Faces and Image-Based Lighting

Digital Visual Effects

Yung-Yu Chuang

with slides by Richard Szeliski, Steve Seitz, Alex Efros, Li-Yi Wei and Paul Debevec

Outline

- Image-based lighting

- 3D acquisition for faces
- Statistical methods (with application to face super-resolution)
- 3D Face models from single images
- Image-based faces
- Relighting for faces

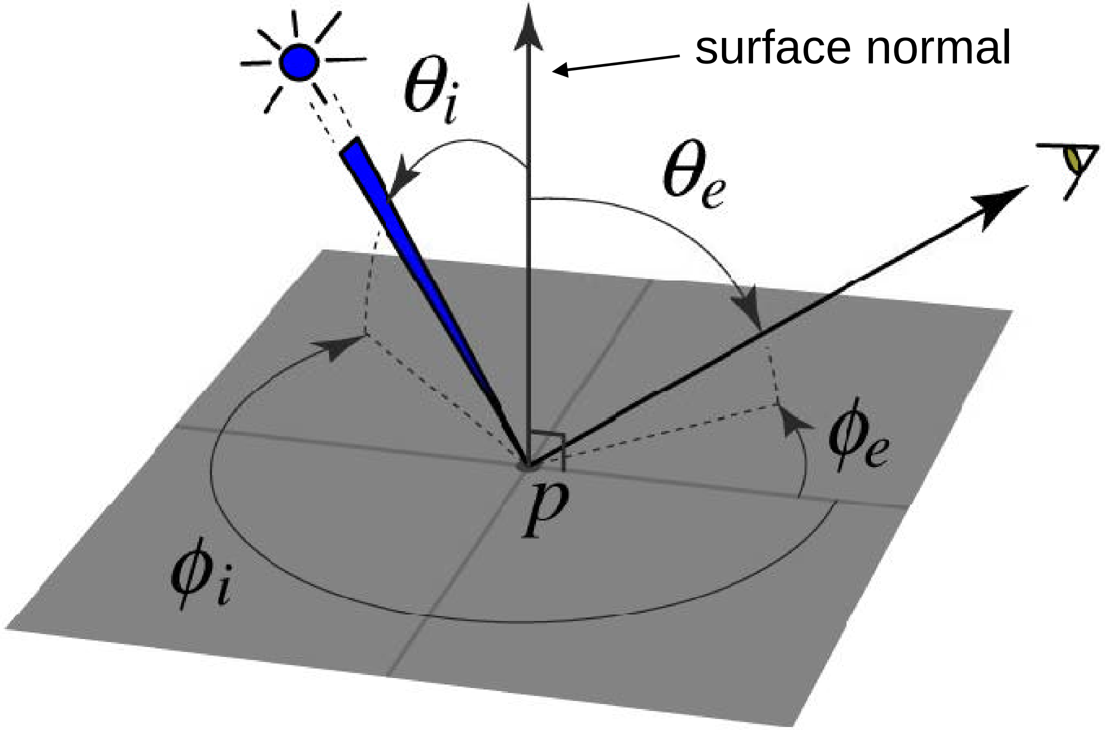
Image-based lighting

Rendering

- Rendering is a function of geometry, reflectance, lighting and viewing.
- To synthesize CGI into real scene, we have to match the above four factors.
- Viewing can be obtained from *calibration* or *structure from motion*.
- Geometry can be captured using *3D photography* or made by hands.
- How to capture lighting and reflectance?

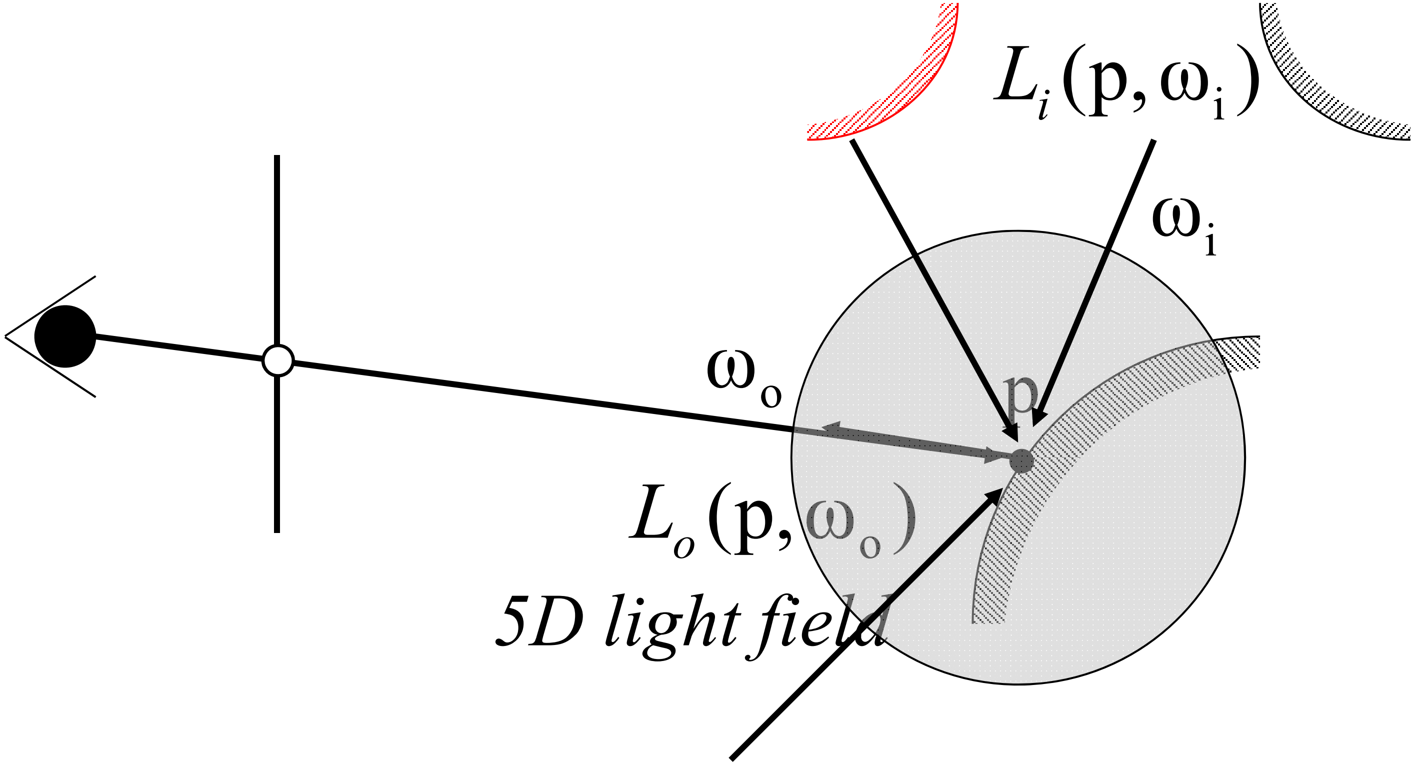
Reflectance

- The Bidirectional Reflection Distribution Function
  - Given an incoming ray (θ_i, ϕ_i) and outgoing ray (θ_e, ϕ_e) what proportion of the incoming light is reflected along out



Answer given by the BRDF: $\rho(\theta_i, \phi_i, \theta_e, \phi_e)$

Rendering equation



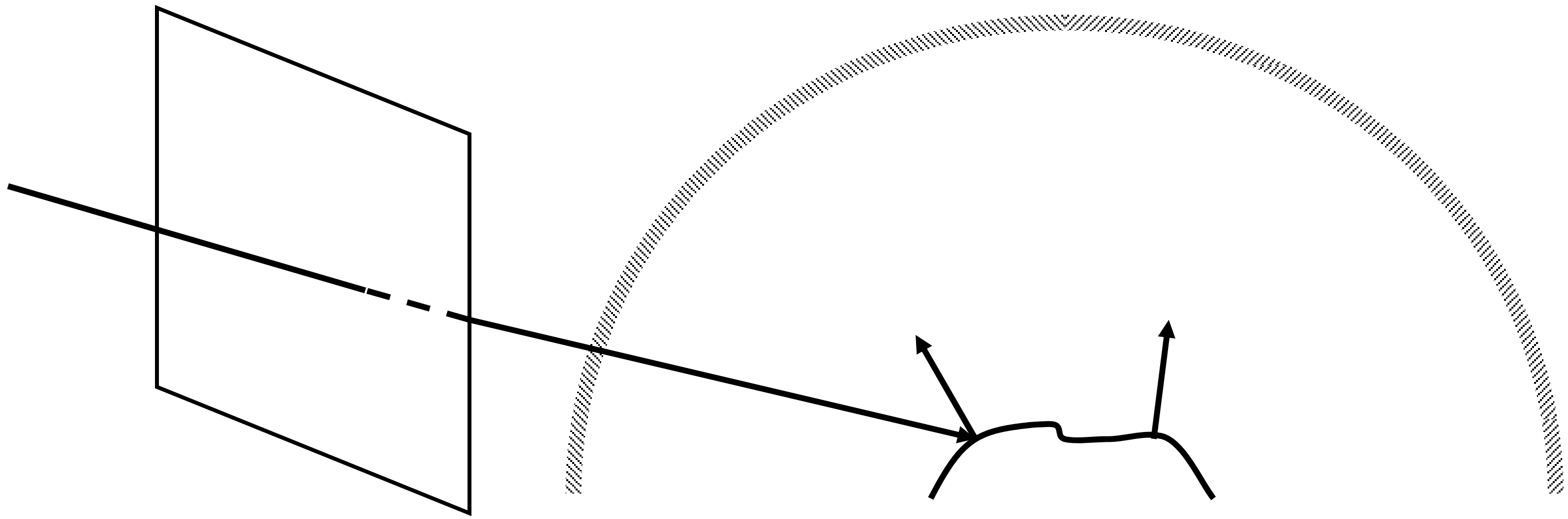
$$L_o(\mathrm{p}, \omega_o) = L_e(\mathrm{p}, \omega_o) + \int_{s^2} \rho(\mathrm{p}, \omega_o, \omega_i) L_i(\mathrm{p}, \omega_i) \left|\cos\theta_i\right| d\omega_i$$

Complex illumination

$$L_o(\mathrm{p},\omega_o) = L_e(\mathrm{p},\omega_o) + \int_{s^2} f(\mathrm{p},\omega_o,\omega_i) L_i(\mathrm{p},\omega_i) \left|\cos\theta_i\right| d\omega_i$$

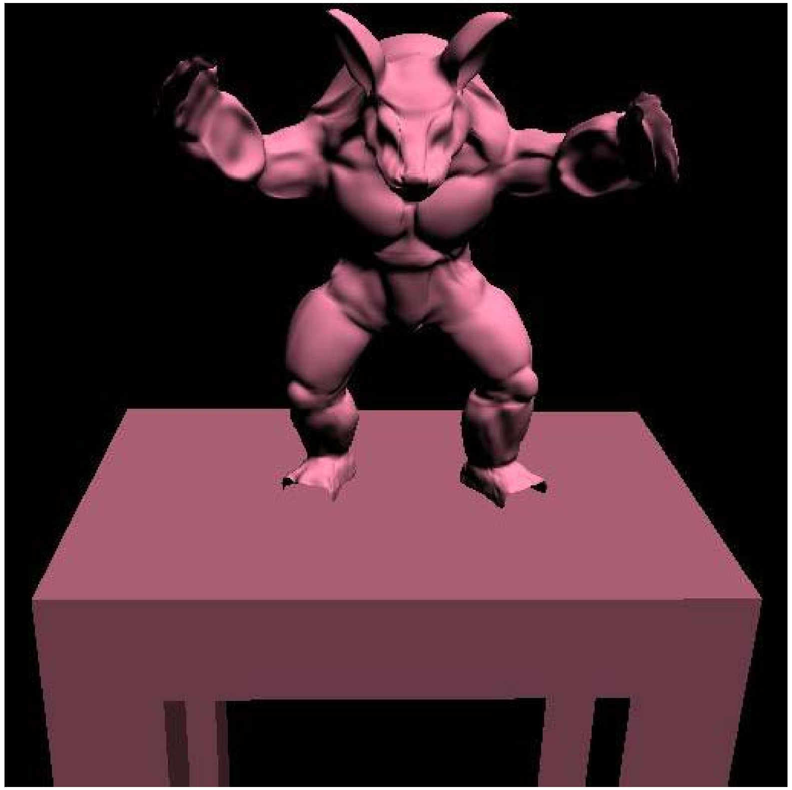
$$B(\mathrm{p},\omega_o) = \int_{s^2} f(\mathrm{p},\omega_o,\omega_i) L_d(\mathrm{p},\omega_i) \left|\cos\theta_i\right| d\omega_i$$

$$B_p(\omega_o) = \int_{s^2} f_{p,\omega_o}(\omega_i) L_d(\omega_i) \left|\cos\theta_i\right| d\omega_i$$



Point lights

Classically, rendering is performed assuming point light sources



directional source

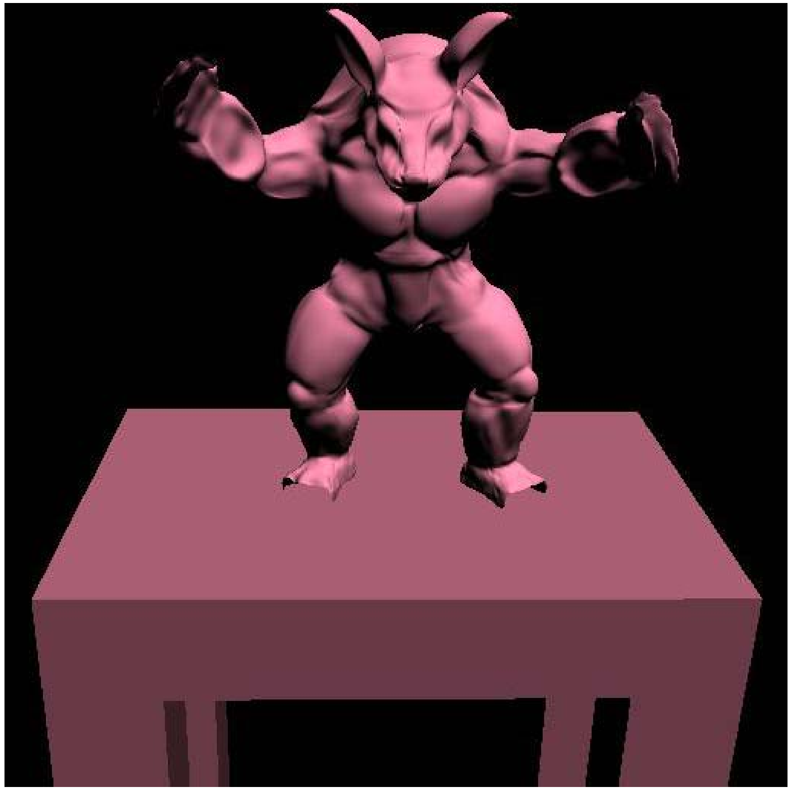
Natural illumination

People perceive materials more easily under natural illumination than simplified illumination.

