A linear generative model for 3D Human Faces

Umberto Crema

Mentore: Iacopo Masi

3° Weekend Ortogonale — Bertinoro 20-22 marzo 2026

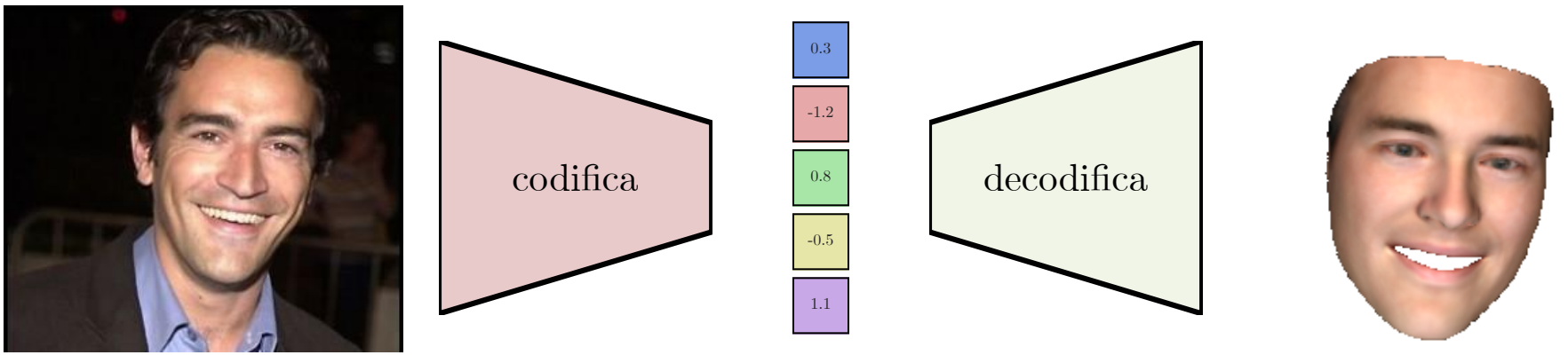
Basato su: V. Blanz & T. Vetter, SIGGRAPH 1999




Quanti parametri servono per descrivere qualsiasi volto umano?




Rappresentazione del dato

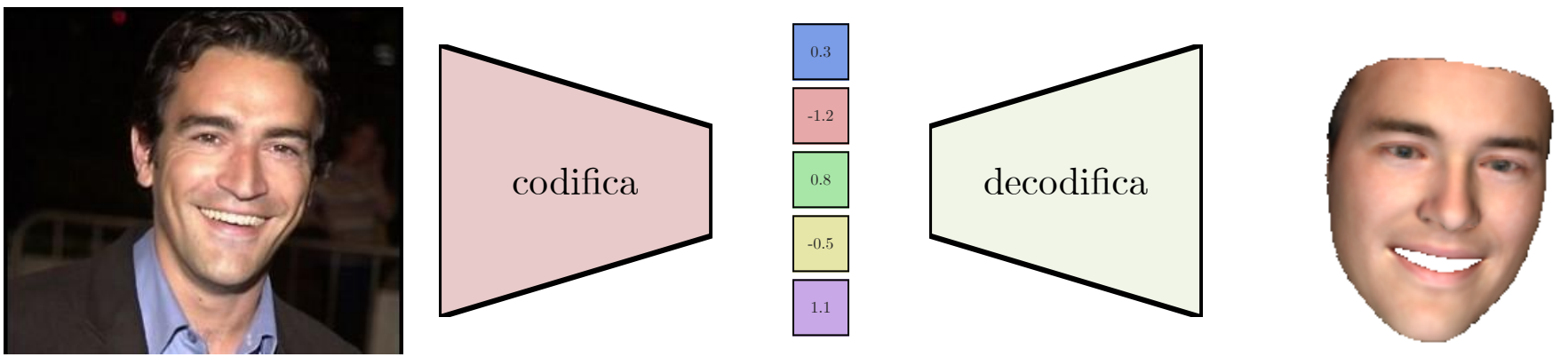


rappresentazione ad alta dimensione

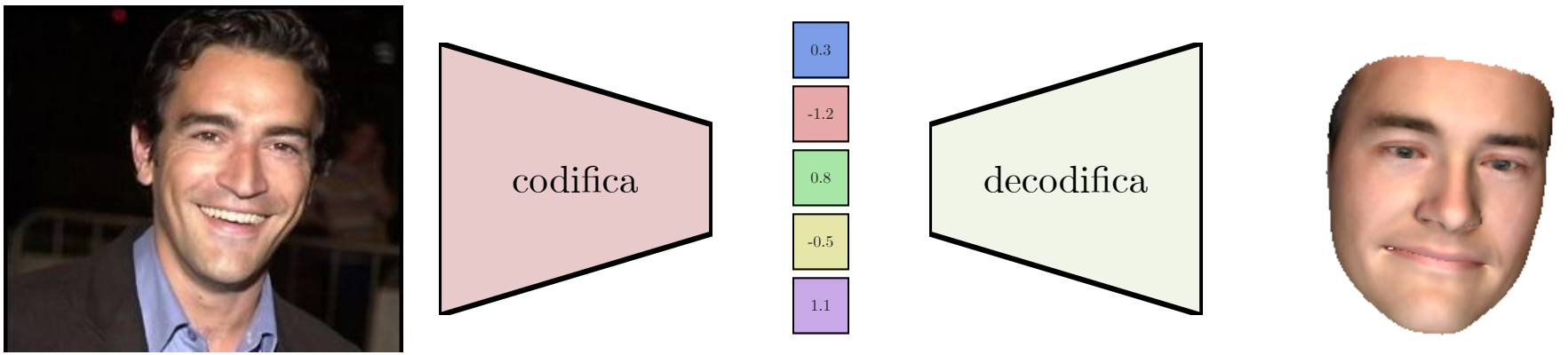
rappresentazione a bassa dimensione

ricostruzione ad alta dimensione

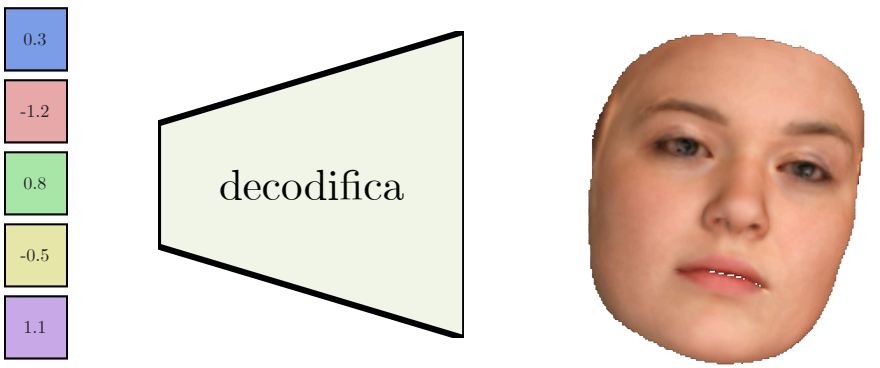
Obiettivo 1: trovare i parametri di un volto dato



