A Morphable Model for the Synthesis of 3D Faces

R. Knothe, T. Vetter

May 11, 2009
A Morphable Model for the Synthesis of 3D Faces.
Blanz, Volker and Vetter, Thomas
SIGGRAPH’99 Conference Proceedings, pp. 187-194
Questions?
Questions?

- Is this a good paper?
Questions?

- Is this a good paper?
- What could you criticize?
Questions?

- Is this a good paper?
- What could you criticize?
- Was it interesting? (Is the approach interesting?)
Questions?

- Is this a good paper?
- What could you criticize?
- Was it interesting? (Is the approach interesting?)
- Is the paper understandable (reimplementable)?
Problem Definition
Problem Definition

Goal

generate 3D faces automatically from one (or more) photographs.
Database

- 200 heads of young adults
Database

- 200 heads of young adults
- 100 males / 100 females
Database

- 200 heads of young adults
- 100 males / 100 female
- Cylindrical representation, with radii $r(h, \phi)$ of surface points sampled at
  - 512 equally-spaced angles $\phi$ and
  - 512 equally spaced vertical steps $h$. 
Database

- 200 heads of young adults
- 100 males / 100 female
- Cylindrical representation, with radii \( r(h, \phi) \) of surface points sampled at
  - 512 equally-spaced angles \( \phi \) and
  - 512 equally spaced vertical steps \( h \).
- RGB-color values: \( R(h, \phi), G(h, \phi), B(h, \phi) \) vertex color.
Database

- 200 heads of young adults
- 100 males / 100 female
- Cylindrical representation, with radii $r(h, \phi)$ of surface points sampled at
  - 512 equally-spaced angles $\phi$ and
  - 512 equally spaced vertical steps $h$.
- RGB-color values: $R(h, \phi)$, $G(h, \phi)$, $B(h, \phi)$ vertex color.
- Face is represented by 70,000 vertices and the same number of color values.
Database

- 200 heads of young adults
- 100 males / 100 female
- cylindrical representation, with radii \( r(h, \phi) \) of surface points sampled at
  - 512 equally-spaced angles \( \phi \) and
  - 512 equally spaced vertical steps \( h \).
- RGB-color values: \( R(h, \phi), G(h, \phi), B(h, \phi) \) vertex color.
- Face is represented by 70,000 vertices and the same number of color values.
- Cyberware Laser scanner (each scan: \( \sim 15 \) sec)
Cylindrical Coordinates

rot(h,φ)
grün(h,φ)
blau(h,φ)
Radius(h,φ)
Database
Morphable 3D Face Model

Assumption

All the $m$ example faces are in full correspondence.
Morphable 3D Face Model

Assumption

All the \( m \) example faces are in full correspondence.

Geometry and Texture of a face

\[
S = (X_1, Y_1, Z_1, \ldots, X_n, Y_n, Z_n) \in \mathbb{R}^{3n}
\]

\[
T = (R_1, G_1, B_1, \ldots, R_n, G_n, B_n) \in \mathbb{R}^{3n}
\]
Morphable 3D Face Model

Assumption
All the $m$ example faces are in full correspondence.

Geometry and Texture of a face

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

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

New shapes $S_{\text{model}}$ and new textures $T_{\text{model}}$

..can be expressed in barycentric coordinates as a linear combination ($\sum_{i=1}^{m} a_i = \sum_{i=1}^{m} b_i = 1$):

$$S_{\text{model}} = \sum_{i=1}^{m} a_i S_i$$

$$T_{\text{model}} = \sum_{i=1}^{m} a_i T_i$$
Morphable 3D Face Model

Morphable Model: set faces \( (S_{\text{mod}}(\vec{a}), T_{\text{mod}}(\vec{b})) \) parametrized by coefficients \( \vec{a} \) and \( \vec{b} \).
Plausibility

- quantify the result in terms of plausibility of being a face
Plausibility

- quantify the result in terms of plausibility of being a face
- fit Multivariate normal distribution to data

  - Shape $\Delta S_i = S_i - \bar{S}$
  - Shape $\Delta T_i = T_i - \bar{T}$

\[ data compression: PCA \]

\[ S_{\text{model}} = \bar{S} + \sum_{i=1}^{m-1} \alpha_i s_i; T_{\text{model}} = \bar{T} + \sum_{i=1}^{m-1} \beta_i t_i \]