Images courtesy Ron Dror and Ted Adelson

Natural illumination

Rendering with natural illumination is more expensive compared to using simplified illumination



directional source

natural illumination

Environment maps

Miller and Hoffman, 1984

HDR lighting

Examples of complex environment light

Examples of complex environment light

Complex illumination

$$L_o(\mathrm{p},\omega_\mathrm{o}) = L_e(\mathrm{p},\omega_\mathrm{o}) + \int_{s^2} f(\mathrm{p},\omega_\mathrm{o},\omega_\mathrm{i}) L_i(\mathrm{p},\omega_\mathrm{i}) \left|\cos\theta_\mathrm{i}\right| d\omega_\mathrm{i}$$

$$B(\mathrm{p},\omega_\mathrm{o}) = \int_{s^2} f(\mathrm{p},\omega_\mathrm{o},\omega_\mathrm{i}) L_d(\mathrm{p},\omega_\mathrm{i}) \left|\cos\theta_\mathrm{i}\right| d\omega_\mathrm{i}$$

$$B_p(\omega_\mathrm{o}) = \int_{s^2} \underbrace{f_{p,\omega_\mathrm{o}}}_{\text{reflectance}}(\omega_\mathrm{i}) \underbrace{L_d}_{\text{lighting}}(\omega_\mathrm{i}) \left|\cos\theta_\mathrm{i}\right| d\omega_\mathrm{i}$$

Both are spherical functions

Function approximation

- G(x): the function to approximate
- $B_1(x)$, $B_2(x)$, … $B_n(x)$: basis functions
- We want

$$G(x) = \sum_{i=1}^{n} c_i B_i(x)$$

- Storing a finite number of coefficients c_i gives an approximation of G(x)

Function approximation

- How to find coefficients c_i?
  - Minimize an error measure
- What error measure?
  - L_2 error

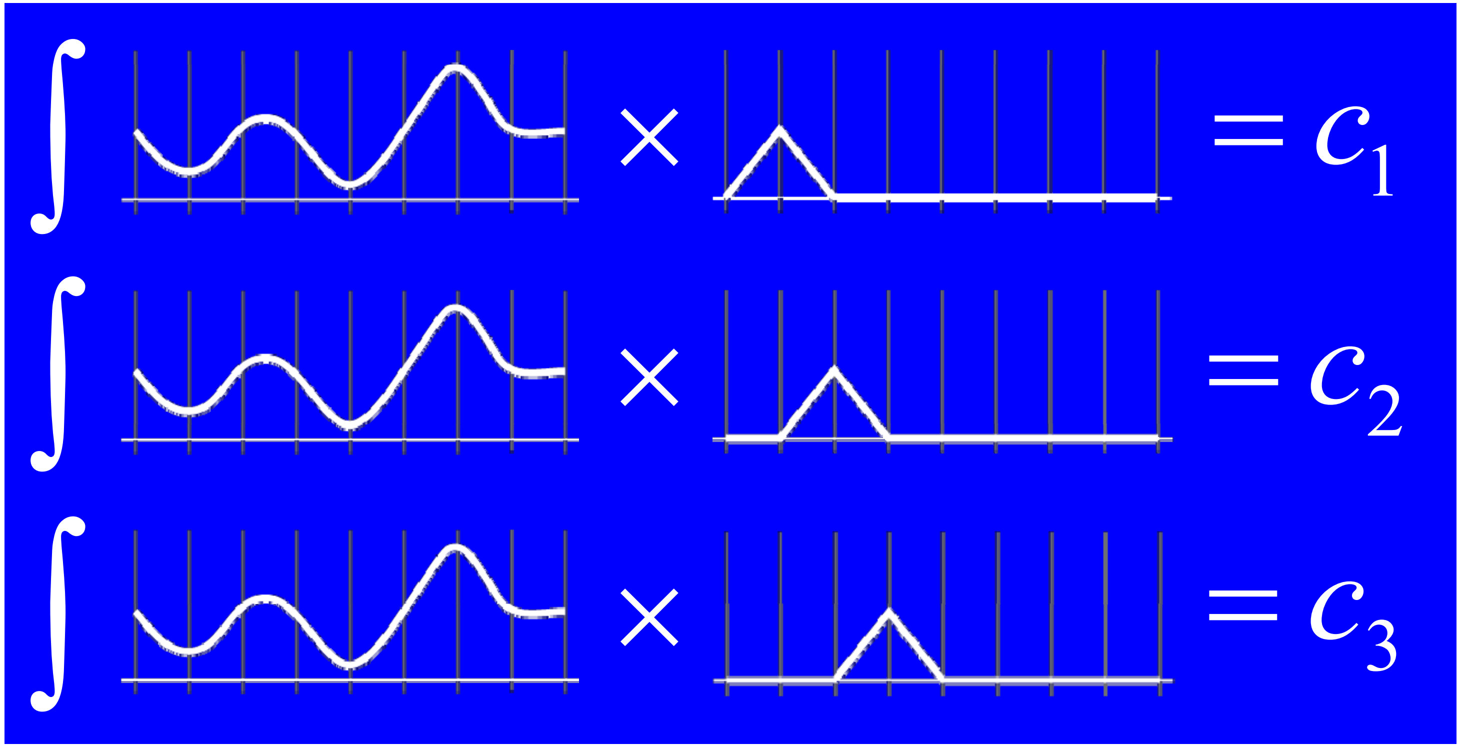
$$E_{L_2} = \int_I [G(x) - \sum_i c_i B_i(x)]^2$$

- Coefficients

$$c_i = \langle G | B_i \rangle = \int_X G(x) B_i(x) dx$$

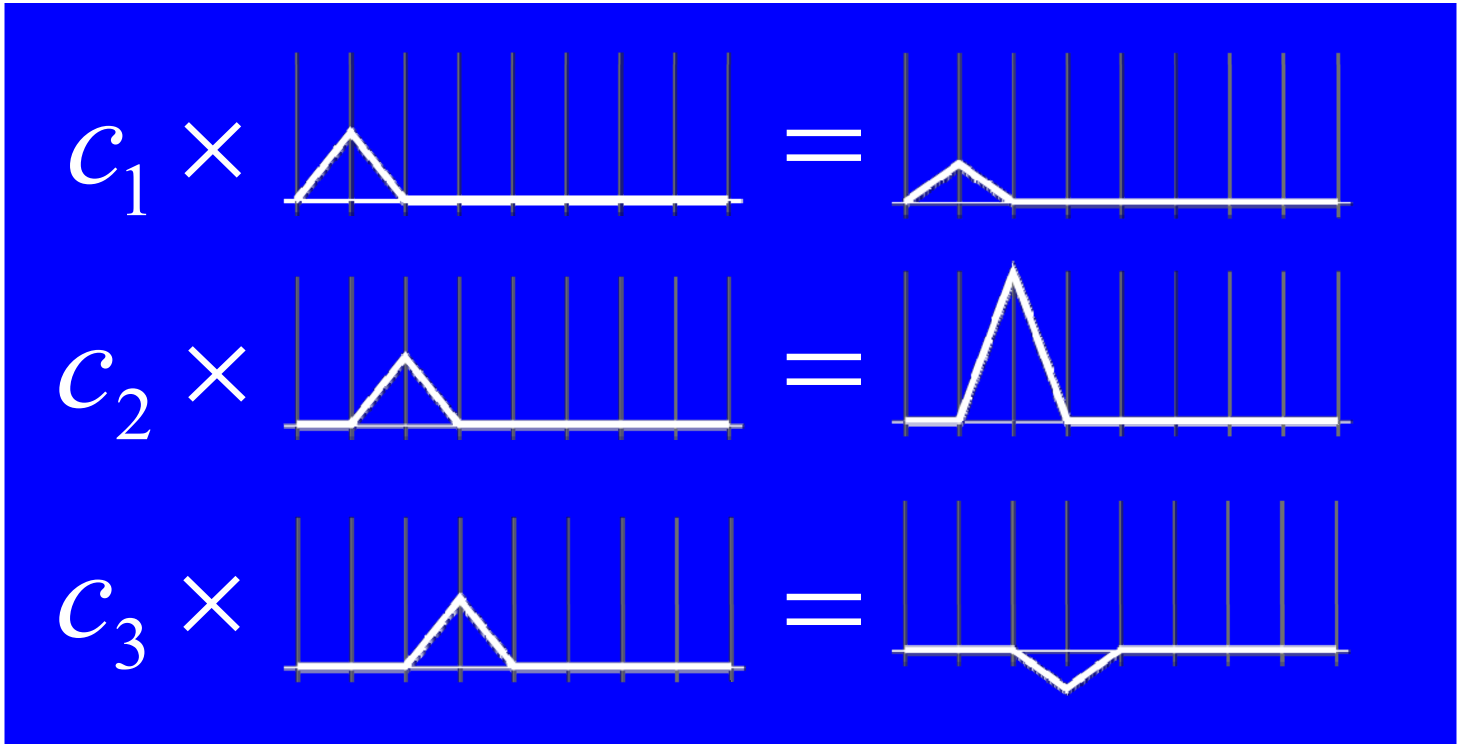
Function approximation

- Basis Functions are pieces of signal that can be used to produce approximations to a function



Function approximation

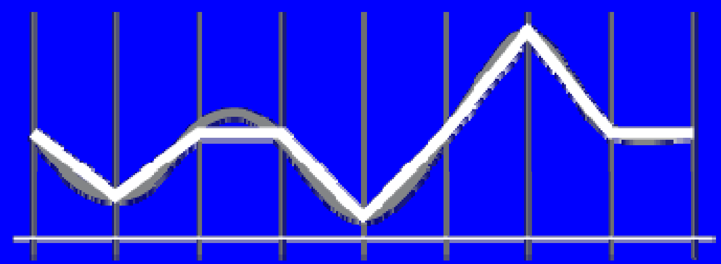
- We can then use these coefficients to reconstruct an approximation to the original signal



Function approximation

- We can then use these coefficients to reconstruct an approximation to the original signal

$$\sum_{i=1}^{N} c_i B_i(x) =$$



Orthogonal basis functions

- Orthogonal Basis Functions
  - These are families of functions with special properties

$$\int B_i(x)B_j(x)\,dx = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases}$$

  - Intuitively, it's like functions don't overlap each other's footprint
    - A bit like the way a Fourier transform breaks a functions into component sine waves

Integral of product

$$I = \int F(x)G(x)\,dx$$

$$F(x) = \sum_i f_i B_i(x) \qquad G(x) = \sum_j g_j B_j(x)$$

$$\int F(x)G(x)\,dx = \int \left(\sum_i f_i B_i(x) \sum_j g_j B_j(x) \right) dx$$

$$= \int \sum_i \sum_j f_i g_j B_i(x) B_j(x)\,dx = \int \sum_i f_i g_i\,dx = \hat{F} \cdot \hat{G}$$

$$B_p(\omega_o) = \int_{s^2} f_{p,\omega_o}(\omega_i) L_d(\omega_i) \left|\cos\theta_i\right| d\omega_i$$

Basis functions

- Transform data to a space in which we can capture the essence of the data better
- Spherical harmonics, similar to Fourier transform in spherical domain, is used in PRT.