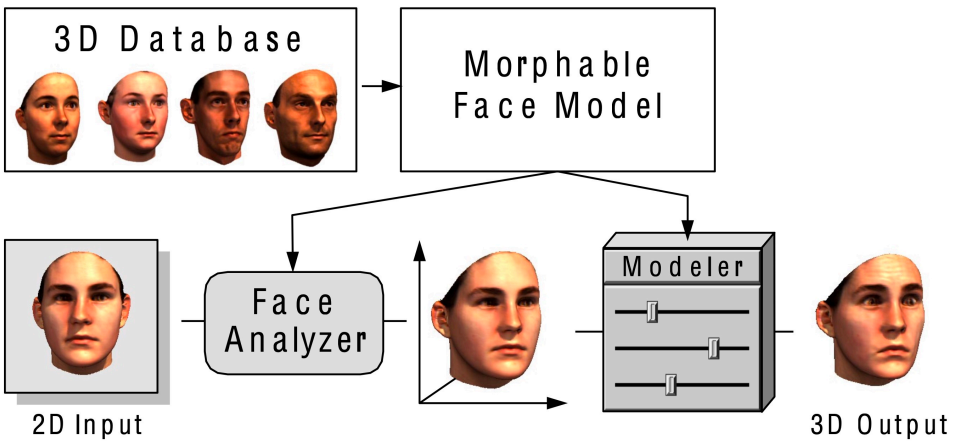
Obiettivo 2: modificare volti esistenti



Obiettivo 3: generare volti nuovi

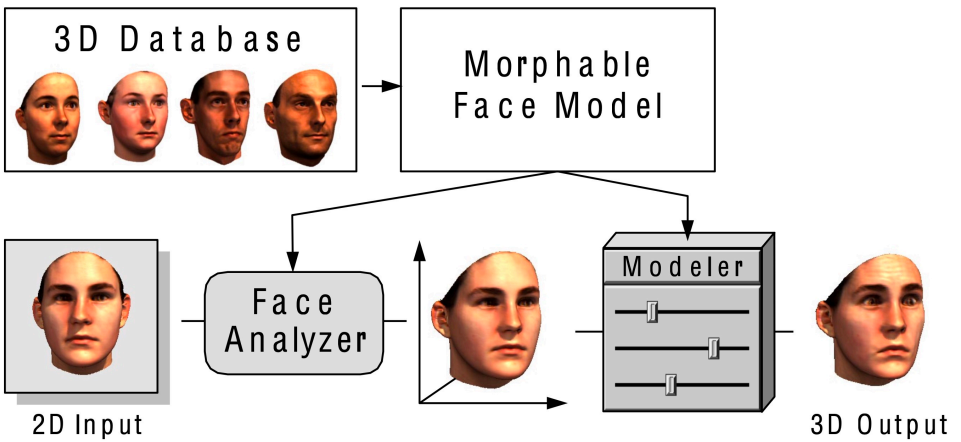


Il Sistema



Riferimenti immagine: Blanz & Vetter, A Morphable Model For The Synthesis Of 3D Faces, SIGGRAPH 1999

Il Sistema

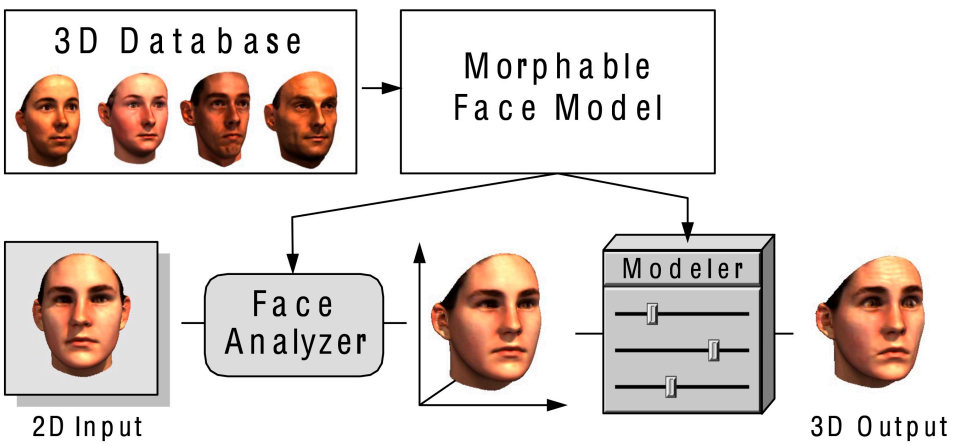


Costruzione del modello

Scansioni 3D → modello parametrico compatto

Riferimenti immagine: Blanz & Vetter, A Morphable Model For The Synthesis Of 3D Faces, SIGGRAPH 1999

Il Sistema

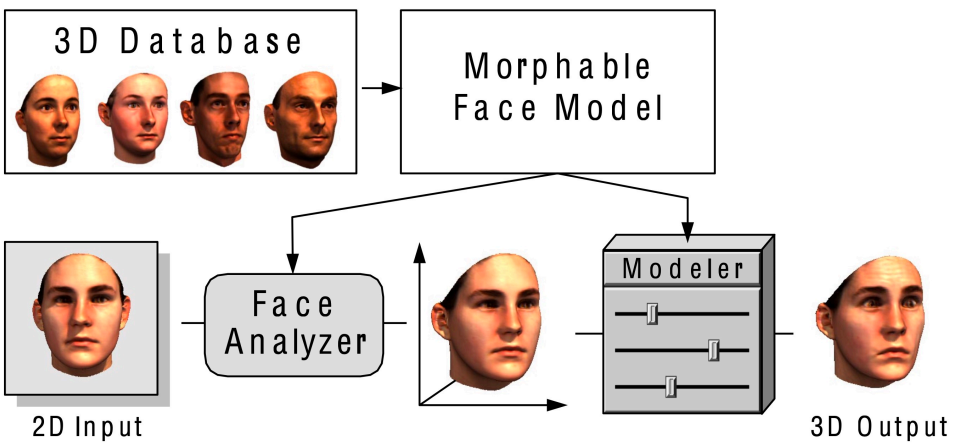


Costruzione del modello
Scansioni 3D → modello parametrico compatto

Analisi (Codifica)
Foto 2D → parametri del volto corrispondente

Riferimenti immagine: Blanz & Vetter, A Morphable Model For The Synthesis Of 3D Faces, SIGGRAPH 1999

Il Sistema



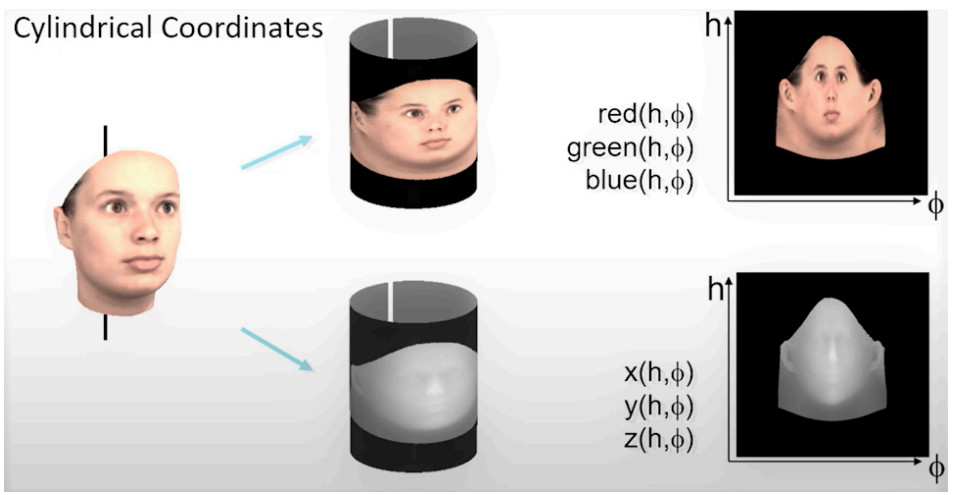
Costruzione del modello
Scansioni 3D → modello parametrico compatto

