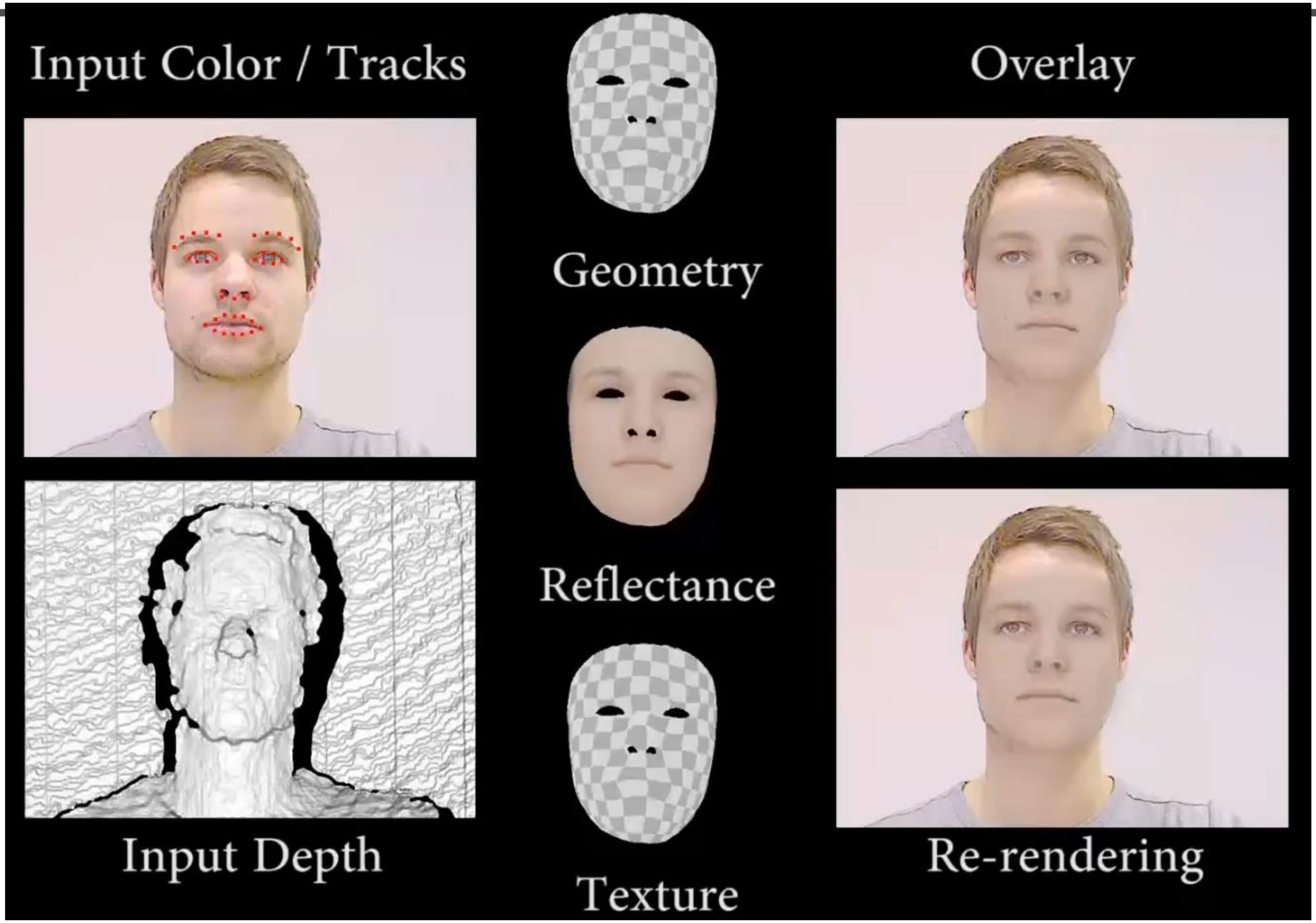


Facial Reenactment



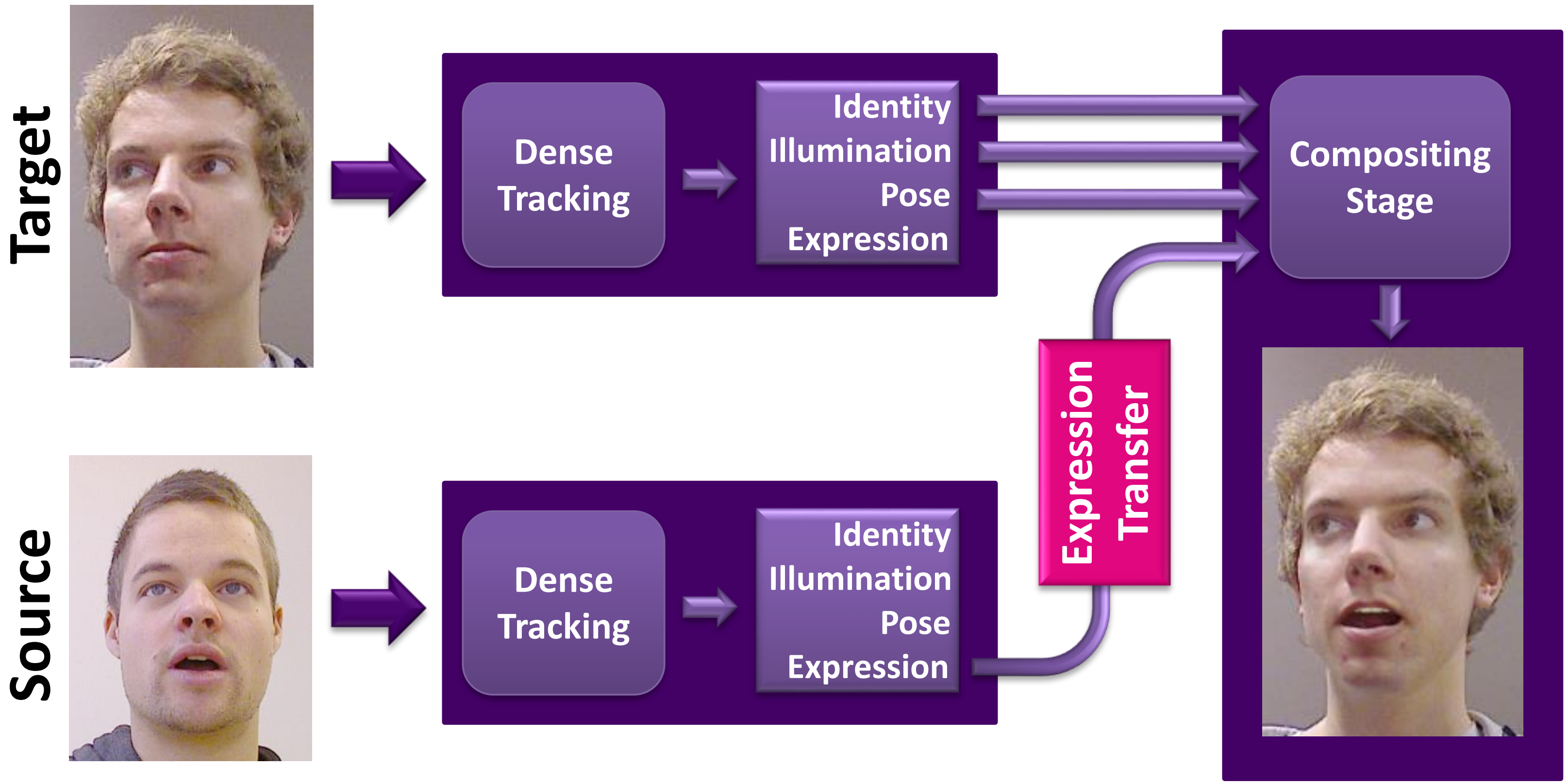
TOG'15 [Thies et al.]: Facical Reenactment