Probability for coefficient $\vec{\alpha}$:

\[ p(\vec{\alpha}) = e^{-\frac{1}{2} \sum_{i=1}^{m-1} (\alpha_i \sigma_i)^2} \]

with $\alpha_i^2$: eigenvalues of shape covariance matrix.
Plausibility

- quantify the result in terms of plausibility of being a face
- fit Multivariate normal distribution to data
  - Shape $\Delta S_i = S_i - \bar{S}$
  - Shape $\Delta T_i = T_i - \bar{T}$
- data compression: PCA

$$S_{model} = \bar{S} + \sum_{i=1}^{m-1} \alpha_i s_i; \quad T_{model} = \bar{T} + \sum_{i=1}^{m-1} \beta_i t_i$$
Plausibility

- quantify the result in terms of plausibility of being a face
- fit Multivariate normal distribution to data
  - Shape $\Delta S_i = S_i - \bar{S}$
  - Shape $\Delta T_i = T_i - \bar{T}$
- data compression: PCA

\[
S_{\text{model}} = \bar{S} + \sum_{i=1}^{m-1} \alpha_i s_i; \quad T_{\text{model}} = \bar{T} + \sum_{i=1}^{m-1} \beta_i t_i
\]

- Probability for coefficient $\bar{\alpha}$:

\[
p(\bar{\alpha}) = e^{-\frac{1}{2} \sum_{i=1}^{m-1} \left(\frac{\alpha_i}{\sigma_i}\right)^2}
\]

with $\alpha_i^2$: eigenvalues of shape covariance matrix.
Principal Components in Shape Space

1. PC.
2. PC.
Principal Components in Texture Space
Segmented Morphable Model
Segmented Morphable Model

- Define segments on reference face.
Segmented Morphable Model

- Define segments on reference face.
- Match entire face
Segmented Morphable Model

- Define segments on reference face.
- Match entire face
- Match segments separately.

R. Knothe, T. Vetter
Segmented Morphable Model

- Define segments on reference face.
- Match entire face
- Match segments separately.
- Blend results in 3D
Segmented Morphable Model

- Define segments on reference face.
- Match entire face
- Match segments separately.
- Blend results in 3D

Original  No Segments  With segments
Two scans of facial expressions of same individual

Transfer of Facial Expression

−  = smile

+ smile =
Transfer of Facial Expressions

original  surprise  joy  sadness

fear  anger  disgust
Facial Attributes

- Coefficients \( \alpha_i, \beta_i \) do not correspond to facial attributes
Facial Attributes

- Coefficients $\alpha_i, \beta_i$ do not correspond to facial attributes
- Facial expressions can be transferred by recording two scans of same individual with different expressions
Facial Attributes

- Coefficients $\alpha_i, \beta_i$ do not correspondent to facial attributes
- Facial expressions can be transferred by recording two scans of same individual with different expressions
- Attributes are more difficult to isolate
Facial Attributes

- Coefficients $\alpha_i, \beta_i$ do not correspond to facial attributes
- Facial expressions can be transferred by recording two scans of same individual with different expressions
- Attributes are more difficult to isolate
- Set of faces $(S_i, T_i)$ with manually assigned labels $\mu_i$
Facial Attributes

- Coefficients $\alpha_i, \beta_i$ do not correspond to facial attributes.
- Facial expressions can be transferred by recording two scans of the same individual with different expressions.
- Attributes are more difficult to isolate.
- Set of faces $(S_i, T_i)$ with manually assigned labels $\mu_i$.
- Multiple of $(\Delta S, \Delta T)$ can be added to individual face.

\[
\Delta S = \sum_{i=1}^{m} \mu_i (S_i - \bar{S}), \quad \Delta T = \sum_{i=1}^{m} \mu_i (T_i - \bar{T})
\]
Facial Attributes