Analisi (Codifica)
Foto 2D → parametri del volto corrispondente

Sintesi (Decodifica)
Parametri → volto 3D generato o modificato

Riferimenti immagine: Blanz & Vetter, A Morphable Model For The Synthesis Of 3D Faces, SIGGRAPH 1999

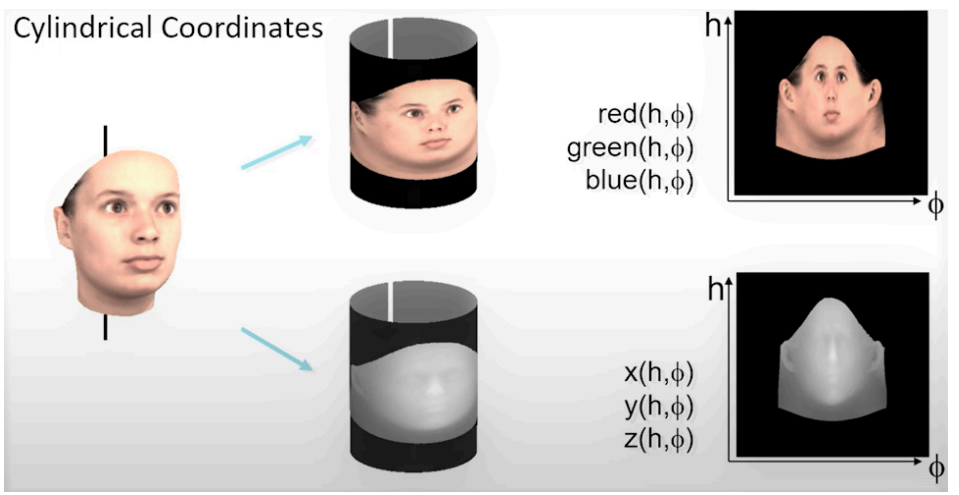
Il Dataset



Riferimenti immagine: datahacker.rs

200 scansioni laser di volti reali

Il Dataset



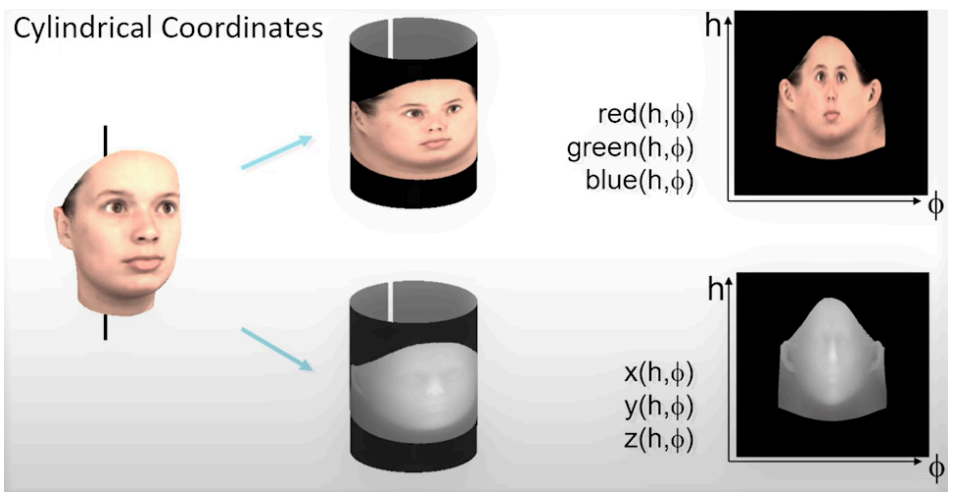
Riferimenti immagine: datahacker.rs

200 scansioni laser di volti reali

Ogni punto campionato in coordinate cilindriche (h, θ):

- **Shape**: $r(h, \theta)$
- **Texture**: R, G, $B\,(h, \theta)$

Il Dataset



Riferimenti immagine: datahacker.rs

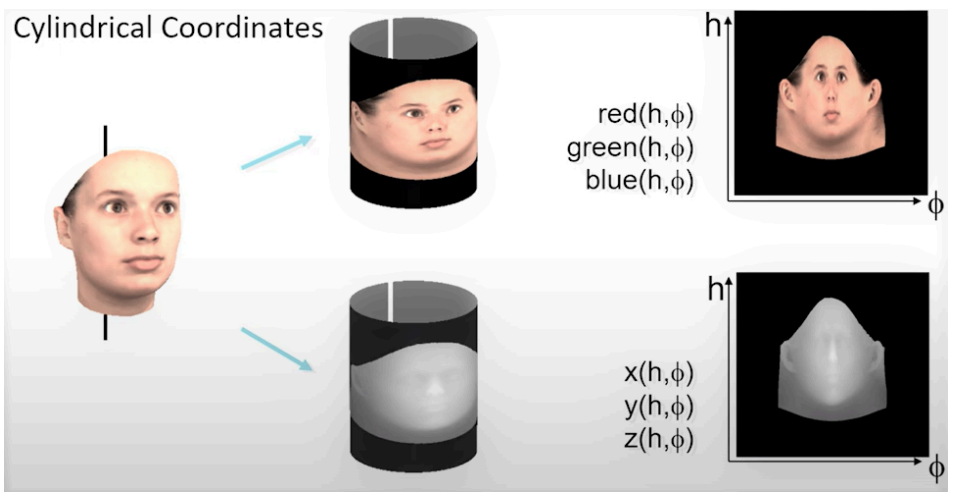
200 scansioni laser di volti reali

Ogni punto campionato in coordinate cilindriche (h, θ):

- **Shape**: $r(h, \theta)$
- **Texture**: R, G, $B\,(h, \theta)$

$n = 70\,000$ punti per scansione

Il Dataset



Riferimenti immagine: datahacker.rs

200 scansioni laser di volti reali

Ogni punto campionato in coordinate cilindriche (h, θ):

- **Shape**: $r(h, \theta)$
- **Texture**: R, G, $B\,(h, \theta)$

$n = 70\,000$ punti per scansione

Ogni volto : vettore di $\approx$ **280 000 numeri**

3D Morphable Face Model

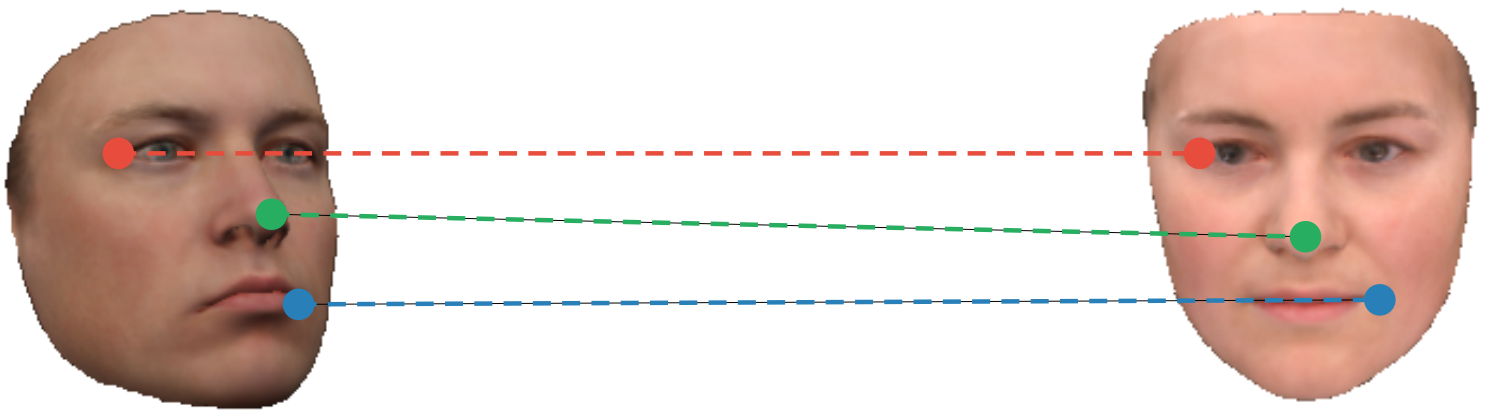
$$\textbf{Shape: } S = (X_1, Y_1, Z_1, \ldots, X_n, Y_n, Z_n)^T \in \mathbb{R}^{3n}$$