Facial Reenactment



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

Facial Expression Transfer

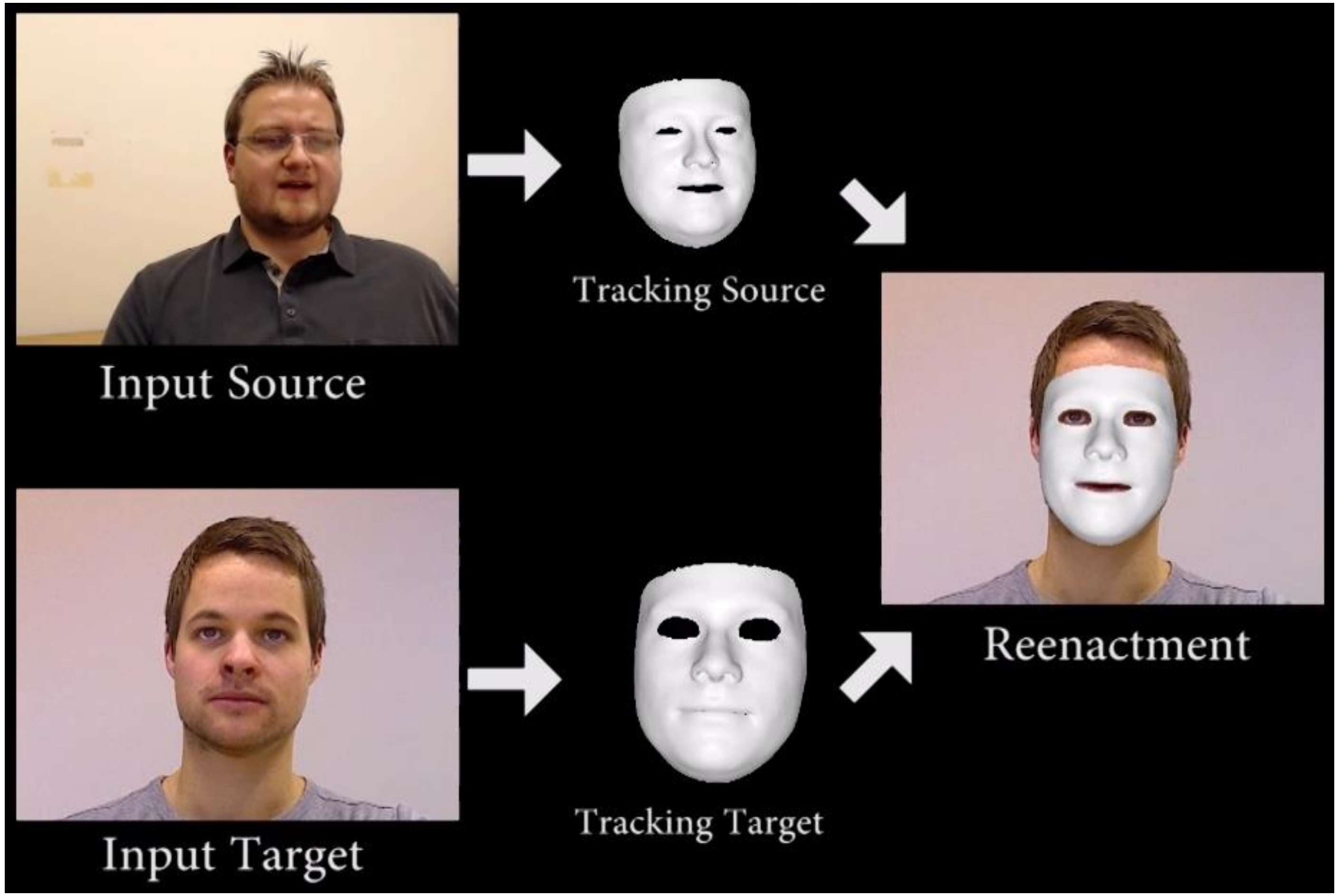


TOG'15 [Thies et al.]: Facial Reenactment

Facial Reenactment

$$P_A = \begin{pmatrix} \Phi_A \\ \alpha_A \\ \beta_A \\ \delta_A \\ \gamma_A \end{pmatrix}$$