- Coefficients $\alpha_i, \beta_i$ do not correspond to facial attributes
- Facial expressions can be transferred by recording two scans of same individual with different expressions
- Attributes are more difficult to isolate
- Set of faces $(S_i, T_i)$ with manually assigned labels $\mu_i$
- Multiple of $(\Delta S, \Delta T)$ can be added to individual face

$$
\Delta S = \sum_{i=1}^{m} \mu_i(S_i - \bar{S}), \quad \Delta T = \sum_{i=1}^{m} \mu_i(T_i - \bar{T})
$$

- Regression problem of estimating $\mu(S, T)$ (Assumption: linear function)
Facial Attributes

Attractiveness

Weight

Gender

original
Average Male and Female Faces

Average Male

Overall Average

Average Female

R. Knothe, T. Vetter

A Morphable Model for the Synthesis of 3D Faces
Caricatures

caricature by increasing the distance from average face
Matching a Morphable Model to images

Database

Modeler

Animation

Face Analyzer

3D Head

Tom Hanks!

Input Image

A Morphable Model for the Synthesis of 3D Faces
Analysis by Synthesis

3D World → Image → Image Description → Analysis → Image Model → Synthesis

model parameter

R. Knothe, T. Vetter
A Morphable Model for the Synthesis of 3D Faces
Analysis by Synthesis

3D World → Image → Analysis → Image Model → Synthesis

Image Description → model parameter

analysis-by-synthesis loop: renders an image and updates the parameters according to the residual difference
Matching a Morphable Model to images

<table>
<thead>
<tr>
<th>Model parameters</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\alpha_j; \ j \in {1, ..m - 1}$</td>
<td>shape coefficients</td>
</tr>
<tr>
<td>$\beta_j; \ j \in {1, ..m - 1}$</td>
<td>texture coefficients</td>
</tr>
<tr>
<td>$\vec{\rho}$</td>
<td>rendering parameters</td>
</tr>
<tr>
<td>$i_{amb}, i_{dir}$</td>
<td>intensity of ambient and directed light</td>
</tr>
<tr>
<td></td>
<td>color contrast, offset and gain</td>
</tr>
</tbody>
</table>
from \((\vec{\alpha}, \vec{\beta}, \vec{\rho})\) color images are rendered using
from $($\vec{\alpha}, \vec{\beta}, \vec{\rho})$) color images are rendered using

- perspective projection
from \((\vec{\alpha}, \vec{\beta}, \vec{\rho})\) color images are rendered using
- perspective projection
- Phong illumination model
from \((\vec{\alpha}, \vec{\beta}, \vec{\rho})\) color images are rendered using
  - perspective projection
  - Phong illumination model
reconstructed image is supposed to minimize
\[
E_I = \sum_{x,y} \| I_{\text{input}}(x, y) - I_{\text{model}}(x, y) \|^2
\]
from \((\vec{\alpha}, \vec{\beta}, \vec{\rho})\) color images are rendered using
- perspective projection
- Phong illumination model

reconstructed image is supposed to minimize

\[
E_I = \sum_{x,y} \| I_{\text{input}}(x, y) - I_{\text{model}}(x, y) \|^2
\]

ill-posedness
from $(\vec{\alpha}, \vec{\beta}, \vec{\rho})$ color images are rendered using
- perspective projection
- Phong illumination model

reconstructed image is supposed to minimize

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

ill-posedness

- many non-face-like surfaces lead to the same image
from \((\vec{\alpha}, \vec{\beta}, \vec{\rho})\) color images are rendered using
- perspective projection
- Phong illumination model

reconstructed image is supposed to minimize

\[
E_I = \sum_{x,y} \|I_{\text{input}}(x, y) - I_{\text{model}}(x, y)\|^2
\]

ill-posedness
- many non-face-like surfaces lead to the same image
- constraint on the set of solutions
Database / Morphable 3D Face Model
Matching a Morphable Model to images

- from $(\vec{\alpha}, \vec{\beta}, \vec{\rho})$ color images are rendered using
  - perspective projection
  - Phong illumination model
- reconstructed image is supposed to minimize

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

ill-posedness
- many non-face-like surfaces lead to the same image
- constraint on the set of solutions
- shape/texture vectors are restricted to the vector space spanned by the database