$$\textbf{Texture: } T = (R_1, G_1, B_1, \ldots, R_n, G_n, B_n)^T \in \mathbb{R}^{3n}$$

3D Morphable Face Model

Shape: $S = (X_1, Y_1, Z_1, \ldots, X_n, Y_n, Z_n)^T \in \mathbb{R}^{3n}$

Texture: $T = (R_1, G_1, B_1, \ldots, R_n, G_n, B_n)^T \in \mathbb{R}^{3n}$



Ipotesi fondamentale: volti allineati

Modello lineare

Morphable face model: combinazione lineare di m prototipi

$$S_{mod} = \sum_{i=1}^{m} a_i S_i \qquad T_{mod} = \sum_{i=1}^{m} b_i T_i$$

Lo spazio dei volti

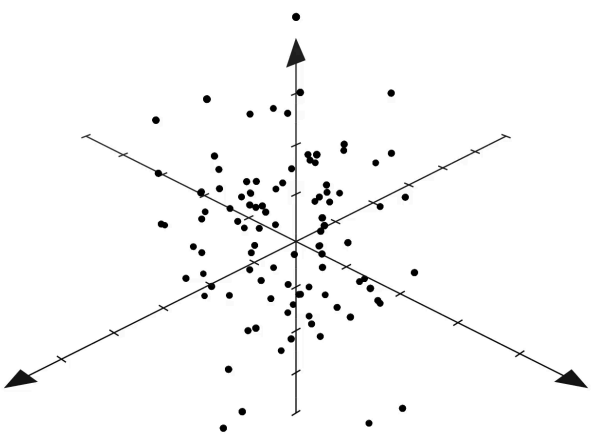
Morphable model: insieme di volti

$$\left(S_{mod}(\vec{a}),\ T_{mod}(\vec{b})\right)$$

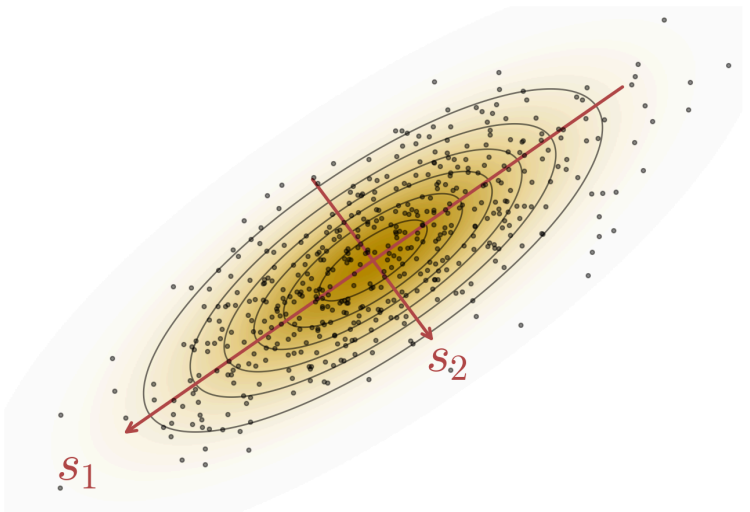
parametrizzato da:

- $\vec{a} = (a_1, \ldots, a_m)^T$
- $\vec{b} = (b_1, \ldots, b_m)^T$

Nuovi volti generati variando $\vec{a}$ e $\vec{b}$



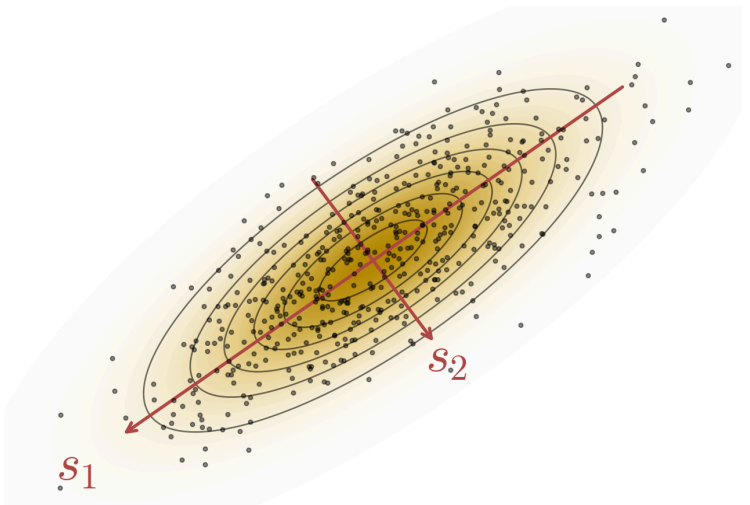
PCA sullo spazio dei volti



Proiezione 2D dello spazio dei volti (analogia)

PCA sullo spazio dei volti

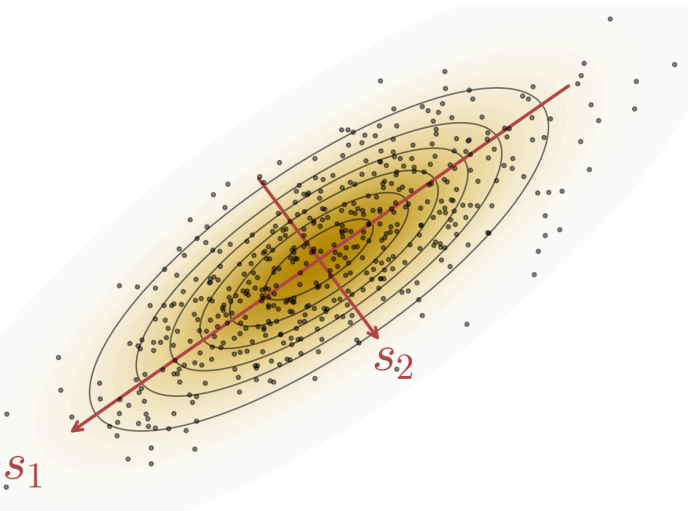
Volto medio: $\bar{S}$, $\bar{T}$



Proiezione 2D dello spazio dei volti (analogia)

PCA sullo spazio dei volti

Volto medio: $\bar{S}$, $\bar{T}$



Proiezione 2D dello spazio dei volti (analogia)

Matrici di covarianza:

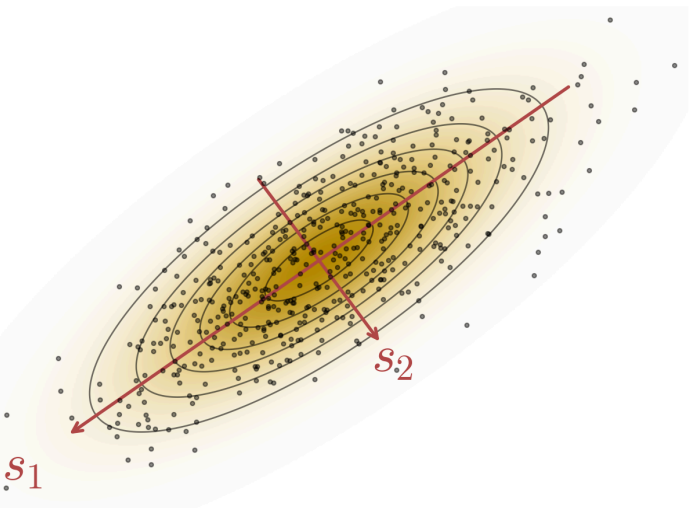
- C_S dalla shape: $\Delta S_i = S_i - \bar{S}$
- C_T dalla texture: $\Delta T_i = T_i - \bar{T}$

PCA su C_S, C_T

→ componenti principali s_j, t_j
ordinate per varianza (autovalore) decrescente

PCA sullo spazio dei volti

Volto medio: $\bar{S}$, $\bar{T}$



Proiezione 2D dello spazio dei volti (analogia)

Matrici di covarianza:

- C_S dalla shape: $\Delta S_i = S_i - \bar{S}$
- C_T dalla texture: $\Delta T_i = T_i - \bar{T}$

PCA su C_S, C_T

→ componenti principali s_j, t_j
ordinate per varianza (autovalore) decrescente

Rappresentazione compatta

Ogni volto si esprime come scostamento dal volto medio lungo le componenti principali:

$$S_{mod} = \bar{S} + \sum_{i=1}^{m-1} \alpha_i \boldsymbol{s}_i \qquad T_{mod} = \bar{T} + \sum_{i=1}^{m-1} \beta_i \boldsymbol{t}_i$$

Rappresentazione compatta

Ogni volto si esprime come scostamento dal volto medio lungo le componenti principali:

$$S_{mod} = \bar{S} + \sum_{i=1}^{m-1} \alpha_i \boldsymbol{s}_i \qquad T_{mod} = \bar{T} + \sum_{i=1}^{m-1} \beta_i \boldsymbol{t}_i$$

I coefficienti α_i, β_i sono i **parametri del modello**.

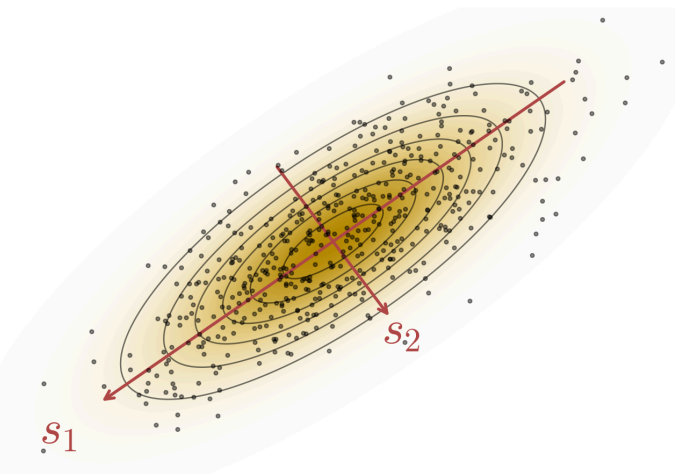
da $\approx 280\,000$ numeri a poche decine di parametri significativi

Modellazione del volto variando solo i primi k coefficienti

Plausibilità dei volti

Probabilità a priori per distinguere facce plausibili da innaturali:

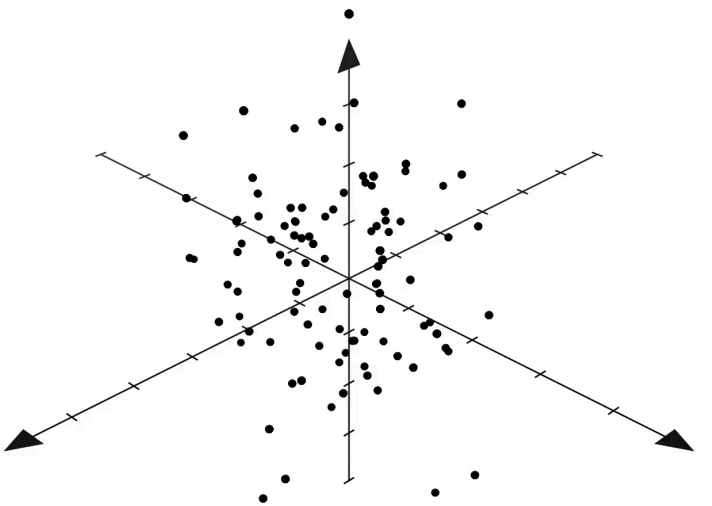
$$p(\vec{\alpha}) \sim \exp\left[-\frac{1}{2}\sum_{i=1}^{m-1}\left(\frac{\alpha_i}{\sigma_i}\right)^2\right]$$



$\sigma_i^2 = \lambda_i^S$ (autovalore di C_S)

Fitting ad un'immagine

Obiettivo: data un'immagine,
trovare i parametri per i quali
il rendering del volto modellato
è simile all'immagine