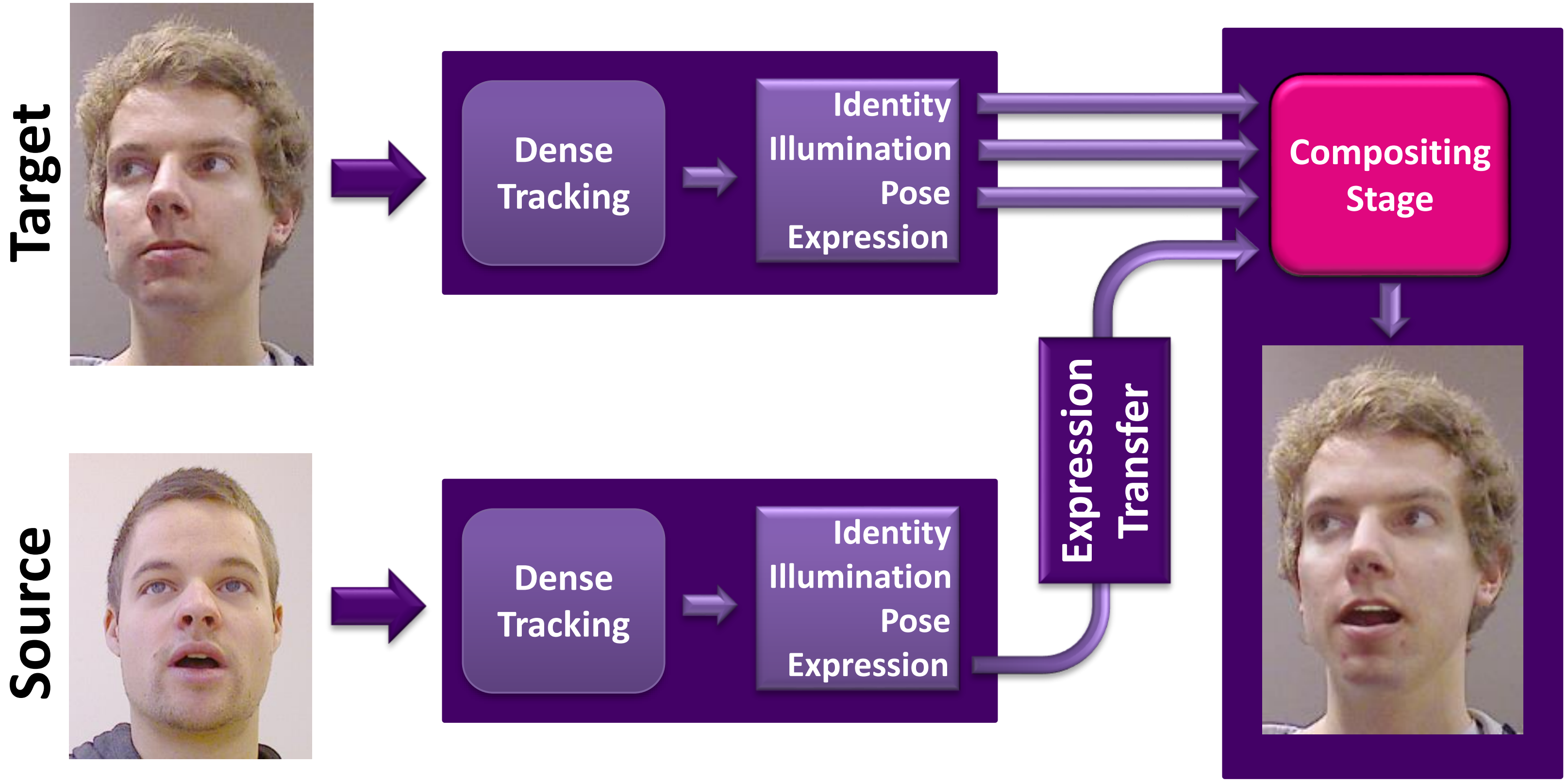
$$P_B = \begin{pmatrix} \Phi_B \\ \alpha_B \\ \beta_B \\ \delta_B \\ \gamma_B \end{pmatrix}$$