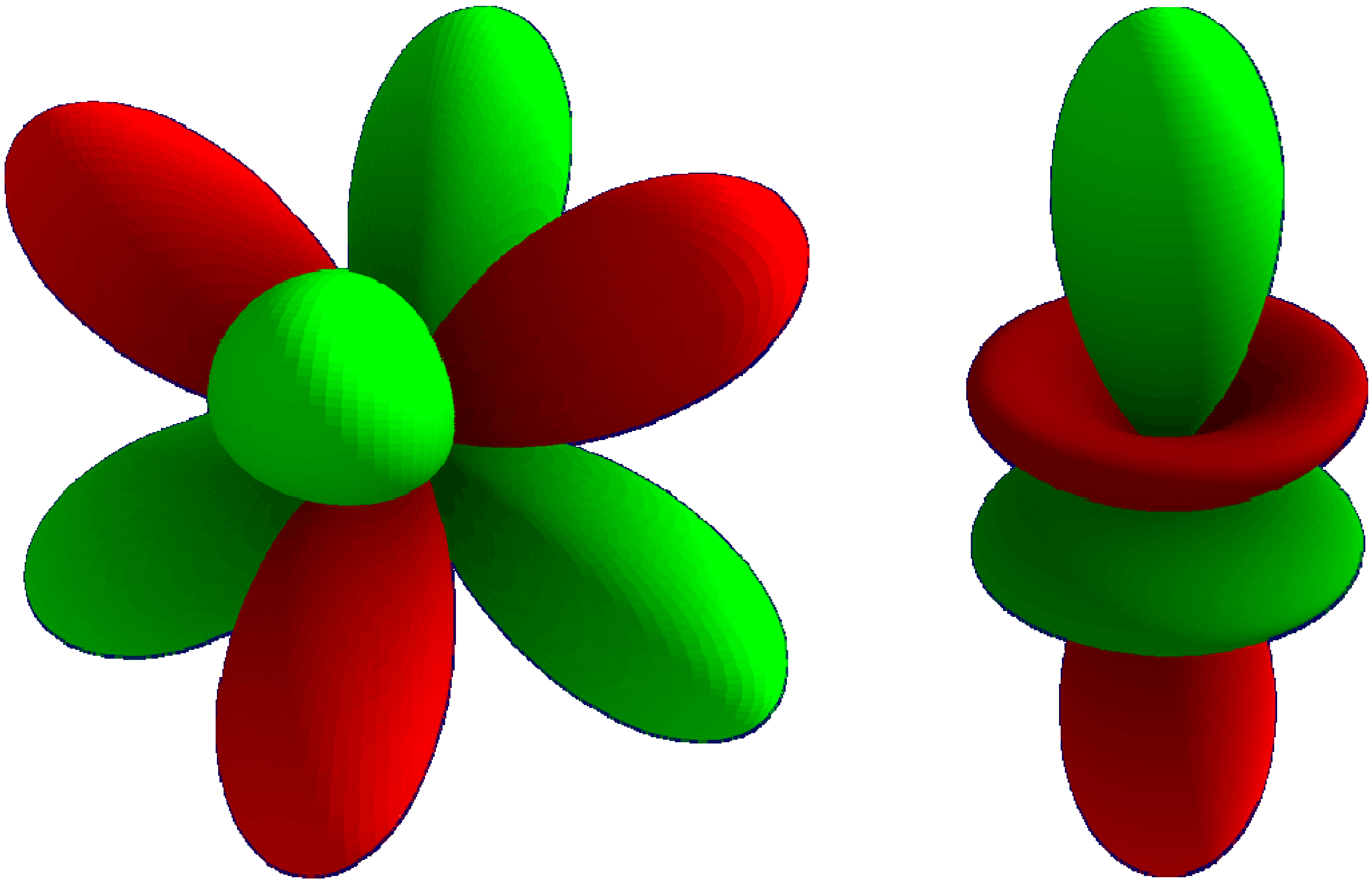
Real spherical harmonics

- A system of signed, orthogonal functions over the sphere
- Represented in spherical coordinates by the function

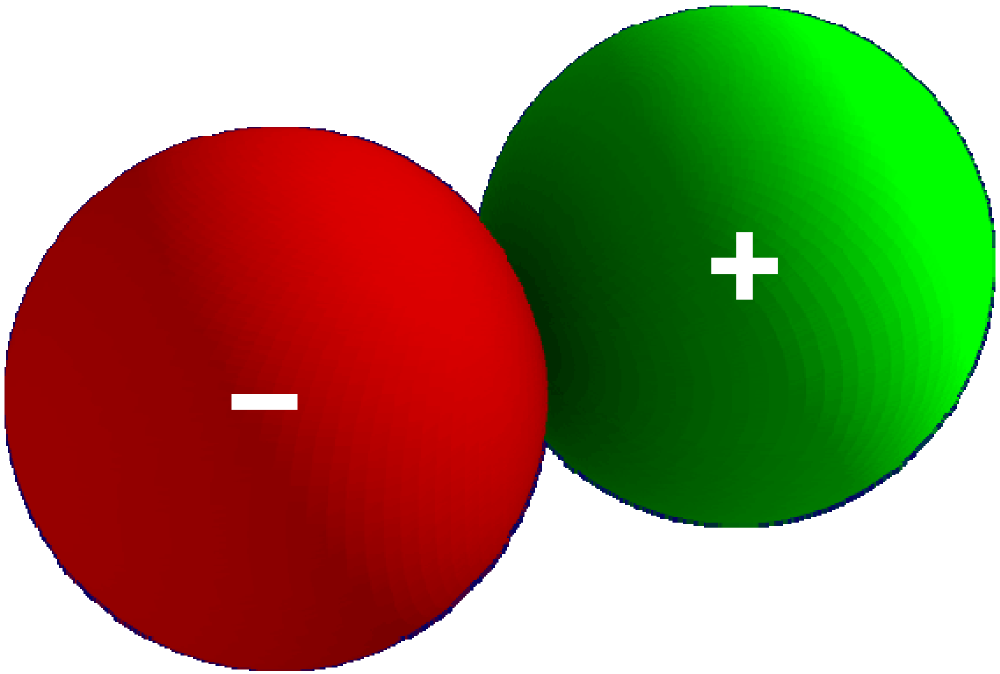
$$y_l^m(\theta,\varphi)=\begin{cases}\sqrt{2}K_l^m\cos(m\varphi)P_l^m(\cos\theta), & m>0\\ \sqrt{2}K_l^m\sin(-m\varphi)P_l^{-m}(\cos\theta), & m<0\\ K_l^0P_l^0(\cos\theta), & m=0\end{cases}$$