R. Knothe, T. Vetter
A Morphable Model for the Synthesis of 3D Faces
within the vector space of faces: further restrict solution.
• within the vector space of faces: further restrict solution.
• trade off: matching quality vs prior probability

Bayes decision theory: find a set of parameters $(\vec{\alpha},\vec{\beta},\vec{\rho})$ with maximal posterior probability, given an image $I$:

$$p(I \mid \vec{\alpha},\vec{\beta},\vec{\rho}) \sim e^{-\frac{1}{2\sigma^2}N \cdot E_I}$$

Maximum posterior probability is achieved by minimizing:

$$E = \frac{1}{\sigma^2}N \cdot E_I + m - 1 \sum_{j=1}^{m} \alpha_j^2 \sigma^2_S(j) + m - 1 \sum_{j=1}^{m} \beta_j^2 \sigma^2_T(j) + \sum_j (\rho_j - \bar{\rho}_j)^2 \sigma^2_{\rho,j}$$

R. Knothe, T. Vetter

A Morphable Model for the Synthesis of 3D Faces
within the vector space of faces: further restrict solution.

trade off: matching quality vs prior probability

Bayes decision theory: find a set of parameters () with maximal posterior probability, given an image $I_{\text{input}}$.
within the vector space of faces: further restrict solution.

- trade off: matching quality vs prior probability
- Bayes decision theory: find a set of parameters (\(\alpha, \beta, \rho\)) with maximal posterior probability, given an image \(I_{\text{input}}\)

Likelihood to observe \(I_{\text{input}}\) (with Gaussian noise; standard deviation \(\sigma_N\)):

\[
p(I_{\text{input}} | \vec{\alpha}, \vec{\beta}, \vec{\rho}) \sim e^{\frac{-1}{2\sigma_N^2}} \cdot E_I
\]
within the vector space of faces: further restrict solution.

trade off: matching quality vs prior probability

Bayes decision theory: find a set of parameters ($\alpha$, $\beta$, $\rho$) with maximal posterior probability, given an image $I_{\text{input}}$

Likelihood to observe $I_{\text{input}}$ (with Gaussian noise; standard deviation $\sigma_N$):

$$p(I_{\text{input}}|\vec{\alpha}, \vec{\beta}, \vec{\rho}) \sim e^{-\frac{1}{2\sigma_N^2} \cdot E_I}$$

Maximum posterior probability is achieved by minimizing

$$E = \frac{1}{\sigma^2_N} E_I + \sum_{j=1}^{m-1} \frac{\alpha^2_j}{\sigma_{S,j}^2} + \sum_{j=1}^{m-1} \frac{\beta^2_j}{\sigma_{T,j}^2} + \sum_j \frac{(\rho_j - \bar{\rho}_j)^2}{\sigma_{\rho,j}^2}$$
$I_{\text{phong}} = r_A \cdot I_A + r_D \cdot I_D \cdot \langle \hat{N}, \hat{L} \rangle + r_S \cdot I_S \cdot \langle \hat{R}, \hat{V} \rangle^m$

- $l_i$: light intensity (ambient/diffuse/specular)
- $r_i$: material (ambient/diffuse/specular)
- $m$: material shininess
Phong Illumination Model

\[ I_{r,\text{model},k} = (i_{r,\text{ambient}} + i_{r,\text{directed}} \cdot (n_k l)) \bar{R}_k + i_{r,\text{directed}} \cdot (r_k v_k)^\nu \]
Phong Illumination Model

$$I_{r, \text{model}, k} = (i_{r, \text{ambient}} + i_{r, \text{directed}} \cdot (n_k \cdot l)) \tilde{R}_k + i_{r, \text{directed}} \cdot (r_k v_k)^\nu$$

- Estimate of $E$ based on random selection of surface points.
Phong Illumination Model

\[ I_{r,\text{model},k} = (i_{r,\text{ambient}} + i_{r,\text{directed}} \cdot (n_k \cdot l)) \bar{R}_k + i_{r,\text{directed}} \cdot (r_k \cdot v_k)^\nu \]

- Estimate of \( E \) based on random selection of surface points.
- \((\bar{R}_k, \bar{G}_k, \bar{B}_k)\) texture in center of \( k \)th triangle
Phong Illumination Model

\[ I_{r,\text{model},k} = \left( i_{r,\text{ambient}} + i_{r,\text{directed}} \cdot (n_k \cdot l) \right) R_k + i_{r,\text{directed}} \cdot (r_k v_k)^\nu \]