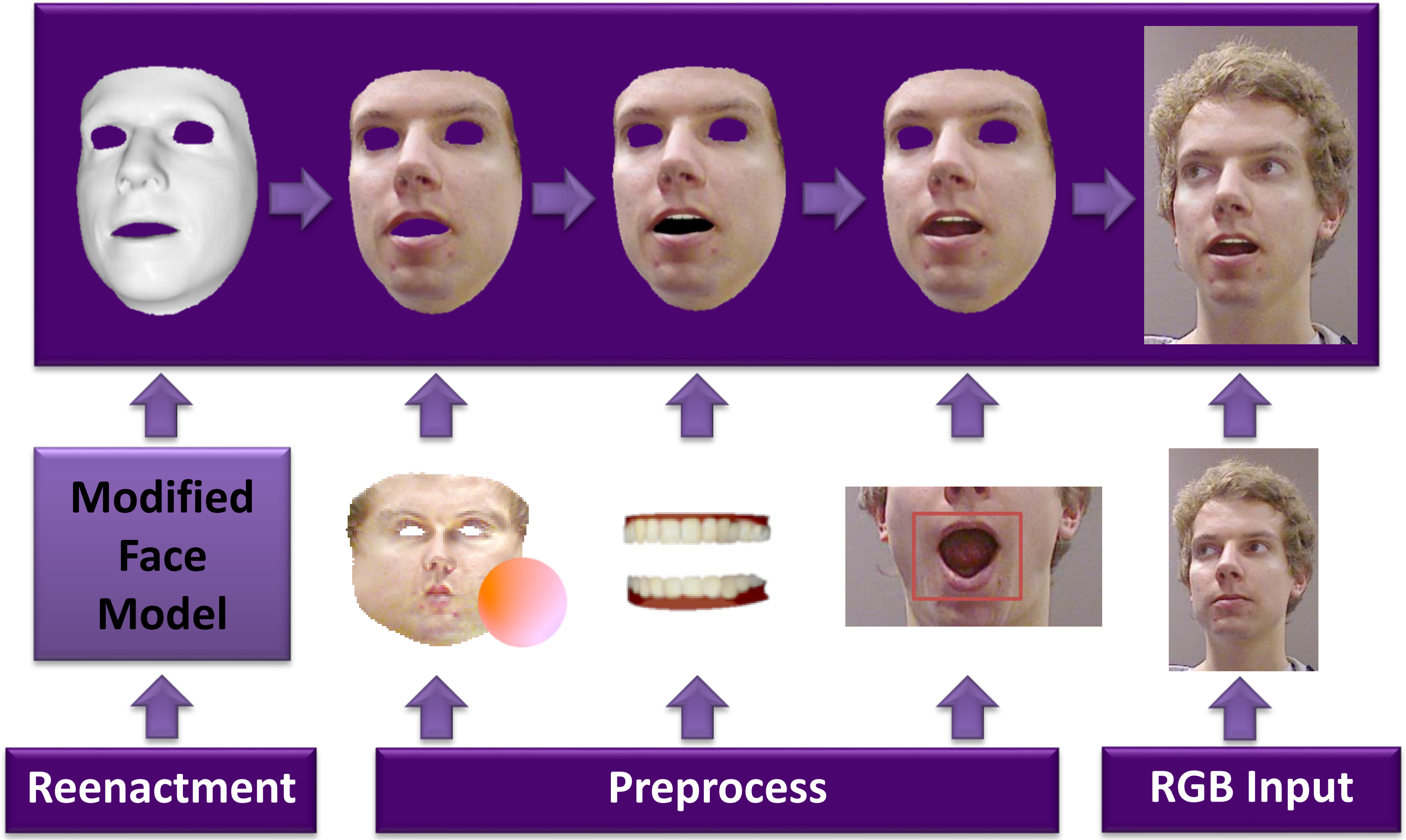
$$P_C = \begin{pmatrix} \Phi_B \\ \alpha_B \\ \beta_B \\ \delta_A \\ \gamma_B \end{pmatrix}$$