where l is the band and m is the index within the band

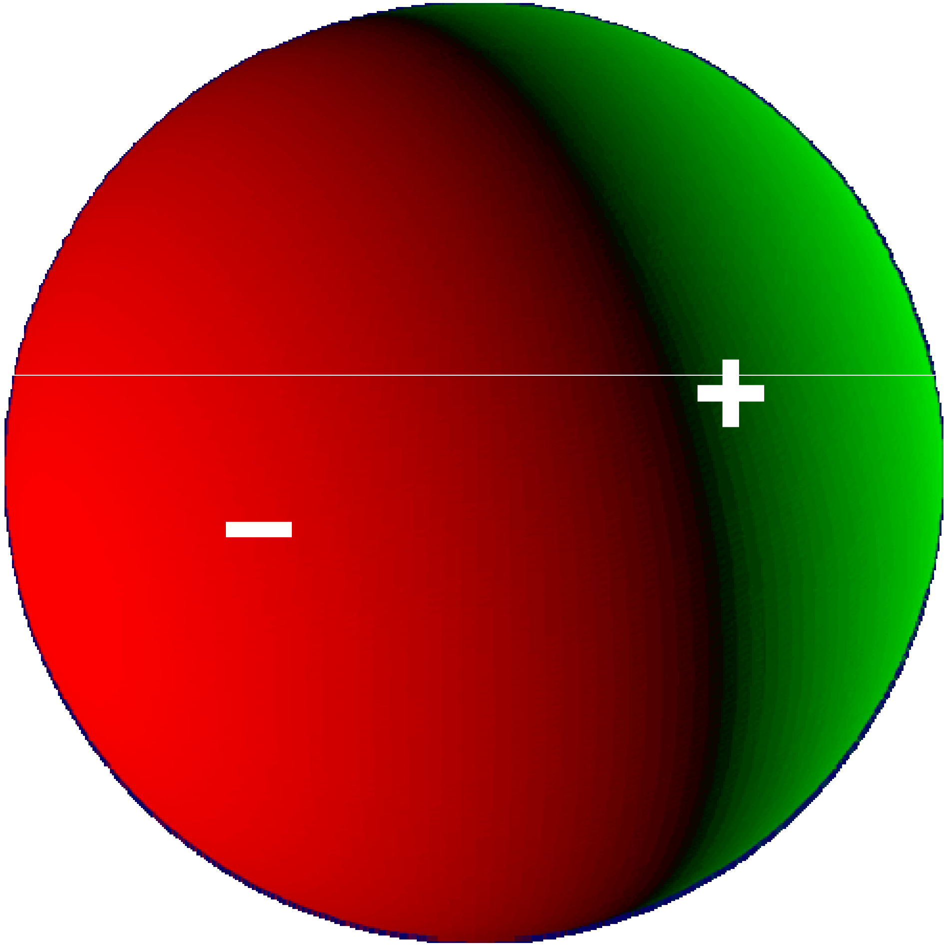
Real spherical harmonics



Reading SH diagrams

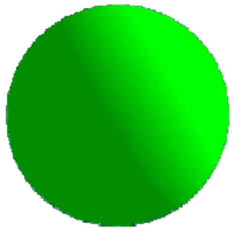


Reading SH diagrams

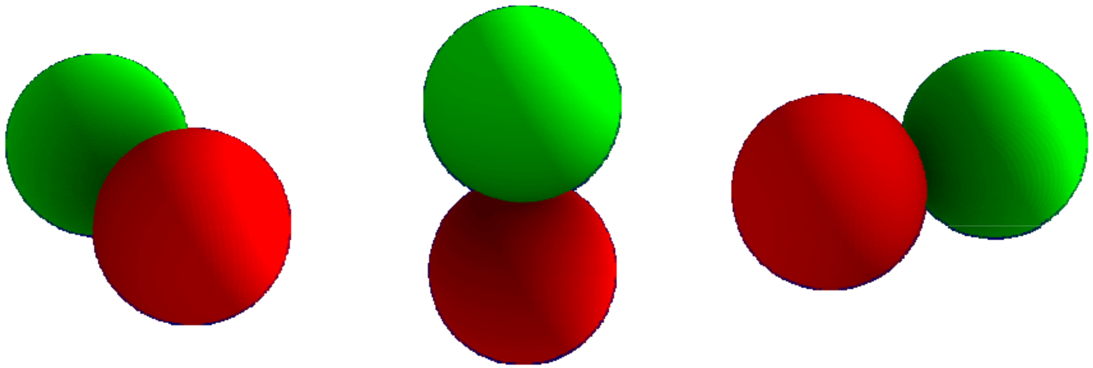


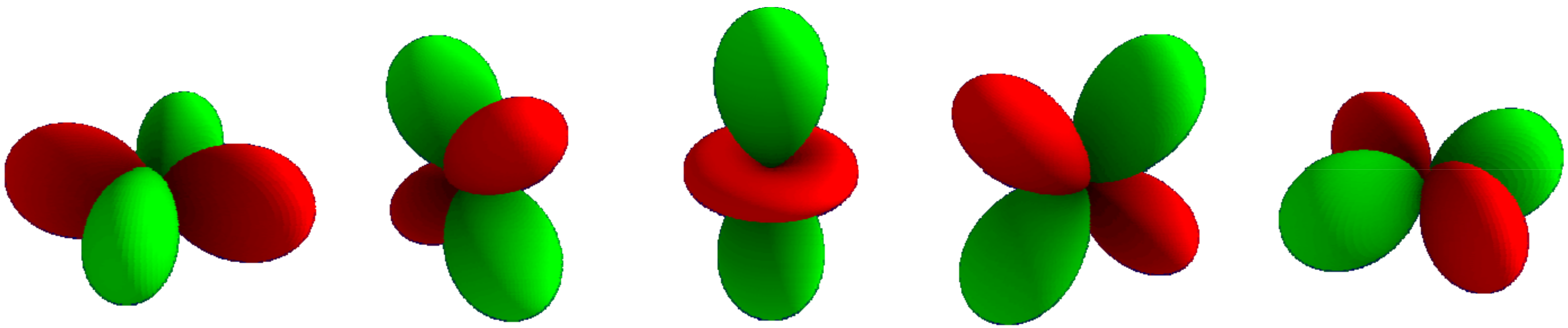
The SH functions

$$y_0^0 =$$

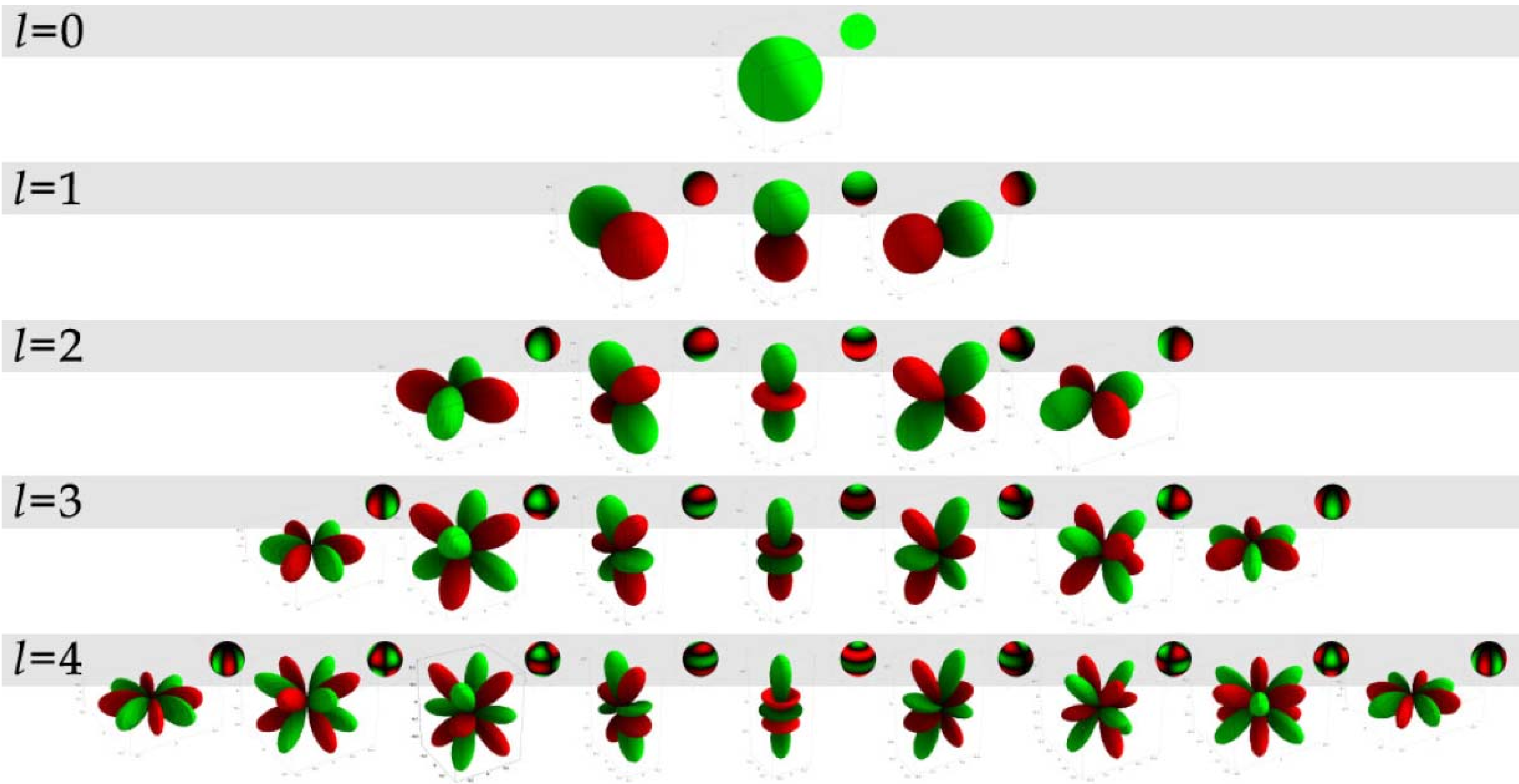


$$y_1^{-1} =$$





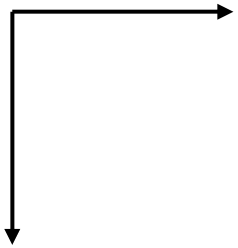
The SH functions

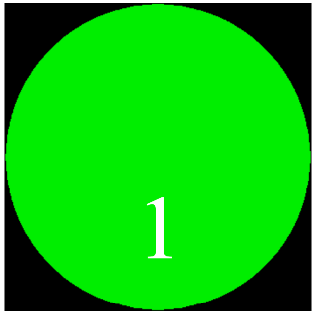


Spherical harmonics

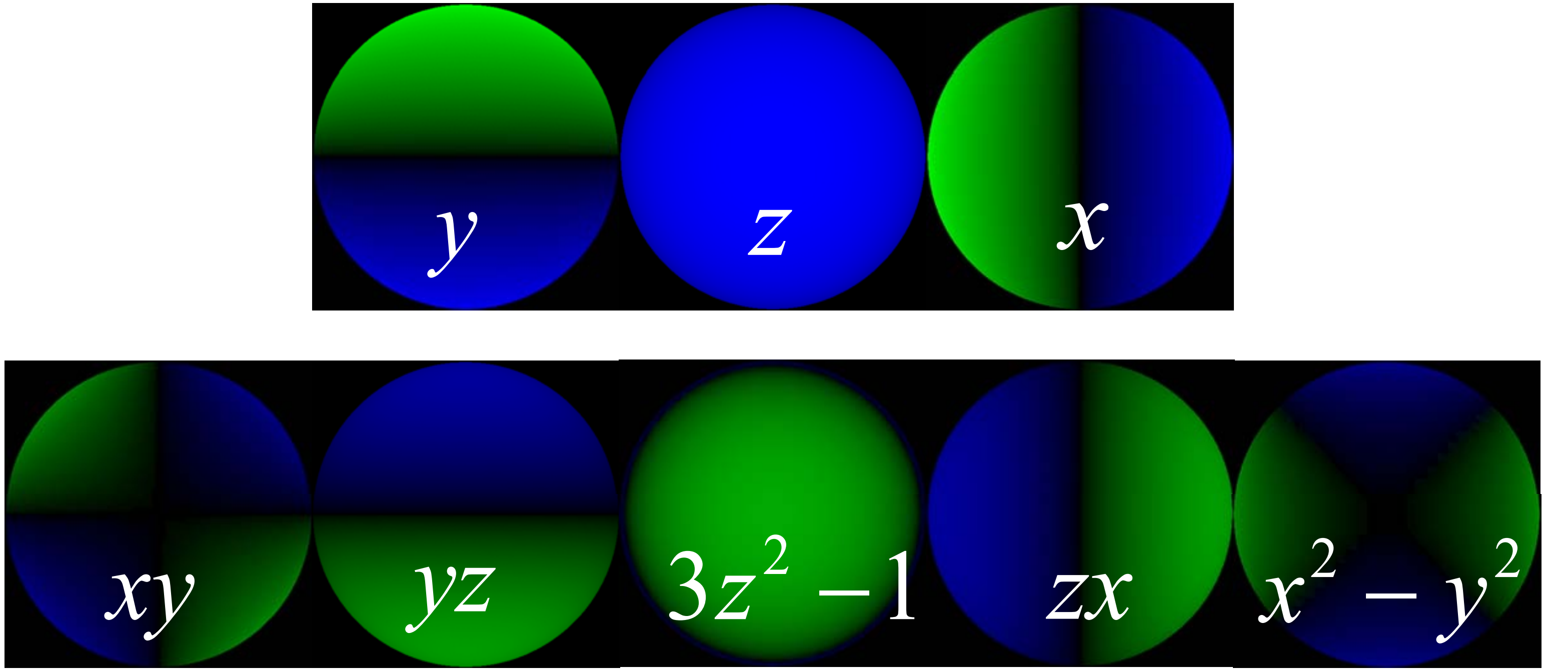
$$\begin{aligned}
(x, y, z) &= (\sin\theta\cos\phi, \sin\theta\sin\phi, \cos\theta) \\
Y_{00}(\theta, \phi) &= 0.282095 \\
(Y_{11}; Y_{10}; Y_{1-1})\,(\theta, \phi) &= 0.488603\,(x; z; y) \\
(Y_{21}; Y_{2-1}; Y_{2-2})\,(\theta, \phi) &= 1.092548\,(xz; yz; xy) \\
Y_{20}(\theta, \phi) &= 0.315392\left(3z^2 - 1\right) \\
Y_{22}(\theta, \phi) &= 0.546274\left(x^2 - y^2\right)
\end{aligned}$$

Spherical harmonics





$$Y_{lm}(\theta, \varphi)$$



SH projection

- **First we define a strict order for SH functions**

$$i = l(l+1) + m$$

- **Project a spherical function into a vector of SH coefficients**

$$c_i = \int_S f(s) y_i(s) ds$$

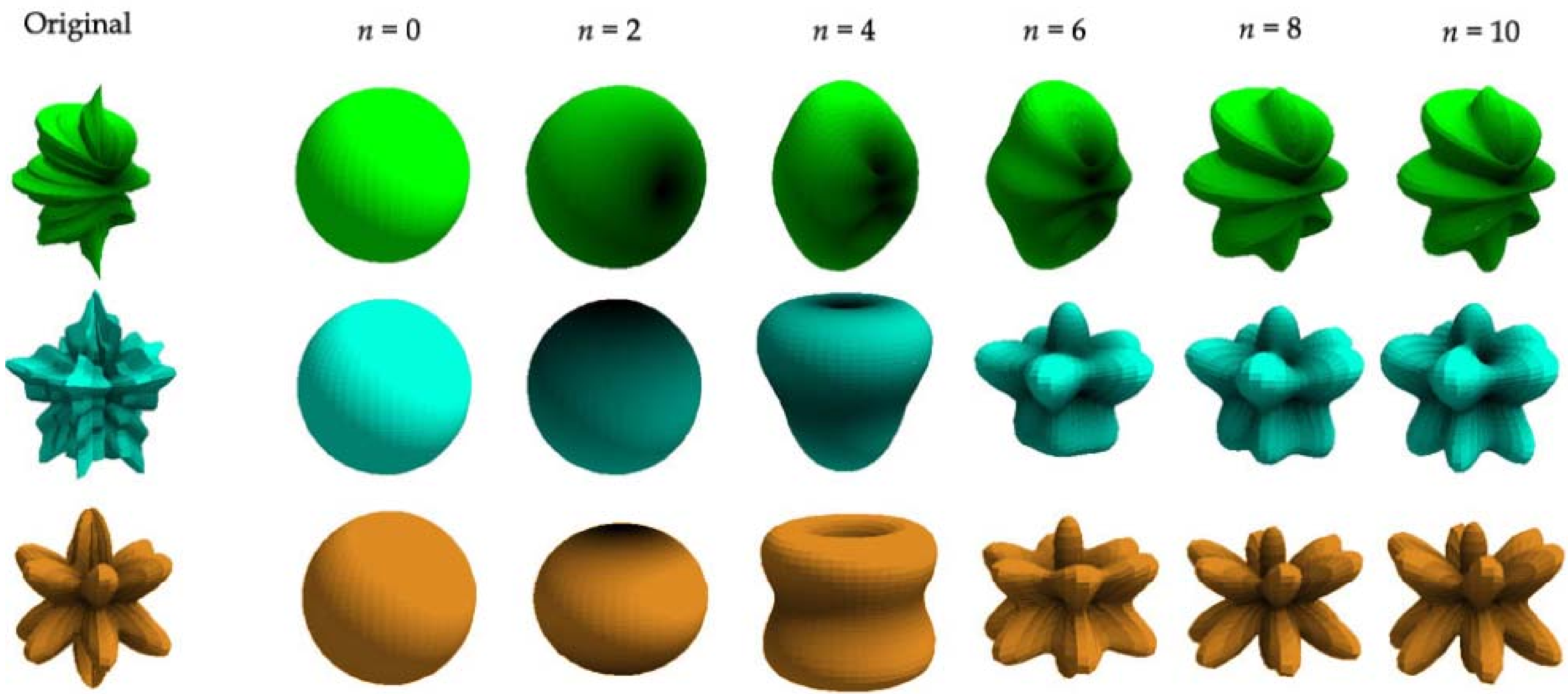
SH reconstruction

- To reconstruct the approximation to a function

$$\tilde{f}(s) = \sum_{i=0}^{N^2} c_i y_i(s)$$

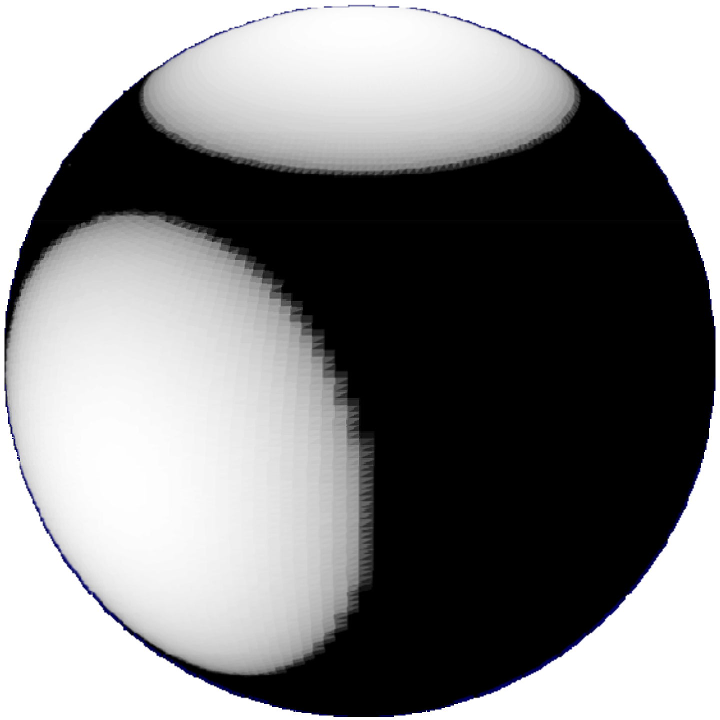
- We truncate the infinite series of SH functions to give a **low frequency** approximation

Examples of reconstruction



An example

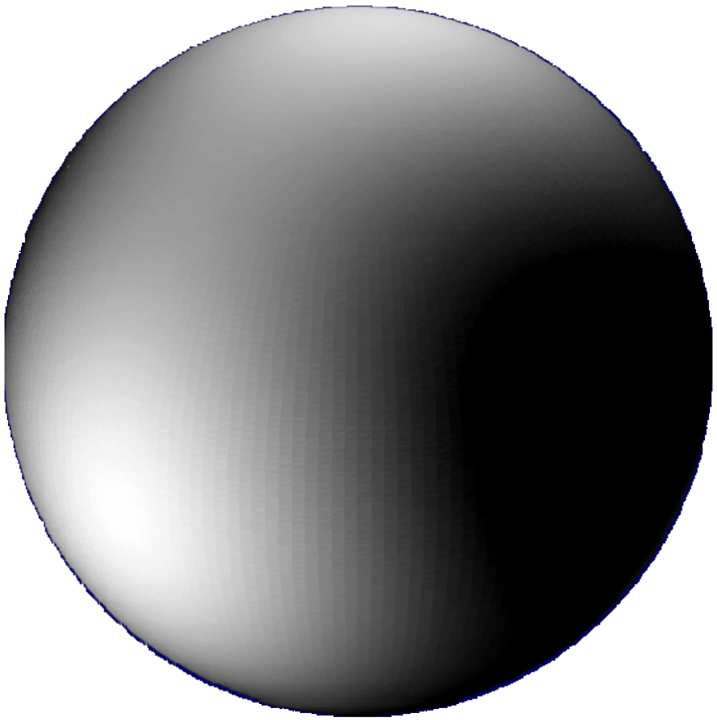
- Take a function comprised of two area light sources
  - SH project them into 4 bands = 16 coefficients