Due tipi di parametri

Coefficienti 3DMM

$$\vec{\alpha},\ \vec{\beta}$$

- shape del volto
- texture del volto

proprietà intrinseche

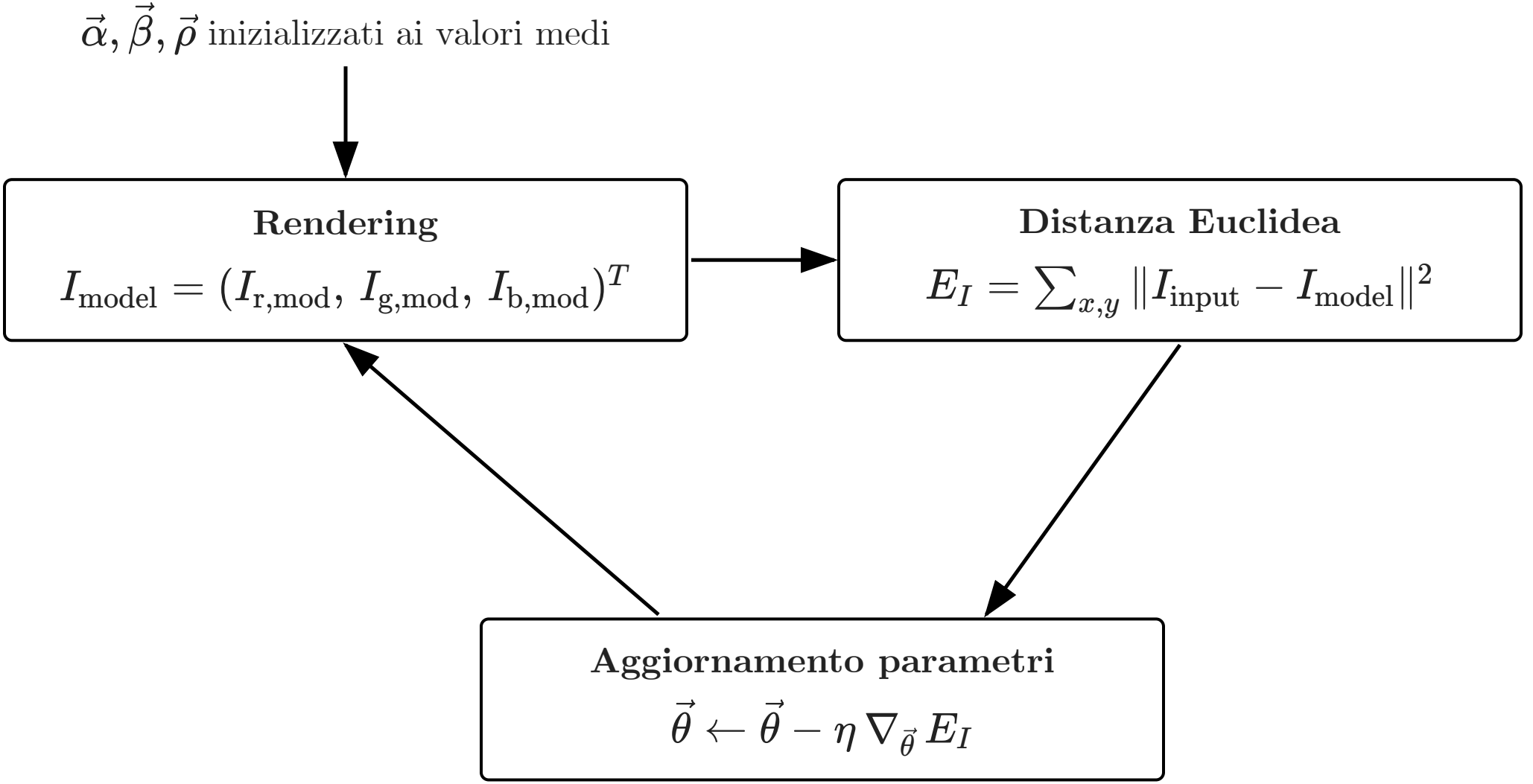
Parametri di rendering

$$\vec{\rho}$$

- posizione, rotazione, scala
- illuminazione

proprietà estrinseche

Loop Analysis-By-Synthesis



Loop Analysis-By-Synthesis

$\vec{\alpha}, \vec{\beta}, \vec{\rho}$ inizializzati ai valori medi

Rendering

$$I_{\text{model}} = (I_{\text{r,mod}},\ I_{\text{g,mod}},\ I_{\text{b,mod}})^T$$

Distanza Euclidea

$$E_I = \sum_{x,y} \|I_{\text{input}} - I_{\text{model}}\|^2$$

Aggiornamento parametri

$$\vec{\theta} \leftarrow \vec{\theta} - \eta \nabla_{\vec{\theta}} E_I$$

Loop Analysis-By-Synthesis

$\vec{\alpha}, \vec{\beta}, \vec{\rho}$ inizializzati ai valori medi

Rendering

$$I_{\text{model}} = (I_{\text{r,mod}},\ I_{\text{g,mod}},\ I_{\text{b,mod}})^T$$

Distanza Euclidea

$$E_I = \sum_{x,y} \|I_{\text{input}} - I_{\text{model}}\|^2$$

Aggiornamento parametri

$$\vec{\theta} \leftarrow \vec{\theta} - \eta \nabla_{\vec{\theta}} E_I$$

Loop Analysis-By-Synthesis

$\vec{\alpha}, \vec{\beta}, \vec{\rho}$ inizializzati ai valori medi

Rendering

$$I_{\text{model}} = (I_{\text{r,mod}},\ I_{\text{g,mod}},\ I_{\text{b,mod}})^T$$

Distanza Euclidea

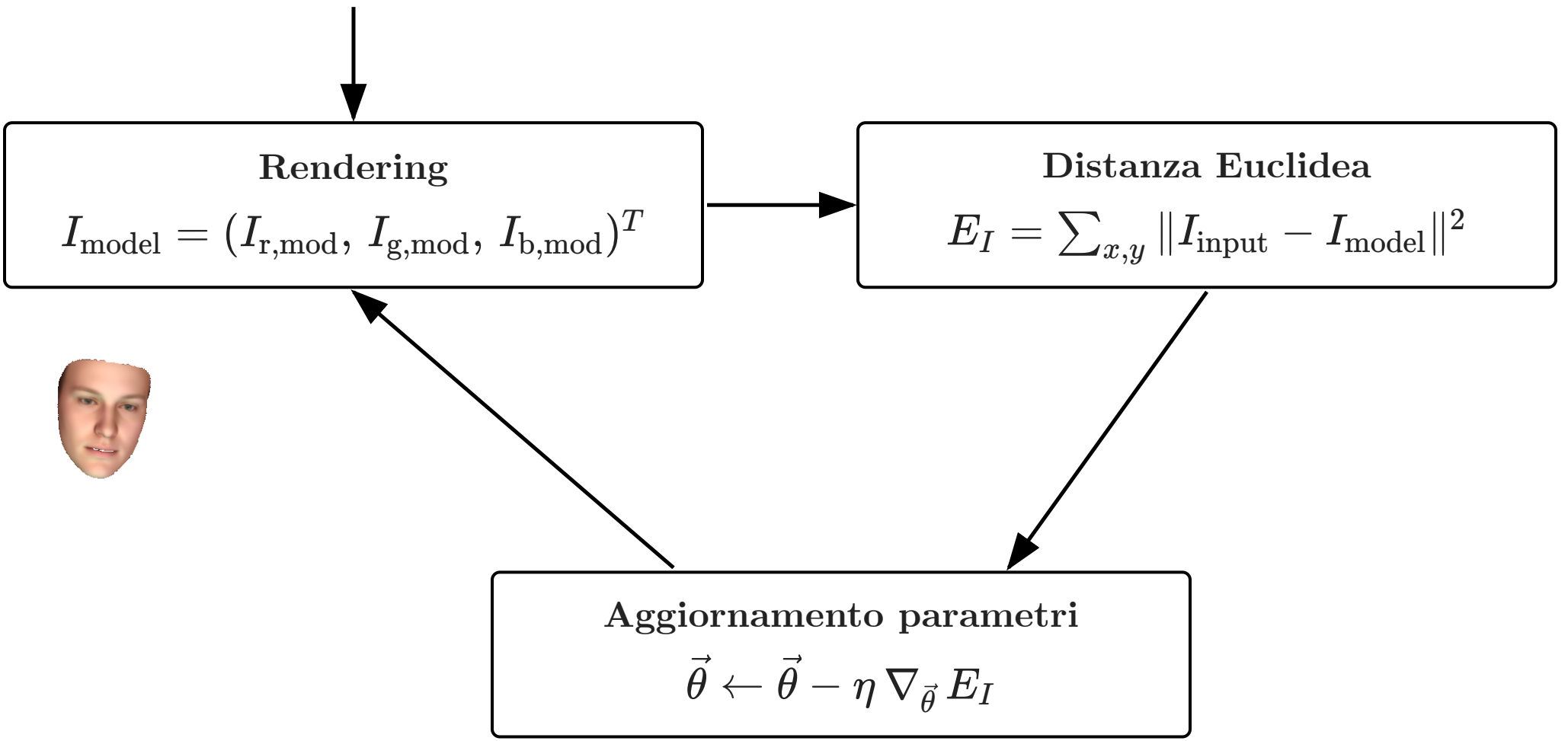
$$E_I = \sum_{x,y} \| I_{\text{input}} - I_{\text{model}} \|^2$$