- Estimate of \( E \) based on random selection of surface points.
- \((\bar{R}_k, \bar{G}_k, \bar{B}_k)\) texture in center of \( k \)th triangle
- \((\bar{p}_{x,y}, \bar{p}_{y,k})^T\) image location of center of \( k \)th triangle
Phong Illumination Model

\[ I_{r,model,k} = (i_{r,ambient} + i_{r,dircetd} \cdot (n_k l)) \bar{R}_k + i_{r,dircetd} \cdot (r_k v_k)^\nu \]

- Estimate of \( E \) based on random selection of surface points.
- \((\bar{R}_k, \bar{G}_k, \bar{B}_k)\) texture in center of \( k \)th triangle
- \((\bar{p}_{x,y}, \bar{p}_{y,k})^T\) image location of center of \( k \)th triangle
- \( n_k \) surface normal (at location of corners of triangle)
Phong Illumination Model

\[ I_{r,\text{model},k} = (i_{r,\text{ambient}} + i_{r,\text{directed}} \cdot (n_k \cdot l)) R_k + i_{r,\text{directed}} \cdot (r_k v_k) \]

- Estimate of \( E \) based on random selection of surface points.
- \((\bar{R}_k, \bar{G}_k, \bar{B}_k)\) texture in center of \( k \)th triangle
- \((\bar{p}_{x,y}, \bar{p}_{y,k})^T\) image location of center of \( k \)th triangle
- \( n_k \) surface normal (at location of corners of triangle)
- \( l \) direction of illumination
Phong Illumination Model

\[ \text{lr, model, } k = (\text{lr, ambient} + \text{lr, directd} \cdot (\text{n}_k \cdot \text{l})) \tilde{R}_k + \text{lr, directd} \cdot (\text{r}_k \cdot \text{v}_k) \]

- Estimate of \( E \) based on random selection of surface points.
- \( (\tilde{R}_k, \tilde{G}_k, \tilde{B}_k) \) texture in center of \( k \)th triangle
- \( (\tilde{p}_{x, \text{y}}, \tilde{p}_{y, \text{y}}, k) \) texture in center of \( k \)th triangle
- \( \text{n}_k \) surface normal (at location of corners of triangle)
- \( \text{l} \) direction of illumination
- \( \text{v}_k \) normalized difference of camera position and triangle’s center.
Phong Illumination Model

\[ I_{r, \text{model}, k} = (i_{r, \text{ambient}} + i_{r, \text{directed}} \cdot (n_k \cdot l)) \bar{R}_k + i_{r, \text{directed}} \cdot (r_k v_k)^\nu \]

- Estimate of \( E \) based on random selection of surface points.
- \((\bar{R}_k, \bar{G}_k, \bar{B}_k)\) texture in center of \( k \)th triangle
- \((\bar{p}_{x,y}, \bar{p}_{y,k})^T\) image location of center of \( k \)th triangle
- \( n_k \) surface normal (at location of corners of triangle)
- \( l \) direction of illumination
- \( v_k \) normalized difference of camera position and triangle’s center.
- \( r_k = 2(n \cdot l)n - l \) direction of the reflected ray
Phong Illumination Model

\[ I_{r, \text{model}, k} = (i_{r, \text{ambient}}) \bar{R}_k \]

- Estimate of \( E \) based on random selection of surface points.
- \((\bar{R}_k, \bar{G}_k, \bar{B}_k)\) texture in center of \( k \)th triangle
- \((\bar{p}_{x,y}, \bar{p}_{y,k})^T\) image location of center of \( k \)th triangle
- \( n_k \) surface normal (at location of corners of triangle)
- \( l \) direction of illumination
- \( v_k \) normalized difference of camera position and triangle’s center.
- \( r_k = 2(nl)n - l \) direction of the reflected ray
- if cast shadow: \( I_{r, \text{model}, k} = i_{r, \text{ambient}} \bar{R}_k \)
Computer Graphics Refresher: Cast Shadows
For high resolution 3D meshes, variations in $I_{\text{model}}$ across each triangle $k \in \{1, \ldots, n_t\}$ are small.
For high resolution 3D meshes, variations in $I_{\text{model}}$ across each triangle $k \in \{1, \ldots n_t\}$ are small.