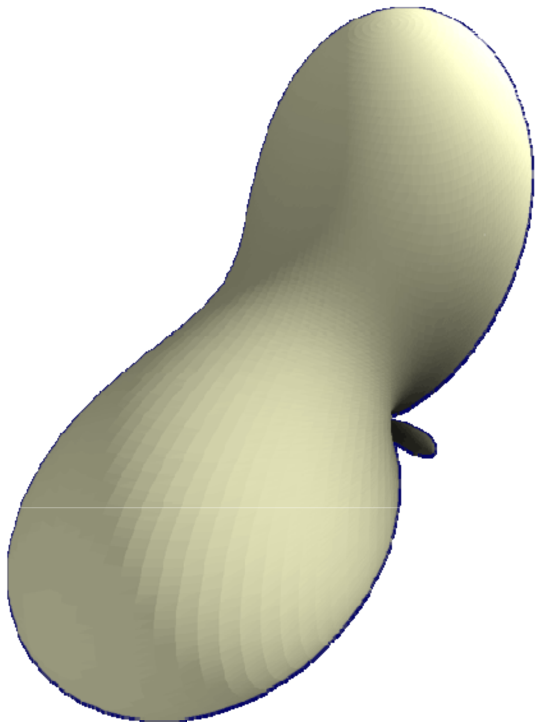


$$\begin{bmatrix} 1.329, \\ -0.679, 0.930, 0.908, \\ -0.940, 0, 0.417, 0, 0.278, \\ -0.642, 0.001, 0.317, 0.837, \\ -0.425, 0, -0.238 \end{bmatrix}$$

Low frequency light source

- We reconstruct the signal
  - Using only these coefficients to find a low frequency approximation to the original light source





SH lighting for diffuse objects

- *An Efficient Representation for Irradiance Environment Maps*, Ravi Ramamoorthi and Pat Hanrahan, SIGGRAPH 2001
- Assumptions
  - Diffuse surfaces
  - Distant illumination
  - No shadowing, interreflection

$$B(p,\omega_o) = \int_{s^2} f(\mathrm{p},\omega_\mathrm{o},\omega_\mathrm{i}) L_d(\mathrm{p},\omega_\mathrm{i}) \left|\cos\theta_\mathrm{i}\right| d\omega_\mathrm{i}$$

$$B(p,n) = \rho(p) E(\mathrm{n})$$

irradiance is a function of surface normal

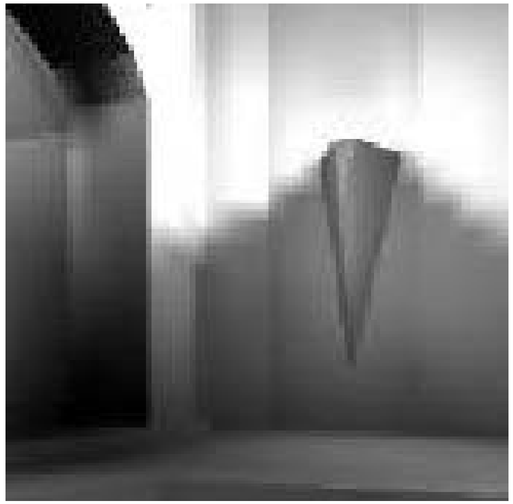
Diffuse reflection

$$B = \rho E$$

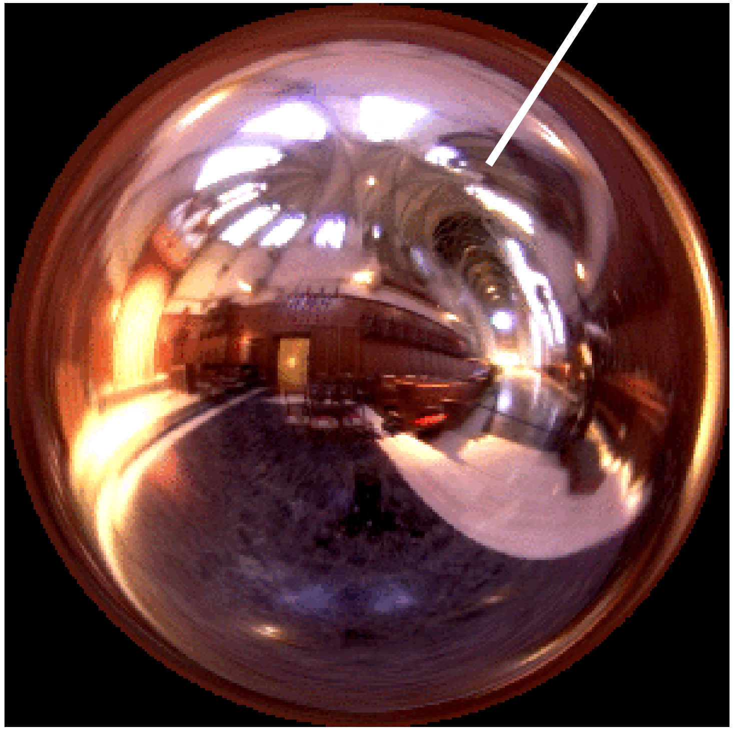
radiosity (image intensity) | reflectance (albedo/texture) | irradiance (incoming light)

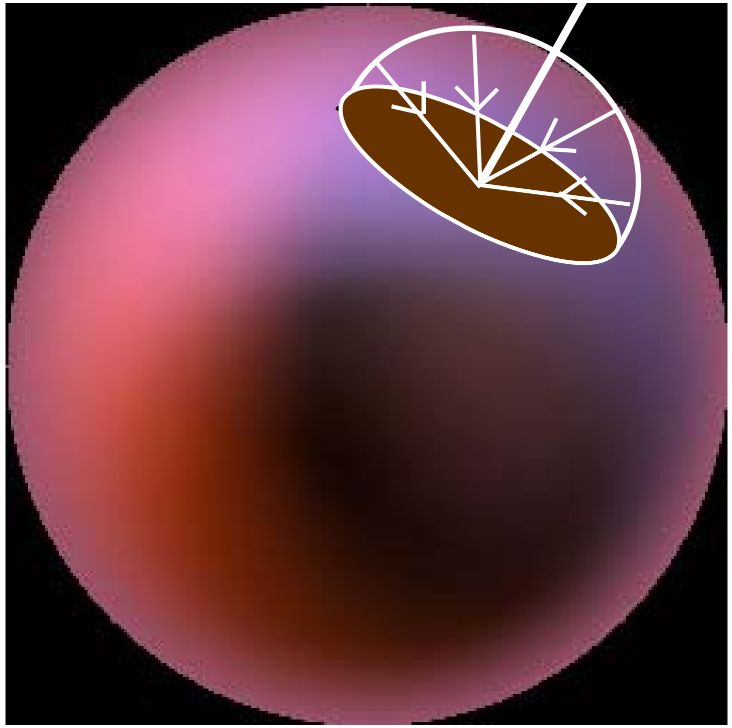
$=$

$\times$



Irradiance environment maps





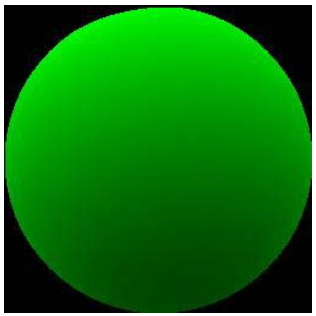
$$E(n) = \int_{\Omega} L(\omega)(n \cdot \omega)\, d\omega$$

Spherical harmonic expansion

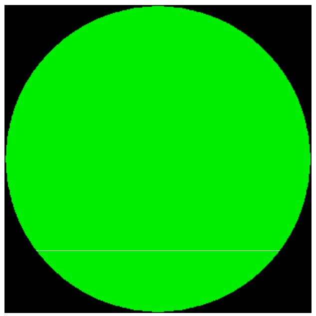
Expand lighting (L), irradiance (E) in basis functions

$$L(\theta,\phi)=\sum_{l=0}^{\infty}\sum_{m=-l}^{+l}L_{lm}Y_{lm}(\theta,\phi)$$

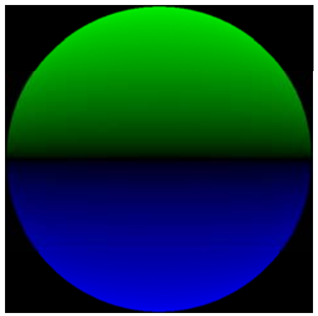
$$E(\theta,\phi)=\sum_{l=0}^{\infty}\sum_{m=-l}^{+l}E_{lm}Y_{lm}(\theta,\phi)$$



$= .67$



$+ .36$



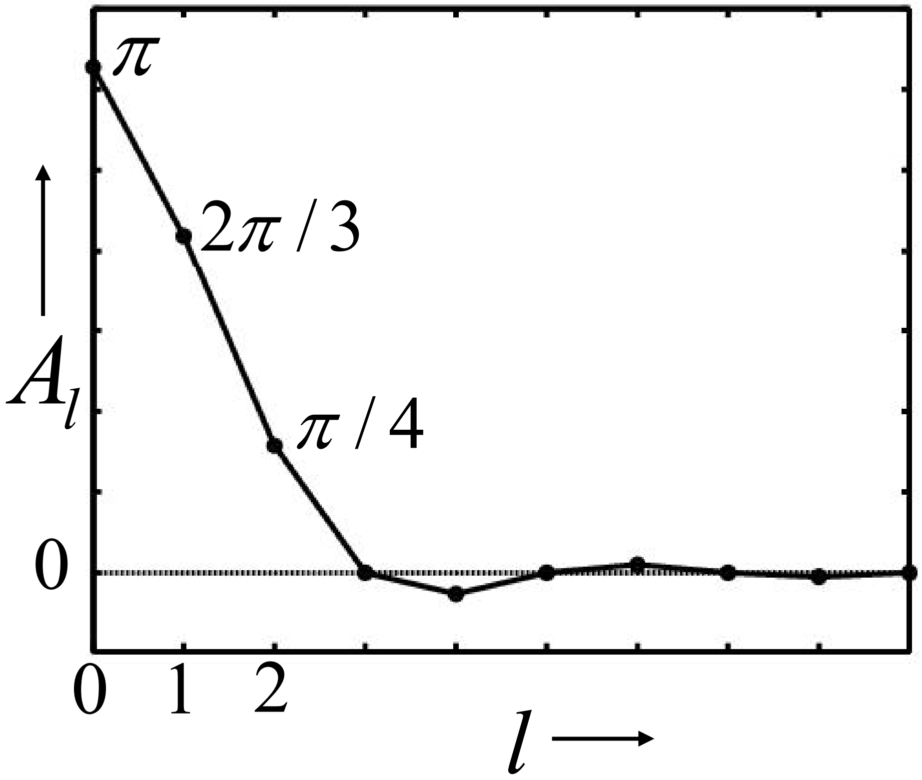
$+ \ldots$

Analytic irradiance formula

Lambertian surface acts like low-pass filter

$$E_{lm} = A_l L_{lm}$$

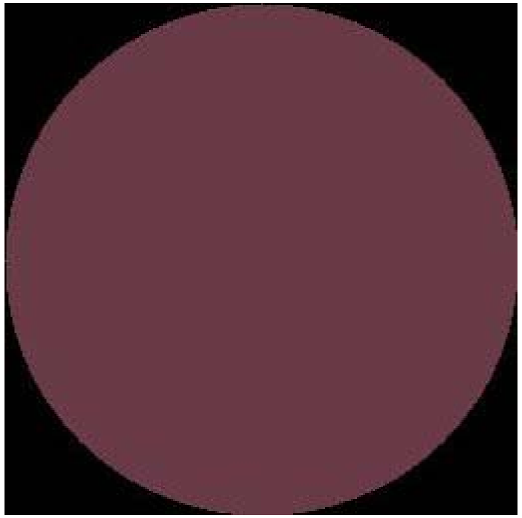
cosine term



$$A_l = 2\pi \frac{(-1)^{\frac{l}{2}-1}}{(l+2)(l-1)}\left[\frac{l!}{2^l\left(\frac{l}{2}!\right)^2}\right] \quad l \text{ even}$$