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

Facial Expression Transfer

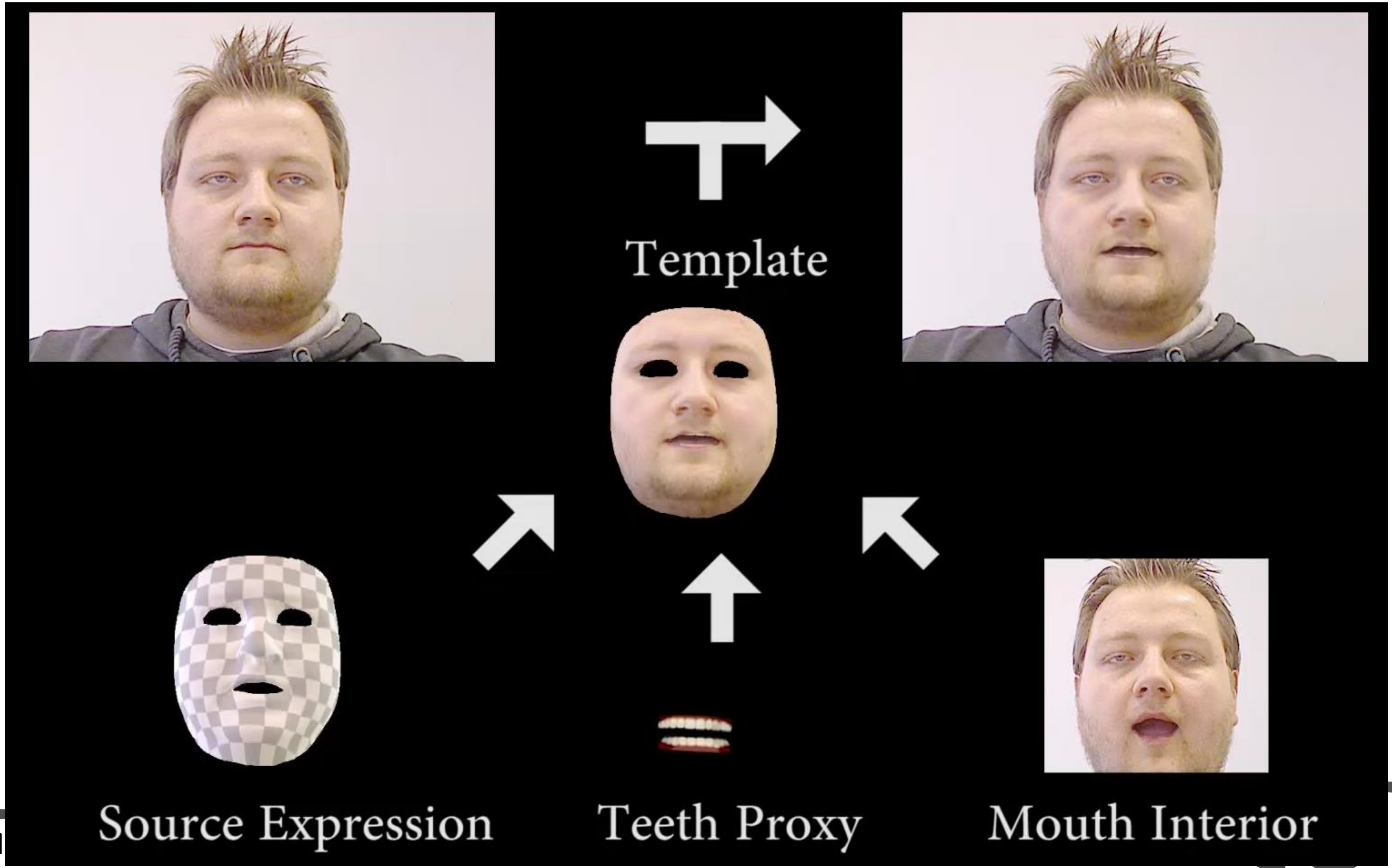