Aggiornamento parametri

$$\vec{\theta} \leftarrow \vec{\theta} - \eta \nabla_{\vec{\theta}} E_I$$

Loop Analysis-By-Synthesis

$\vec{\alpha}, \vec{\beta}, \vec{\rho}$ inizializzati ai valori medi



Loop Analysis-By-Synthesis

$\vec{\alpha}, \vec{\beta}, \vec{\rho}$ inizializzati ai valori medi

Rendering

$$I_{\text{model}} = (I_{\text{r,mod}},\ I_{\text{g,mod}},\ I_{\text{b,mod}})^T$$

Distanza Euclidea

$$E_I = \sum_{x,y} \| I_{\text{input}} - I_{\text{model}} \|^2$$

Aggiornamento parametri

$$\vec{\theta} \leftarrow \vec{\theta} - \eta \nabla_{\vec{\theta}} E_I$$

Loop Analysis-By-Synthesis

$\vec{\alpha}, \vec{\beta}, \vec{\rho}$ inizializzati ai valori medi

Rendering

$$I_{\text{model}} = (I_{\text{r,mod}},\ I_{\text{g,mod}},\ I_{\text{b,mod}})^T$$

Distanza Euclidea

$$E_I = \sum_{x,y} \| I_{\text{input}} - I_{\text{model}} \|^2$$

Aggiornamento parametri

$$\vec{\theta} \leftarrow \vec{\theta} - \eta \nabla_{\vec{\theta}} E_I$$

Loop Analysis-By-Synthesis

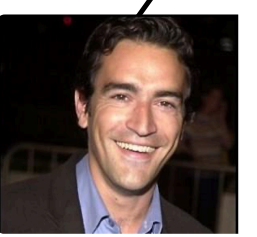
$\vec{\alpha}, \vec{\beta}, \vec{\rho}$ inizializzati ai valori medi

Rendering

$$I_{\text{model}} = (I_{\text{r,mod}},\ I_{\text{g,mod}},\ I_{\text{b,mod}})^T$$

Distanza Euclidea

$$E_I = \sum_{x,y} \| I_{\text{input}} - I_{\text{model}} \|^2$$



-

Aggiornamento parametri

$$\vec{\theta} \leftarrow \vec{\theta} - \eta \nabla_{\vec{\theta}} E_I$$

Problema mal posto

Ricostruzione corretta

Rendering

Ricostruzione degenere

Tradeoff:

- Qualità: E_I
- Probabilità: $p(\vec{\alpha}, \vec{\beta})$

Maximum A Posteriori

Likelihood

$$p(I_{\text{input}} \mid \vec{\alpha}, \vec{\beta}, \vec{\rho}) \sim \exp\left[-\frac{E_I}{2\sigma_N^2}\right]$$

Maximum A Posteriori

Likelihood

$$p(I_{\text{input}} \mid \vec{\alpha}, \vec{\beta}, \vec{\rho}) \sim \exp\left[-\frac{E_I}{2\sigma_N^2}\right]$$

Funzione di costo

$$\arg\max_{\vec{\alpha},\vec{\beta},\vec{\rho}} p(\vec{\alpha}, \vec{\beta}, \vec{\rho} \mid I_{\text{input}}) = \arg\max_{\vec{\alpha},\vec{\beta},\vec{\rho}} \frac{p(I_{\text{input}} \mid \vec{\alpha}, \vec{\beta}, \vec{\rho})\, p(\vec{\alpha}, \vec{\beta}, \vec{\rho})}{p(I_{\text{input}})} =$$

$$\arg\min_{\vec{\alpha},\vec{\beta},\vec{\rho}} \left[\underbrace{\frac{E_I}{\sigma_N^2}}_{\text{qualità}} + \underbrace{\sum_j \frac{\alpha_j^2}{\sigma_{S,j}^2} + \sum_j \frac{\beta_j^2}{\sigma_{T,j}^2} + \sum_j \frac{(\rho_j - \bar{\rho}_j)^2}{\sigma_{\rho,j}^2}}_{\text{plausibilità}} \right]$$

Coarse to fine

2 fasi:

- fitting rigido: posizione, rotazione, scala
- fitting non rigido: shape, texture, illuminazione
  - prima $\alpha_1, \ldots, \alpha_k,\ \beta_1, \ldots, \beta_k$
  - poi $\alpha_{k+1}, \ldots,\ \beta_{k+1}, \ldots$

Coarse to fine

2 fasi:

- fitting rigido: posizione, rotazione, scala
- fitting non rigido: shape, texture, illuminazione
  - prima $\alpha_1, \ldots, \alpha_k,\ \beta_1, \ldots, \beta_k$
  - poi $\alpha_{k+1}, \ldots,\ \beta_{k+1}, \ldots$

σ_N^2 decrescente:

- inizio: valore alto → tanto peso alla prior
- fine: valore basso → tanto peso alla qualità del match

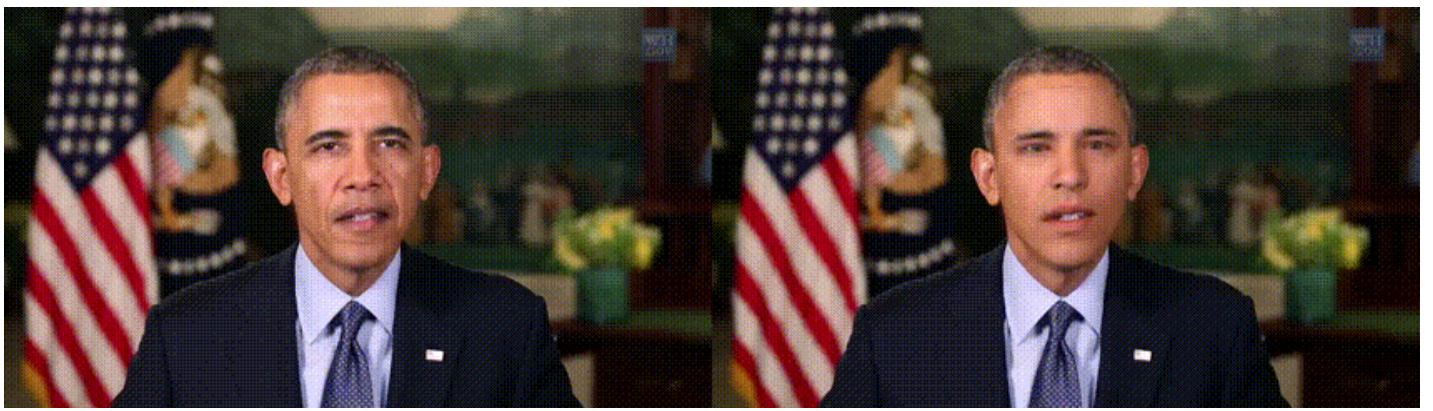
Applicazioni: Riconoscimento facciale

- $\vec{\alpha}, \vec{\beta}$ separano identità da posa e illuminazione
- Metrica di distanza robusta a pose, luci, espressioni