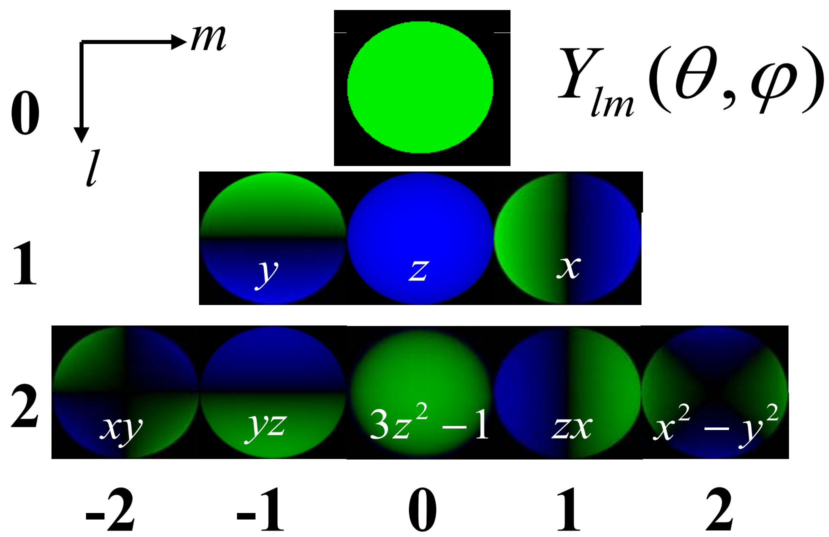
9 parameter approximation

Exact image



Order 0
1 term

RMS error = 25 %

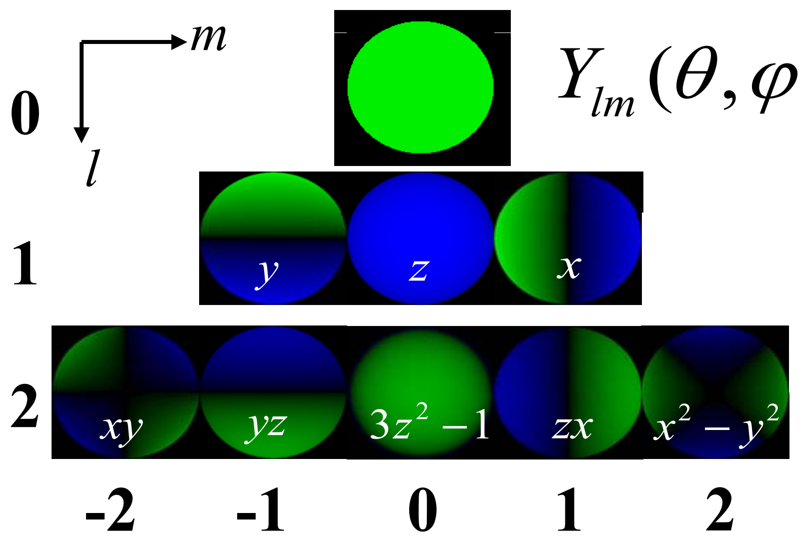


9 Parameter Approximation

Exact image

Order 1
4 terms

RMS Error = 8%



$Y_{lm}(\theta,\varphi)$

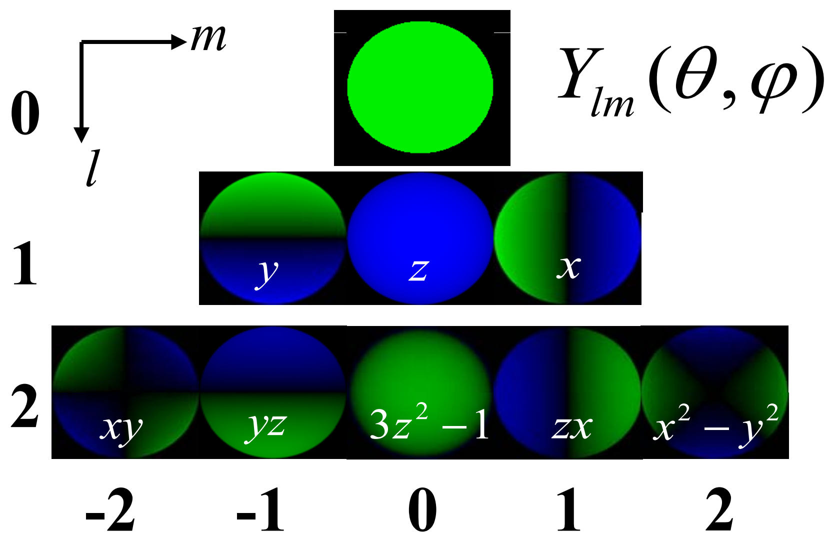
9 Parameter Approximation

Exact image

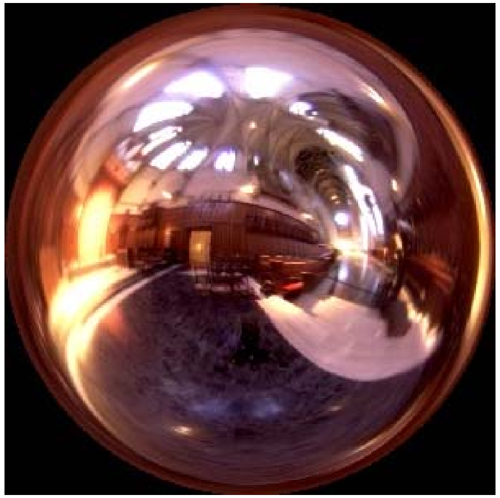
Order 2
9 terms

RMS Error = 1%

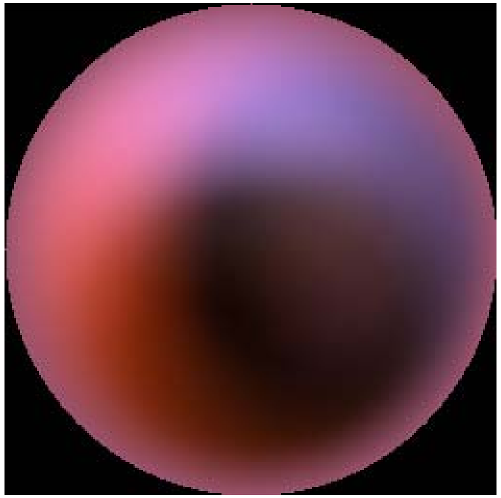
For any illumination, average error < 3% [Basri Jacobs 01]



Comparison

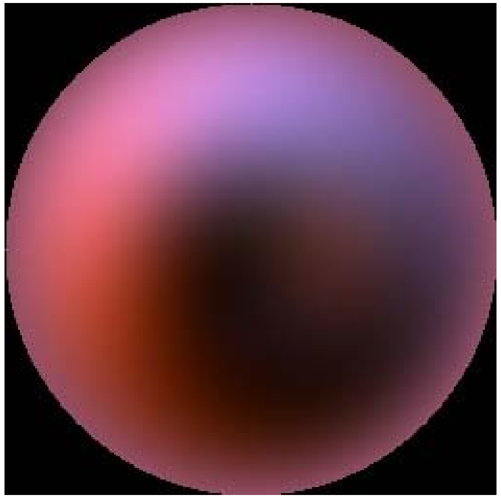


Incident
illumination
300x300



Irradiance map
Texture: 256x256
Hemispherical
Integration 2Hrs

$$\text{Time} \propto 300 \times 300 \times 256 \times 256$$

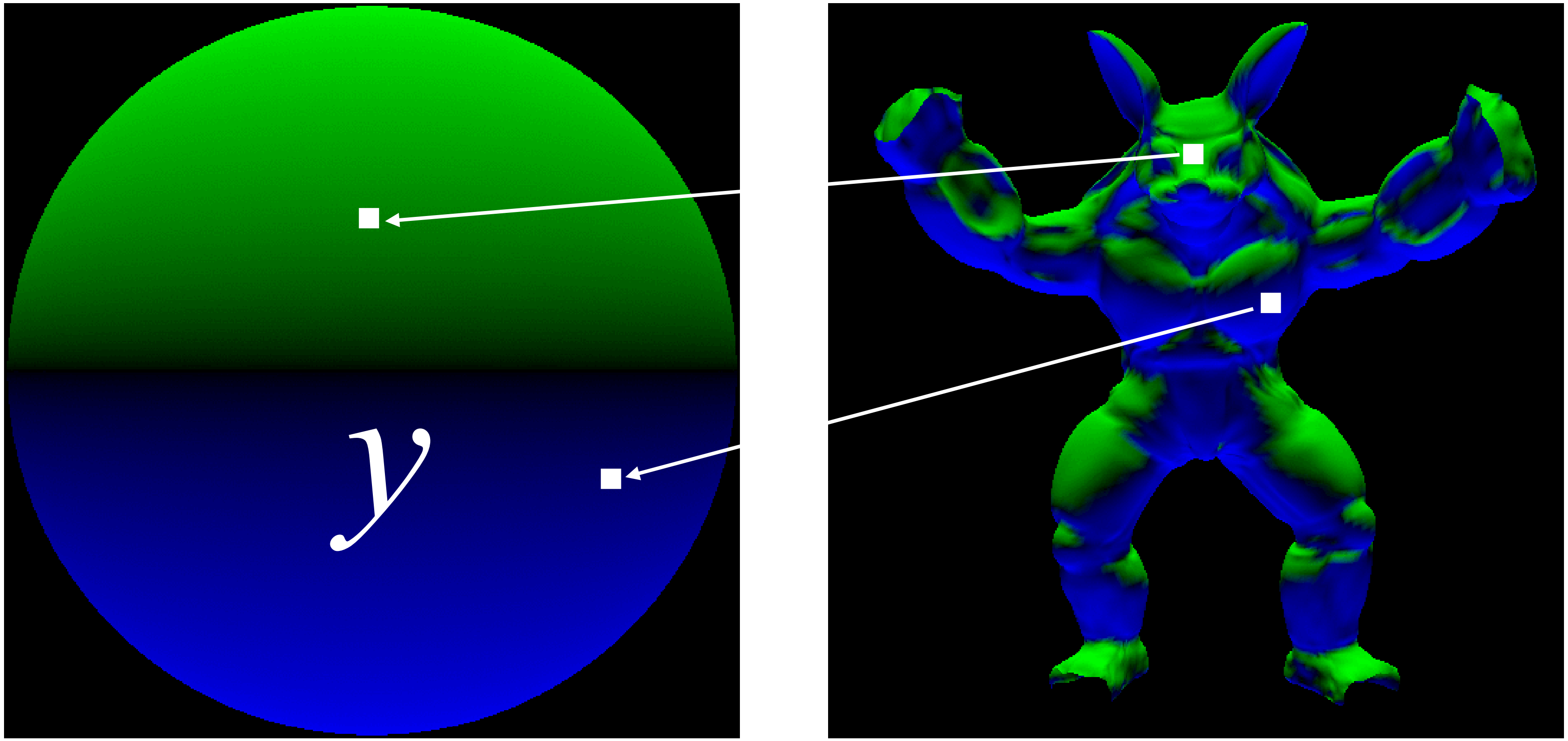


Irradiance map
Texture: 256x256
Spherical Harmonic
Coefficients 1sec

$$\text{Time} \propto 9 \times 256 \times 256$$

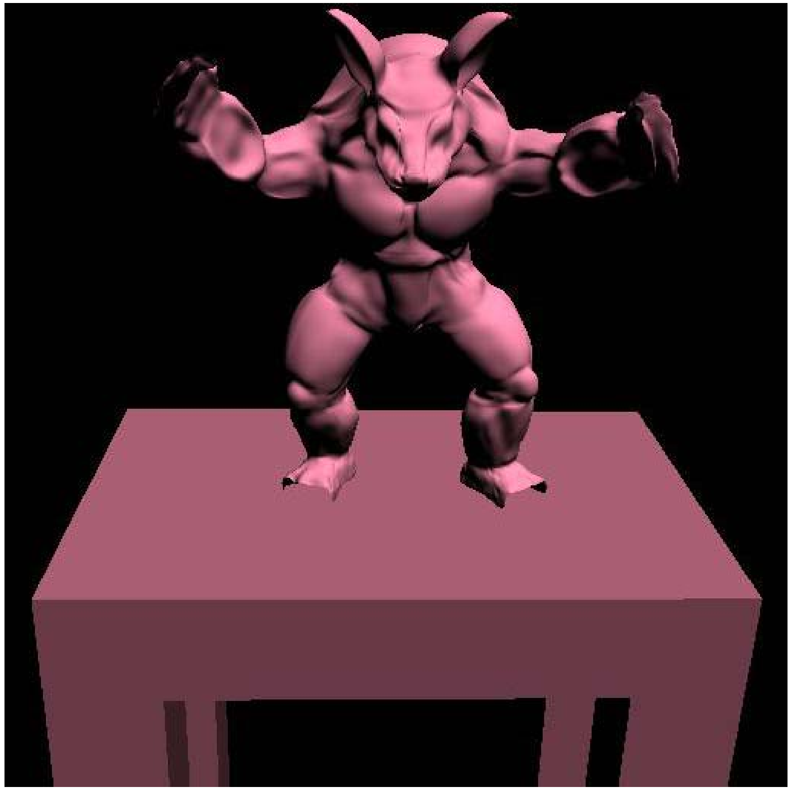
Complex geometry

Assume no shadowing: Simply use surface normal



Natural illumination

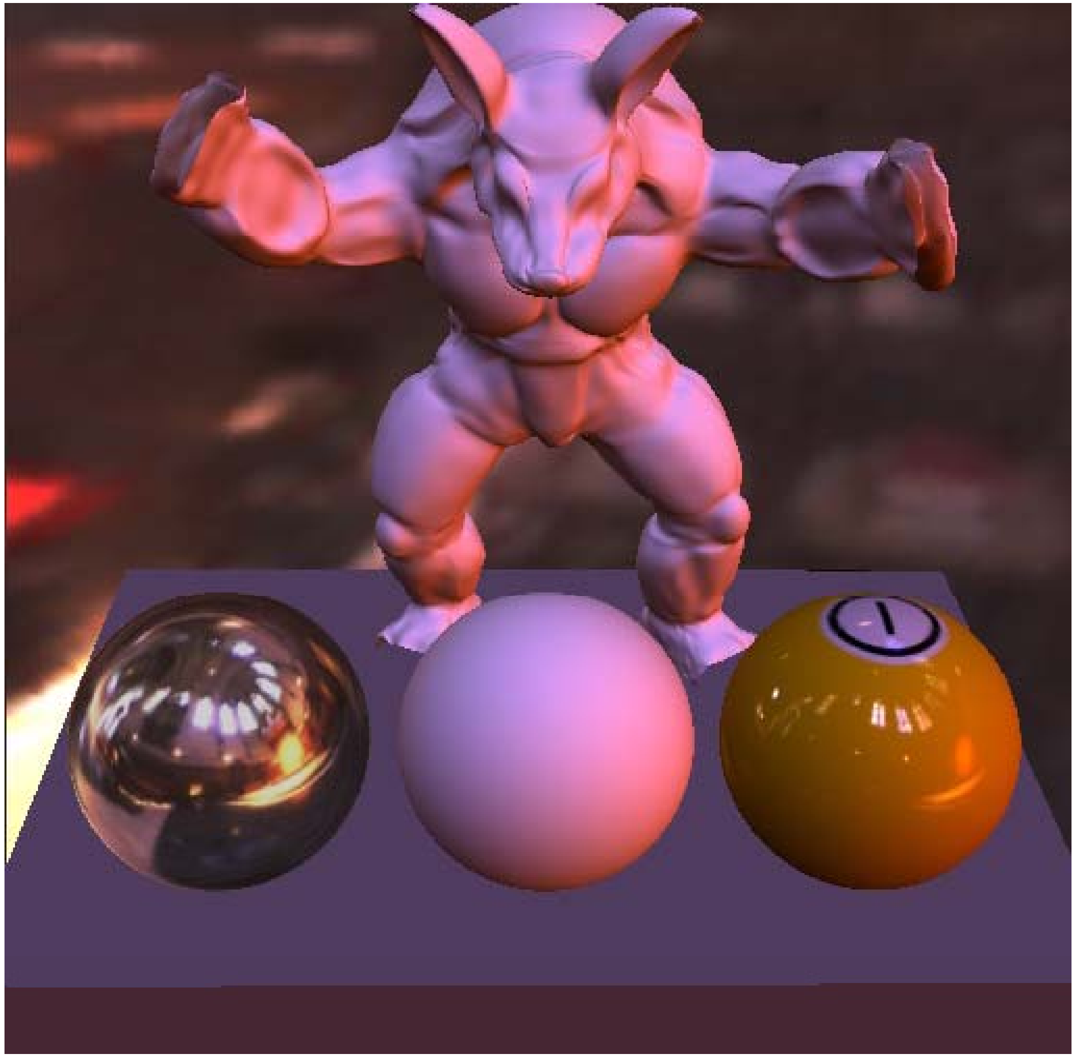
For diffuse objects, rendering with natural illumination can be done quickly