$$E_l \approx \sum_{k=1}^{n_t} a_k \| I_{\text{input}}(\bar{p}_x,k, \bar{p}_y,k) - I_{\text{input},k} \|^2$$

where $a_k$ is the image area of triangle $k$. 
For high resolution 3D meshes, variations in $I_{\text{model}}$ across each triangle $k \in \{1, \ldots, n_t\}$ are small.

$$E_I \approx \sum_{k=1}^{n_t} a_k \| I_{\text{input}}(\bar{p}_x, k, \bar{p}_y, k) - I_{\text{input}, k} \|^2$$

where $a_k$ is the image area of triangle $k$.

**Gradient Descent**
For high resolution 3D meshes, variations in $I_{\text{model}}$ across each triangle $k \in \{1,..n_t\}$ are small.

$$E_l \approx \sum_{k=1}^{n_t} a_k \| I_{\text{input}}(\bar{p}_x,k,\bar{p}_y,k) - I_{\text{input},k} \|^2$$

where $a_k$ is the image area of triangle $k$.

**Gradient Descent**

- select a random subset $\mathcal{K} \subset \{1,..n_t\}$ of 40 triangles for each iteration
For high resolution 3D meshes, variations in $I_{\text{model}}$ across each triangle $k \in \{1, \ldots n_t\}$ are small.

$$E_I \approx \sum_{k=1}^{n_t} a_k \| I_{\text{input}}(\bar{p}_x, k, \bar{p}_y, k) - I_{\text{input}, k} \|^2$$

where $a_k$ is the image area of triangle $k$.

**Gradient Descent**

- select a random subset $\mathcal{K} \subset \{1, \ldots n_t\}$ of 40 triangles for each iteration
- probability of selecting $k$ is $p(k \in \mathcal{K}) \sim a_k$
For high resolution 3D meshes, variations in $I_{\text{model}}$ across each triangle $k \in \{1,..n_t\}$ are small.

\[
E_l \approx \sum_{k=1}^{n_t} a_k \| I_{\text{input}}(\bar{p}_x,k, \bar{p}_y,k) - I_{\text{input},k} \|^2
\]

where $a_k$ is the image area of triangle $k$.

**Gradient Descent**

- select a random subset $\mathcal{K} \subset \{1,..n_t\}$ of 40 triangles for each iteration
- probability of selecting $k$ is $p(k \in \mathcal{K}) \sim a_k$
- helps to avoid local minima by adding noise to gradient estimate
For high resolution 3D meshes, variations in $I_{\text{model}}$ across each triangle $k \in \{1, \ldots n_t\}$ are small.

$$E_I \approx \sum_{k=1}^{n_t} a_k \| I_{\text{input}}(\bar{p}_x, k, \bar{p}_y, k) - I_{\text{input}, k} \|^2$$

where $a_k$ is the image area of triangle $k$.

**Gradient Descent**

- select a random subset $\mathcal{K} \subset \{1, \ldots n_t\}$ of 40 triangles for each iteration
- probability of selecting $k$ is $p(k \in \mathcal{K}) \sim a_k$
- helps to avoid local minima by adding noise to gradient estimate

$$E_{\mathcal{K}} = \sum_{k \in \mathcal{K}} a_k \| I_{\text{input}}(\bar{p}_x, k, \bar{p}_y, k) - I_{\text{input}, k} \|^2$$
Gradient Descent

every 1000 steps:

1. Compute the full 3D shape of the current model.
2. Calculate the 2D positions (px, py) for each triangle.
3. Compute the triangle area $a_k$ for hidden surfaces and cast shadows using a two-pass z-buffer technique.

The parameters are updated as follows:

\[
\alpha_j \rightarrow \alpha_j - \lambda \cdot \delta E / \delta \alpha_j
\]

\[
\beta_j \rightarrow \beta_j - \lambda \cdot \delta E / \delta \beta_j
\]

\[
\rho_j \rightarrow \rho_j - \lambda \cdot \delta E / \delta \rho_j
\]

with suitable factors $\lambda_j$. 