Facial Expression Transfer



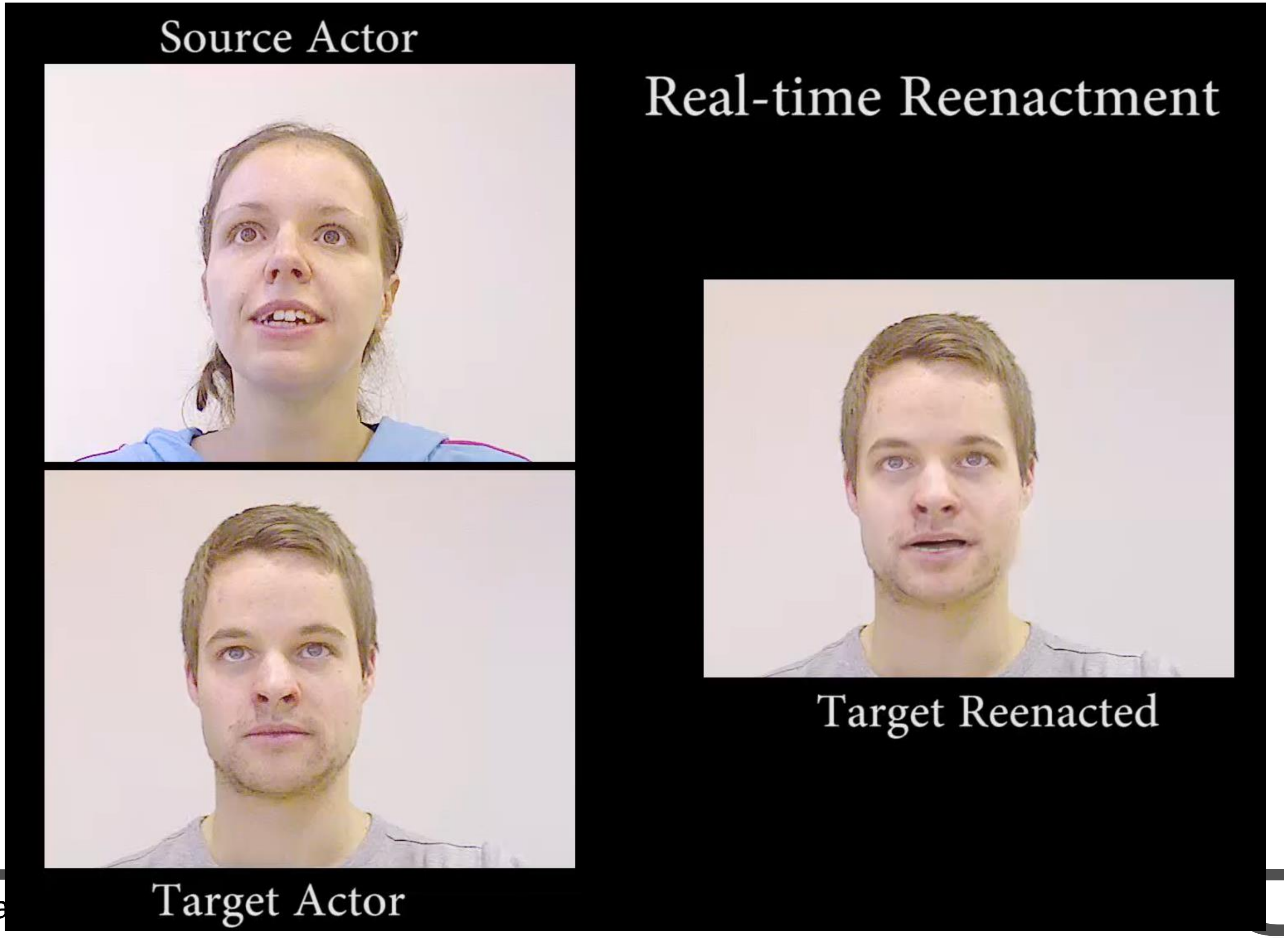
TOG'15 [Thies et al.]: Facical Reenactment