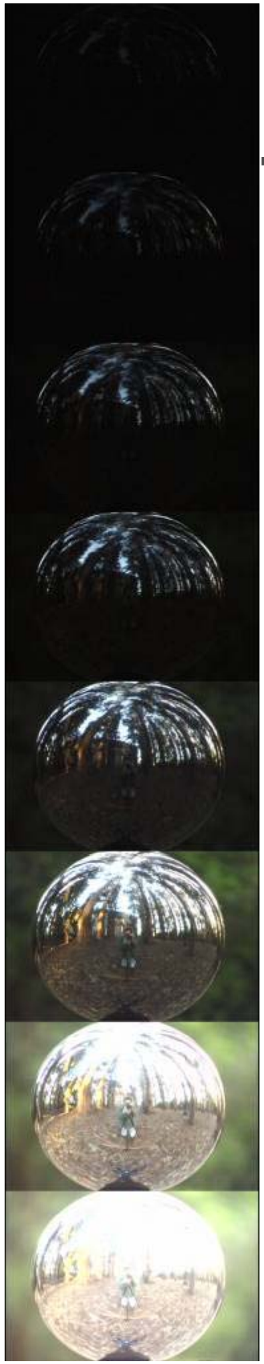
directional source

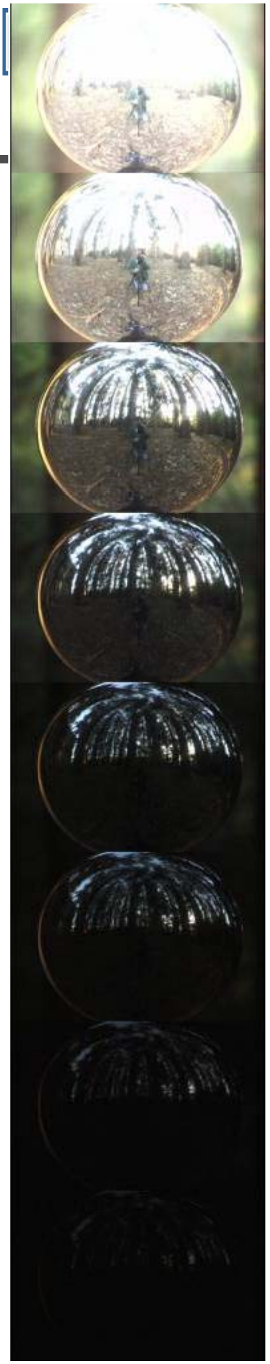
natural illumination

Video



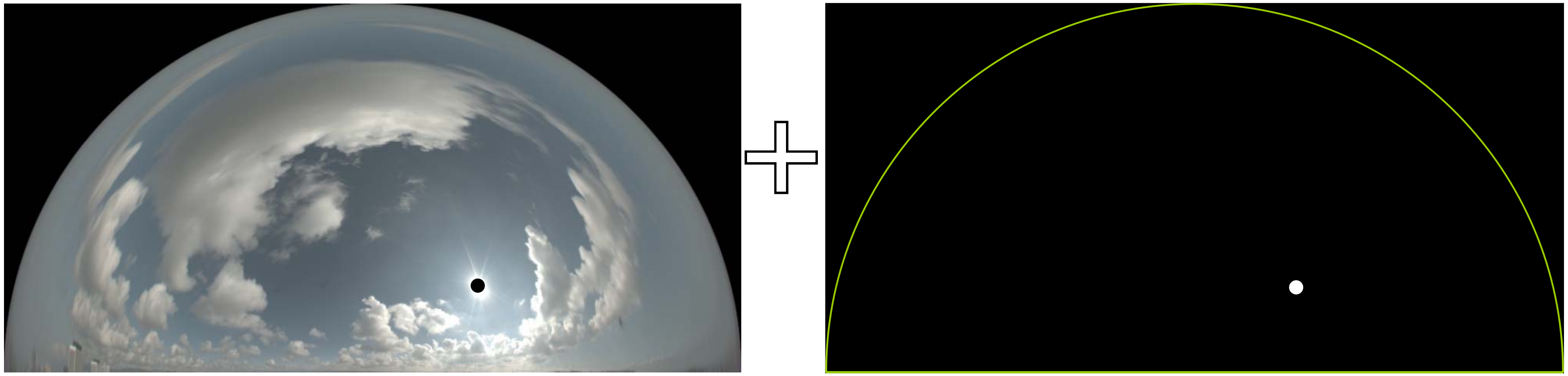
Acquiring the Light Probe



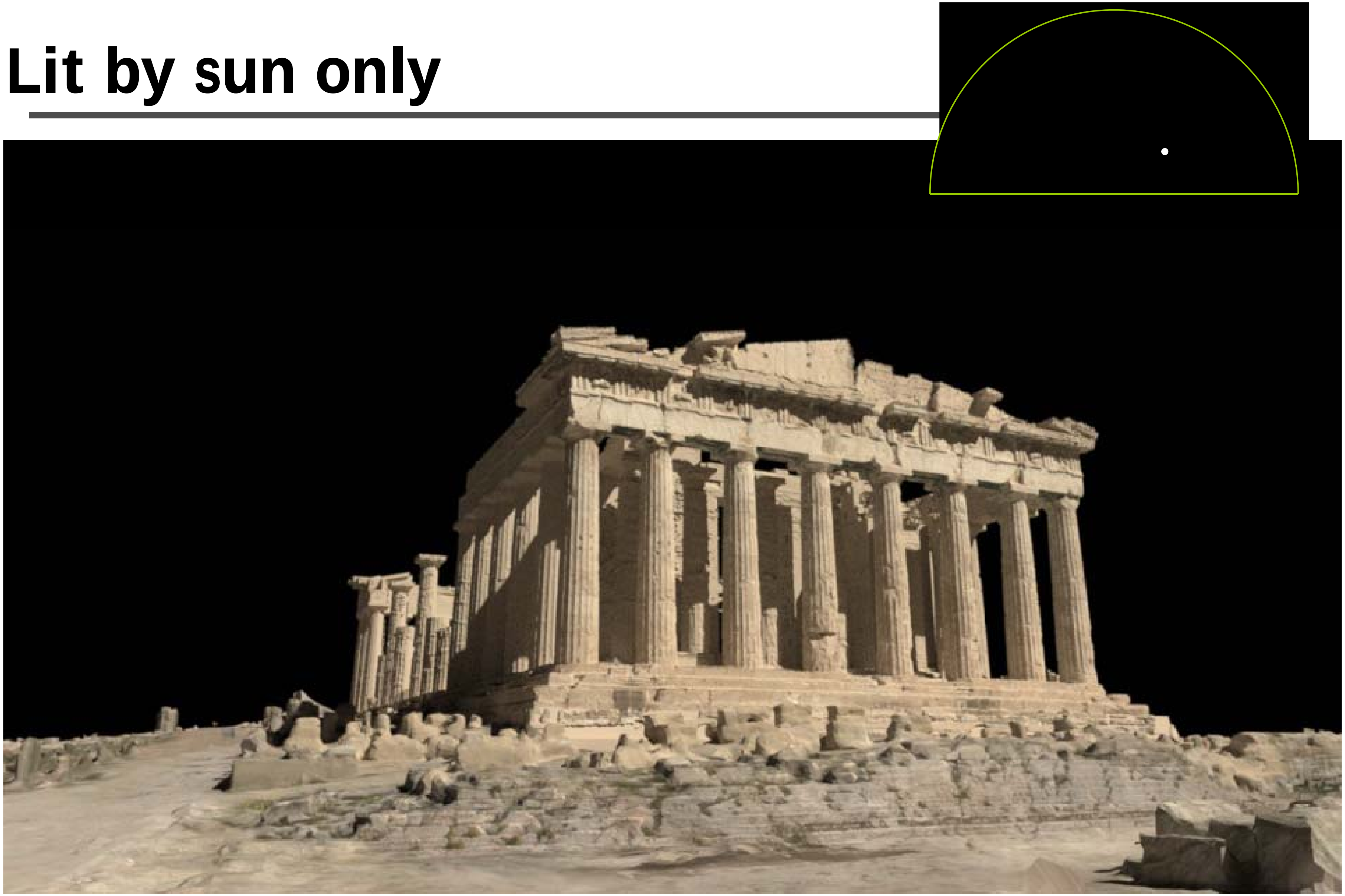


HDRI Sky Probe

Clipped Sky + Sun Source

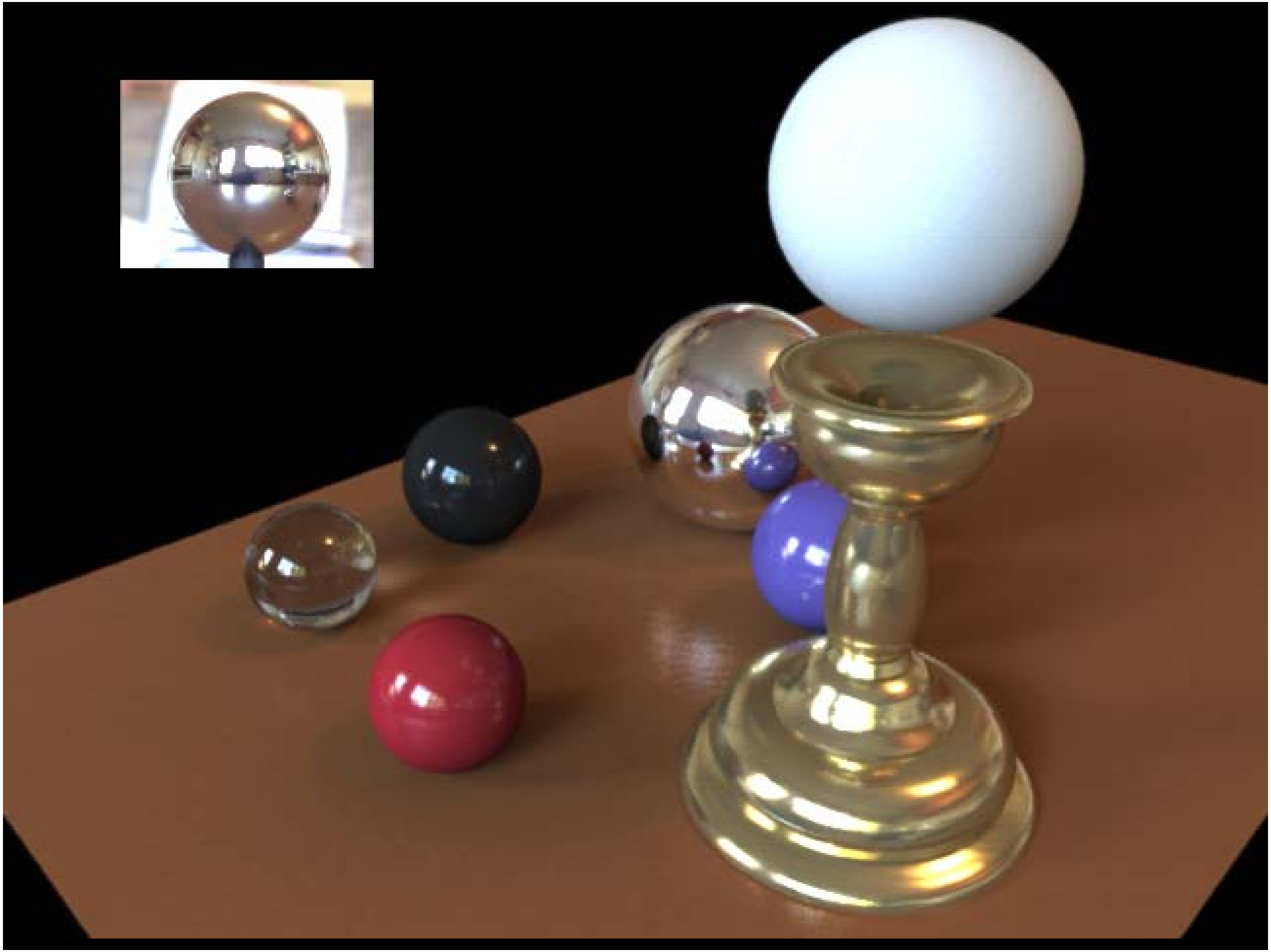


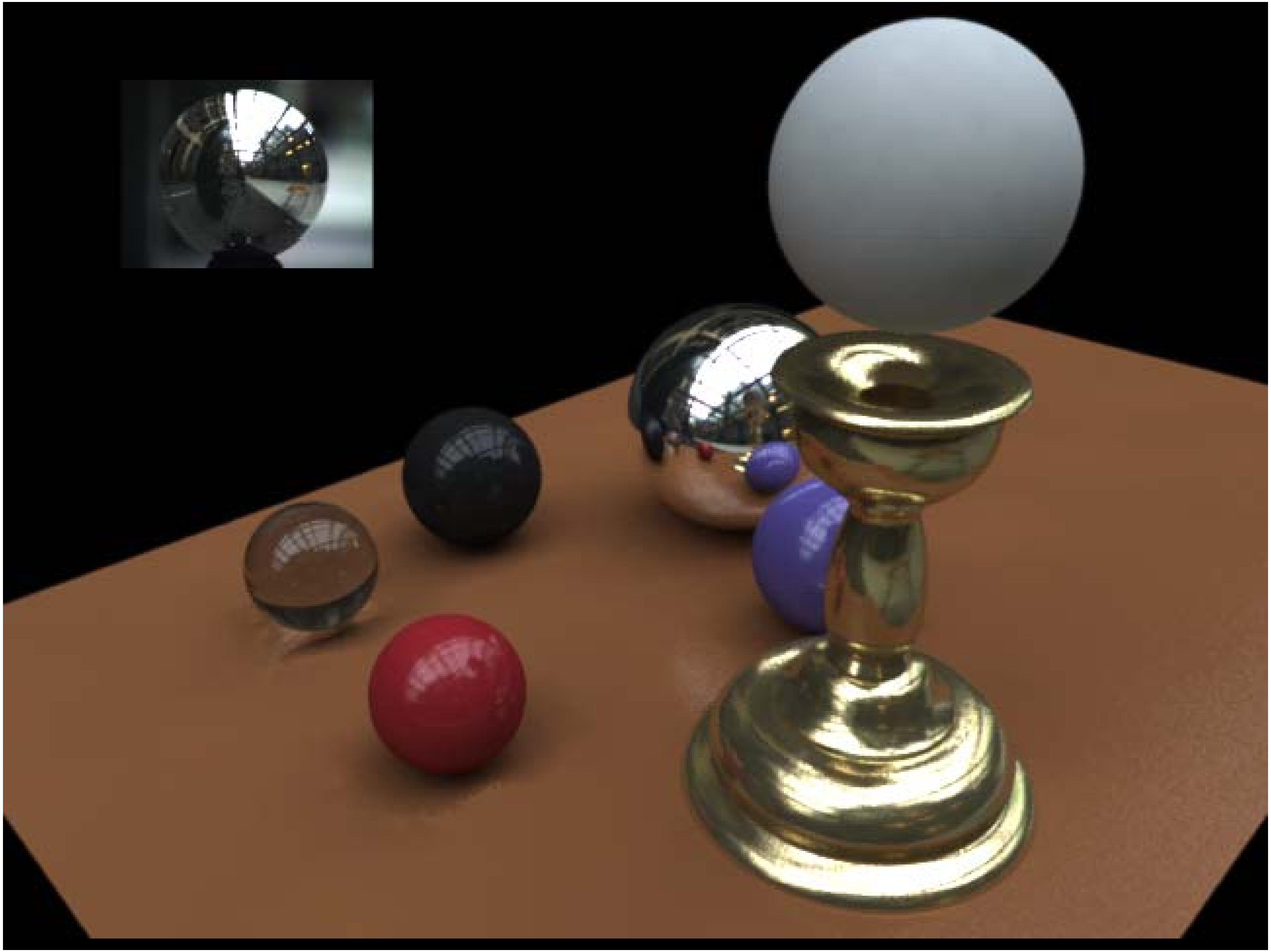
Lit by sun only