Limite: prestazioni in-the-wild non convincenti

Applicazioni: Intrattenimento

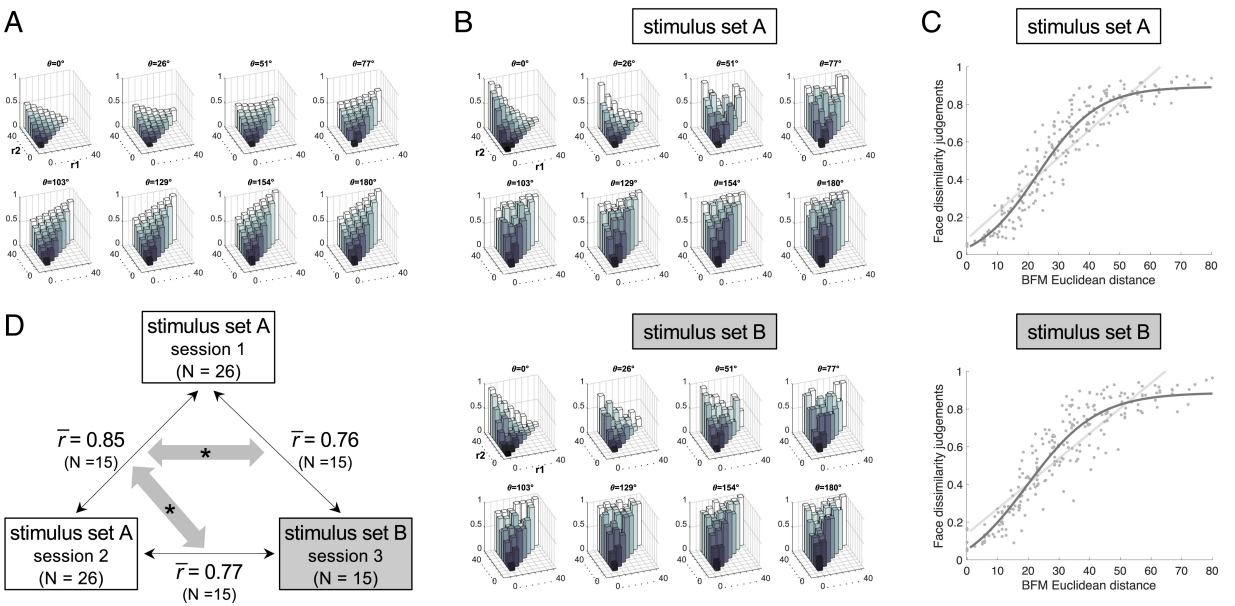
- Avatar in tempo reale: ricostruzione da singola immagine e controllo via webcam
- Face reenactment: trasferimento di espressioni, precursore dei deepfake
- Visual dubbing: adattamento del movimento labiale a nuova traccia audio



Riferimenti immagine: github.com/ascust/3DMM-Fitting-Pytorch

Applicazioni: Neuroscienze e psicologia

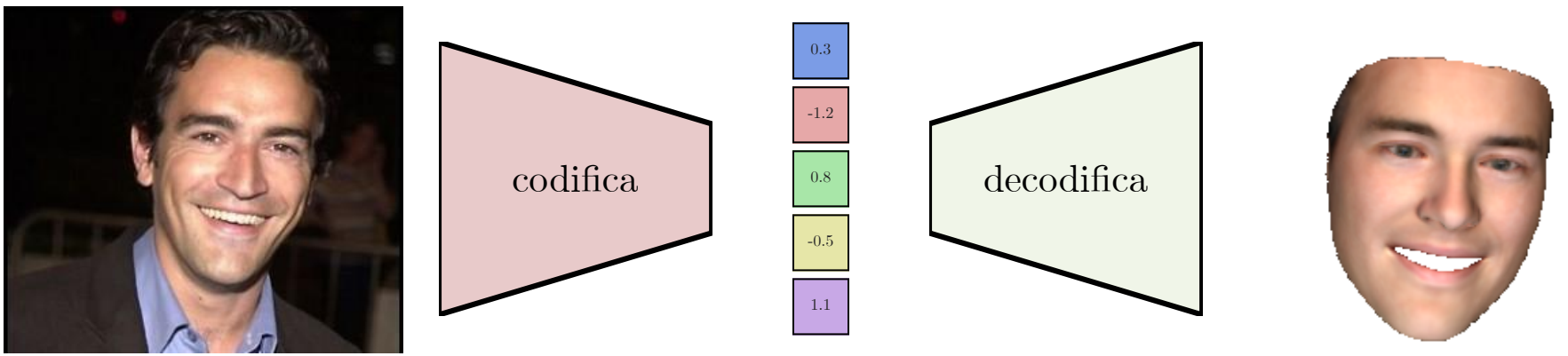
- Vantaggio: controllo preciso degli stimoli facciali
- Singoli neuroni rispondono lungo gli assi principali del 3DMM



Jozwik et al., Face dissimilarity judgments are predicted by representational distance in morphable and image-computable models

L'evoluzione dei 3DMM

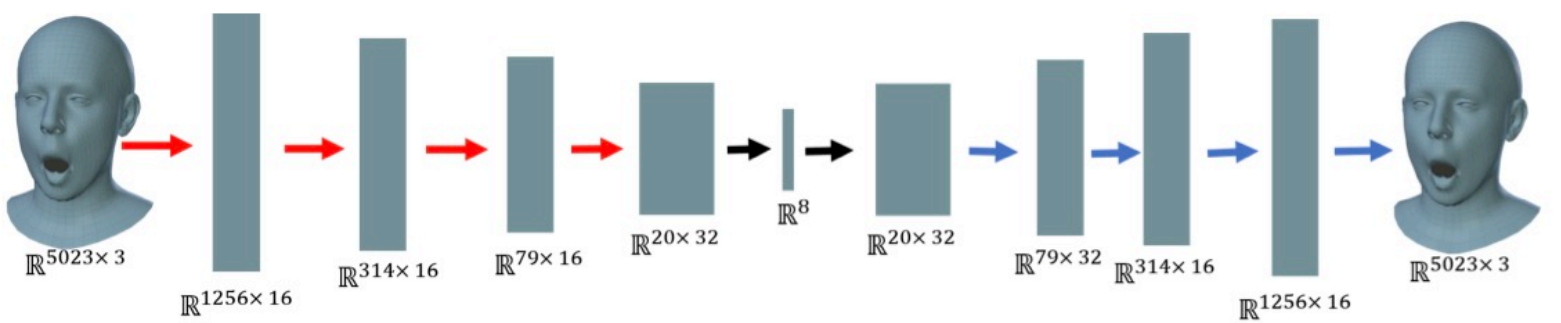
PCA: codifica e decodifica lineare, singolo hidden layer.



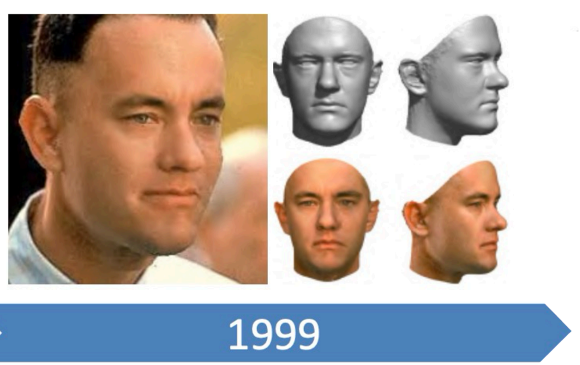
L'evoluzione dei 3DMM

PCA: codifica e decodifica lineare, singolo hidden layer.

Autoencoder: codifica e decodifica non lineare, numerosi hidden layers.



L'evoluzione dei 3DMM



1999

L'evoluzione dei 3DMM