R. Knothe, T. Vetter
A Morphable Model for the Synthesis of 3D Faces
Gradient Descent

every 1000 steps:
- compute full 3D shape of current model
Gradient Descent

every 1000 steps:

- compute full 3D shape of current model
- 2D positions \((p_x, p_y)^T\)
Gradient Descent

every 1000 steps:

- compute full 3D shape of current model
- 2D positions \((p_x, p_y)^T\)
- triangle area \(a_k\), hidden surfaces

\[
\alpha_j \rightarrow \alpha_j - \lambda \cdot \delta E \delta \alpha_j \\
\beta_j \rightarrow \beta_j - \lambda \cdot \delta E \delta \beta_j \\
\rho_j \rightarrow \rho_j - \lambda \cdot \delta E \delta \rho_j
\]

with suitable factors \(\lambda_j\)

R. Knothe, T. Vetter
A Morphable Model for the Synthesis of 3D Faces
Gradient Descent

every 1000 steps:

- compute full 3D shape of current model
- 2D positions \((p_x, p_y)^T\)
- triangle area \(a_k\), hidden surfaces
- cast shadows (two-pass z-buffer technique)
Gradient Descent

every 1000 steps:

- compute full 3D shape of current model
- 2D positions \((p_x, p_y)^T\)
- triangle area \(a_k\), hidden surfaces
- cast shadows (two-pass z-buffer technique)

The parameters are updated:

\[
\begin{align*}
\alpha_j &\rightarrow \alpha_j - \lambda \cdot \delta E_{\delta\alpha_j} \\
\beta_j &\rightarrow \beta_j - \lambda \cdot \delta E_{\delta\beta_j} \\
\rho_j &\rightarrow \rho_j - \lambda \cdot \delta E_{\delta\rho_j}
\end{align*}
\]
Gradient Descent

**every 1000 steps:**
- compute full 3D shape of current model
- 2D positions \((p_x, p_y)^T\)
- triangle area \(a_k\), hidden surfaces
- cast shadows (two-pass z-buffer technique)

The parameters are updated:
- depending on analytical derivatives of cost function \(E\)
Gradient Descent

every 1000 steps:
- compute full 3D shape of current model
- 2D positions \( (p_x, p_y)^T \)
- triangle area \( a_k \), hidden surfaces
- cast shadows (two-pass z-buffer technique)

The parameters are updated:
- depending on analytical derivatives of cost function \( E \)
  \[ \alpha_j \rightarrow \alpha_j - \lambda \cdot \frac{\delta E}{\delta \alpha_j} \]
Gradient Descent

every 1000 steps:

- compute full 3D shape of current model
- 2D positions \((p_x, p_y)^T\)
- triangle area \(a_k\), hidden surfaces
- cast shadows (two-pass z-buffer technique)

The parameters are updated:

- depending on analytical derivatives of cost function \(E\)
- \(\alpha_j \rightarrow \alpha_j - \lambda \cdot \frac{\delta E}{\delta \alpha_j}\)
- \(\beta_j \rightarrow \beta_j - \lambda \cdot \frac{\delta E}{\delta \beta_j}\)
Gradient Descent

every 1000 steps:

- compute full 3D shape of current model
- 2D positions \((p_x, p_y)^T\)
- triangle area \(a_k\), hidden surfaces
- cast shadows (two-pass z-buffer technique)

The parameters are updated:

- depending on analytical derivatives of cost function \(E\)
- \(\alpha_j \rightarrow \alpha_j - \lambda \cdot \frac{\delta E}{\delta \alpha_j}\)
- \(\beta_j \rightarrow \beta_j - \lambda \cdot \frac{\delta E}{\delta \beta_j}\)
- \(\rho_j \rightarrow \rho_j - \lambda \cdot \frac{\delta E}{\delta \rho_j}\)
Gradient Descent

every 1000 steps:
- compute full 3D shape of current model
- 2D positions \((p_x, p_y)^T\)
- triangle area \(a_k\), hidden surfaces
- cast shadows (two-pass z-buffer technique)