Facial Expression Transfer



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

Facial Performance Results



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

Facial Performance Results

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

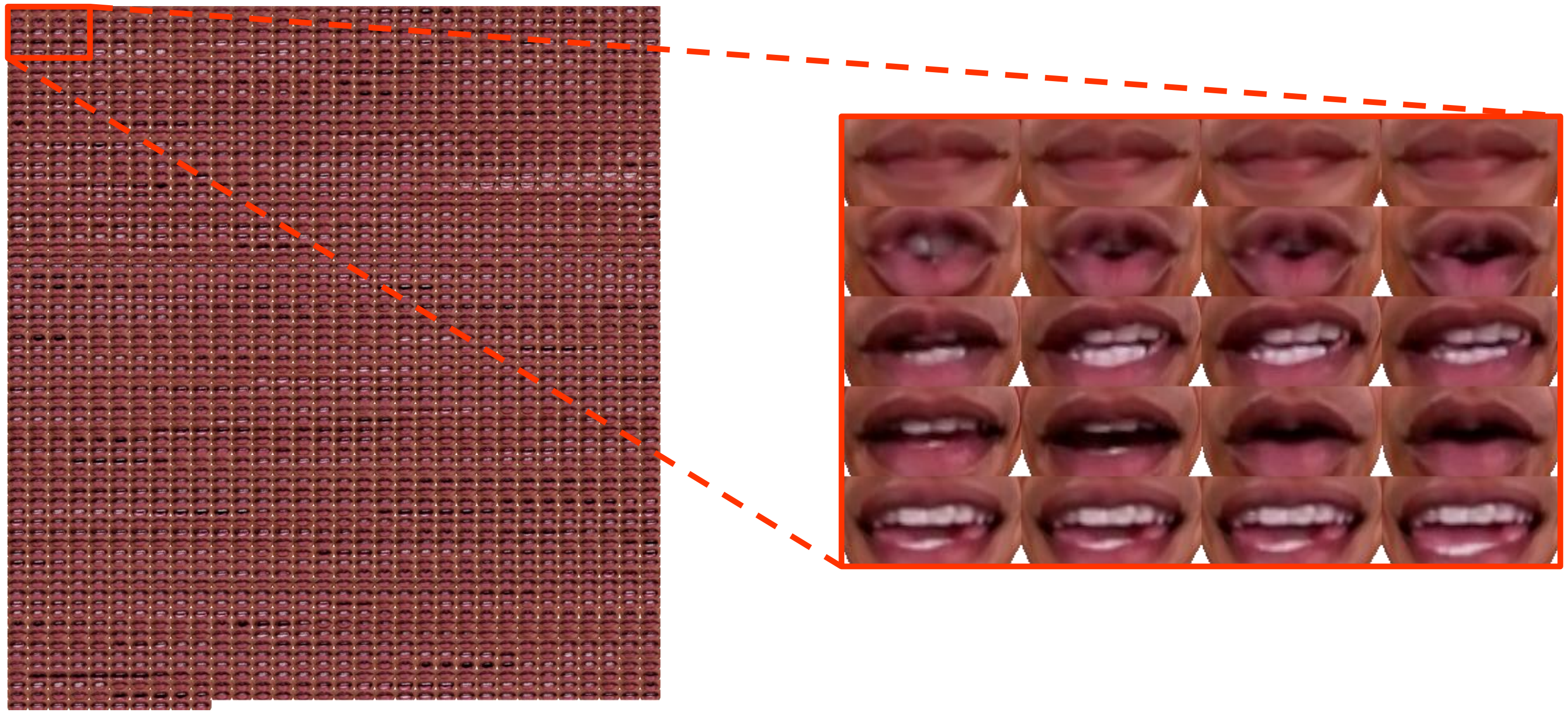
Facial Reenactment

- Teeth proxy

- Requires an RGB-D sensor

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

Mouth Retrieval

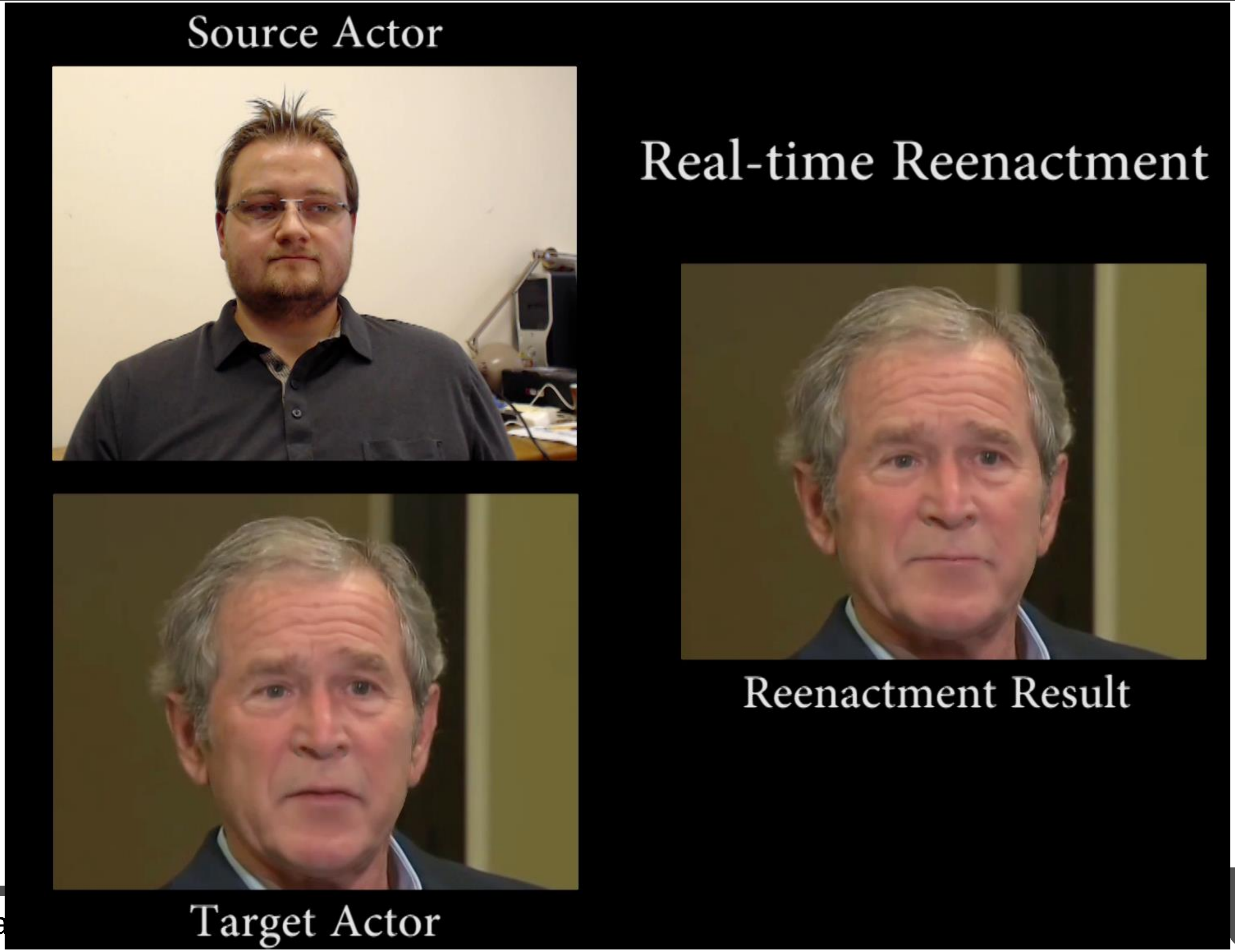


CVPR'16 [Thies et al.]: Face2Face

Face2Face: Results

CVPR'16 [Thies et al.]: Face2Face

Face2Face: Results



CVPR’16 [Thies et al.]: Face2Face

Administrative

- Reading Homework:
  - [Thies et al. 15] Real-time Expression Transfer for Facial Reenactment
    https://niessnerlab.org/projects/thies2015realtime.html
  - [Thies et al. 16] Face2Face: Real-time Face Capture and Reenactment of RGB Videos
    https://niessnerlab.org/projects/thies2016face.html
- Next week(s):
  - Tracking & Reconstruction of Bodies, Hands, etc.

Administrative

See you next week!