The parameters are updated:
- depending on analytical derivatives of cost function \(E\)
  \[ \alpha_j \rightarrow \alpha_j - \lambda \cdot \frac{\delta E}{\delta \alpha_j} \]
  \[ \beta_j \rightarrow \beta_j - \lambda \cdot \frac{\delta E}{\delta \beta_j} \]
  \[ \rho_j \rightarrow \rho_j - \lambda \cdot \frac{\delta E}{\delta \rho_j} \]
- with suitable factors \(\lambda_j\)
Derivatives

From

$$E_K = \sum_{k \in K} a_k \| I_{\text{input}}(\bar{p}_x, k, \bar{p}_y, k) - I_{\text{input}, k} \|^2$$

derivatives $\frac{\delta E_K}{\delta \alpha_j}$, $\frac{\delta E_K}{\delta \beta_j}$, $\frac{\delta E_K}{\delta \rho_j}$ can be obtained.
Derivatives

- From

\[ E_K = \sum_{k \in K} a_k \| I_{\text{input}}(\vec{p}_x, k, \vec{p}_y, k) - I_{\text{input}, k} \|^2 \]

derivatives \( \frac{\delta E_K}{\delta \alpha_j}, \frac{\delta E_K}{\delta \beta_j}, \frac{\delta E_K}{\delta \rho_j} \) can be obtained.

- derivatives of texture and shape yield derivatives of 2D locations, surface normals \( n_k \), vectors \( v_k \) and \( r_k \) and \( I_{\text{model}, k} \).
to avoid local minima, coarse-to-fine strategy in several respects:
Coarse-to-Fine

to avoid local minima, coarse-to-fine strategy in several respects:

- first iterations on down-sampled input image
Coarse-to-Fine

to avoid local minima, coarse-to-fine strategy in several respects:

- first iterations on down-sampled input image
- start only with the first $\alpha_i, \beta_i$
Coarse-to-Fine

to avoid local minima, coarse-to-fine strategy in several respects:
  • first iterations on down-sampled input image
  • start only with the first $\alpha_i, \beta_i$
  • start with large $\sigma_N$ (strong weight on prior probability)
Coarse-to-Fine

to avoid local minima, coarse-to-fine strategy in several respects:

- first iterations on down-sampled input image
- start only with the first $\alpha_i, \beta_i$
- start with large $\sigma_N$ (strong weight on prior probability)
- in the last iterations: break the face model into segments
Illumination-Corrected Texture Extraction

- specific features are not captured by the Morphable Model
Illumination-Corrected Texture Extraction

- specific features are not captured by the Morphable Model
- Texture Extraction
Illumination-Corrected Texture Extraction

- specific features are not captured by the Morphable Model
- Texture Extraction
- separate pure albedo
  - shading
  - cast shadows
Illumination-Corrected Texture Extraction

- specific features are not captured by the Morphable Model
- Texture Extraction
- separate pure albedo
  - shading
  - cast shadows
- to be able to change pose and illumination
Illumination-Corrected Texture Extraction

- specific features are not captured by the Morphable Model
- Texture Extraction
- separate pure albedo
  - shading
  - cast shadows
- to be able to change pose and illumination
- Problems?
  - partial occlusion (e.g. Hair)
  - accuracy of shape reconstruction at contours
  - accuracy of illumination reconstruction (cast shadows)
Illumination-Corrected Texture Extraction

2D input | rough initialization | automated 3D shape and texture reconstruction | illumination corrected extraction | texture
Audrey Hepburn (1929 - 1993) was an Academy Award-winning actress of film and theater, Broadway stage performer, ballerina, fashion model, and humanitarian.
Tom Hanks
Mona Lisa, or La Gioconda. (La Joconde), is a 16th century oil painting on poplar wood by Leonardo da Vinci, and is one of the most famous paintings in the world. Few other works of art have been subject to as much scrutiny, study, mythologizing and parody. It is owned by the French government and hangs in the Musée du Louvre in Paris. The painting, a half-length portrait, depicts a woman whose gaze meets the viewer’s with an expression often described as enigmatic. It is considered to be Leonardo’s magnum opus.
Mona Lisa
Video

watch the Morphable Model in action