Lit by sky only

Lit by sun and sky

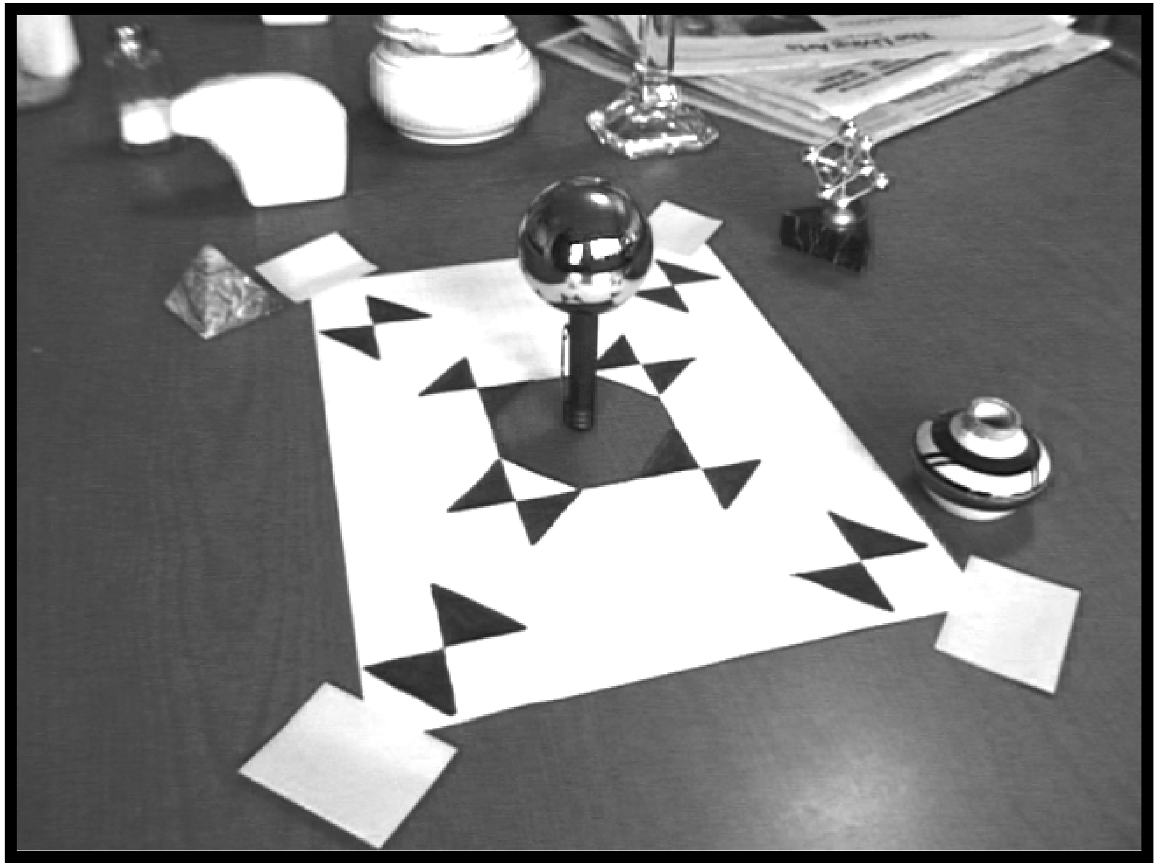




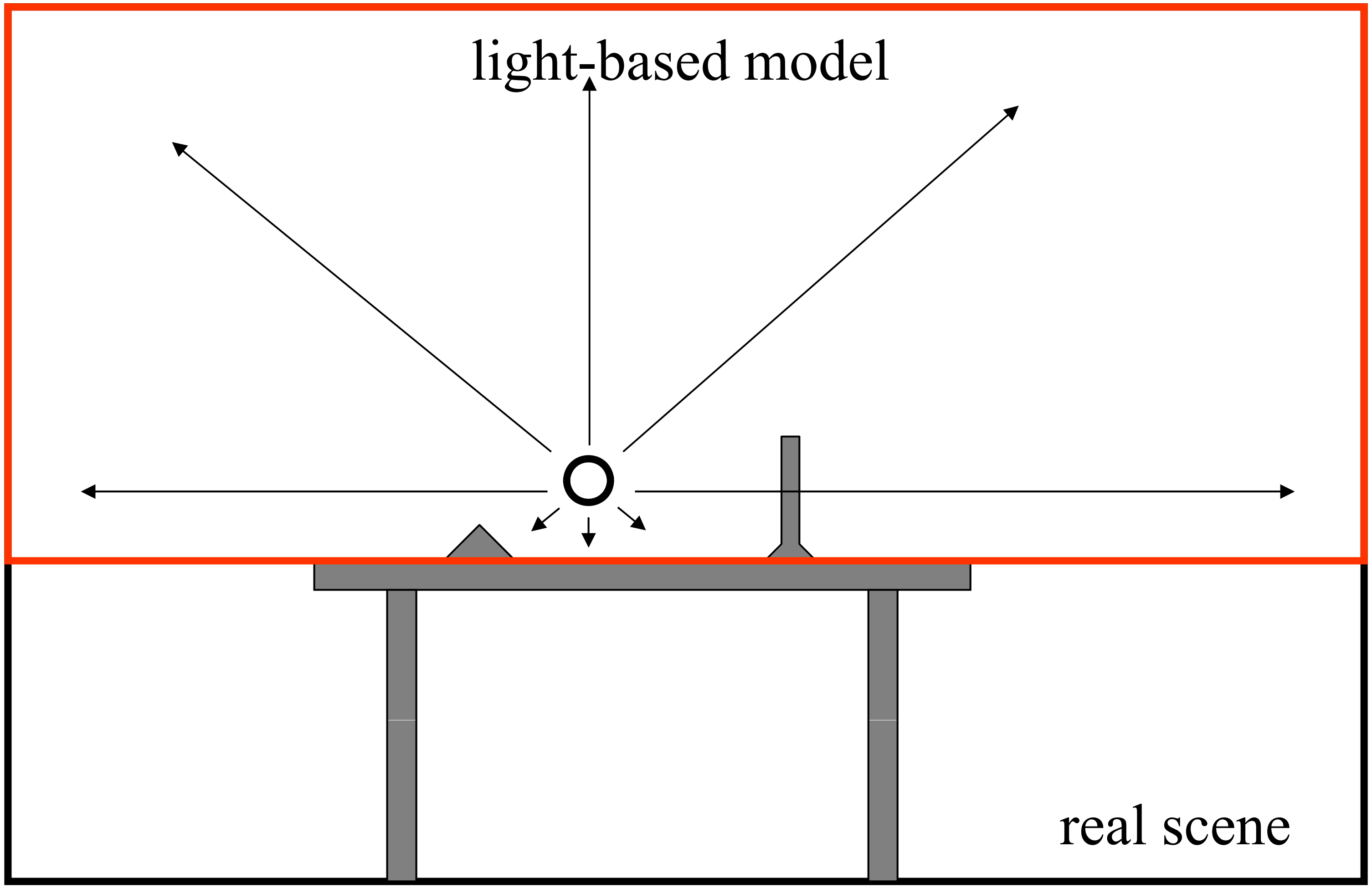
Real Scene Example

- Goal: place synthetic objects on table

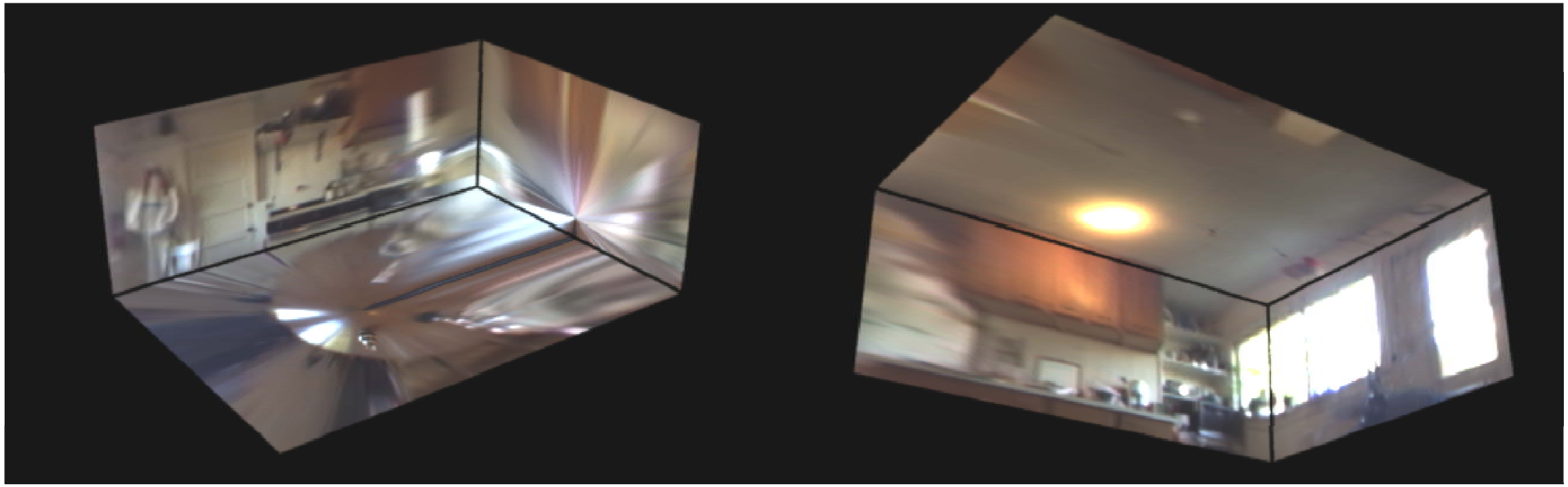
Light Probe / Calibration Grid

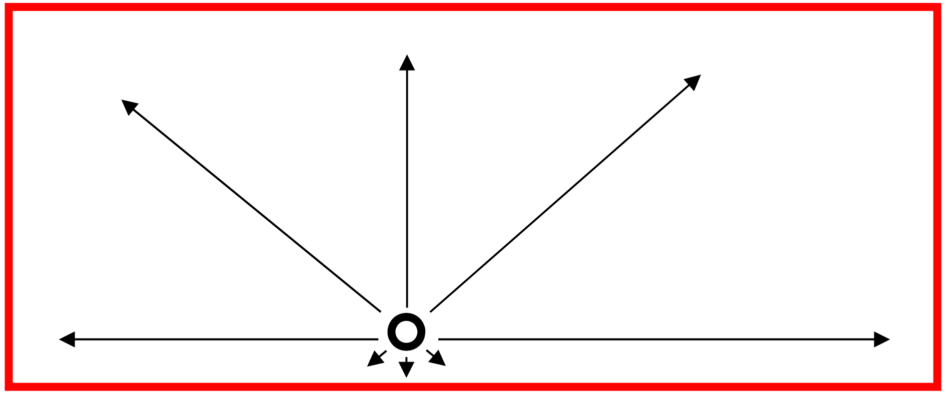


Modeling the Scene



The *Light-Based* Room Model





Rendering into the Scene

- Background Plate

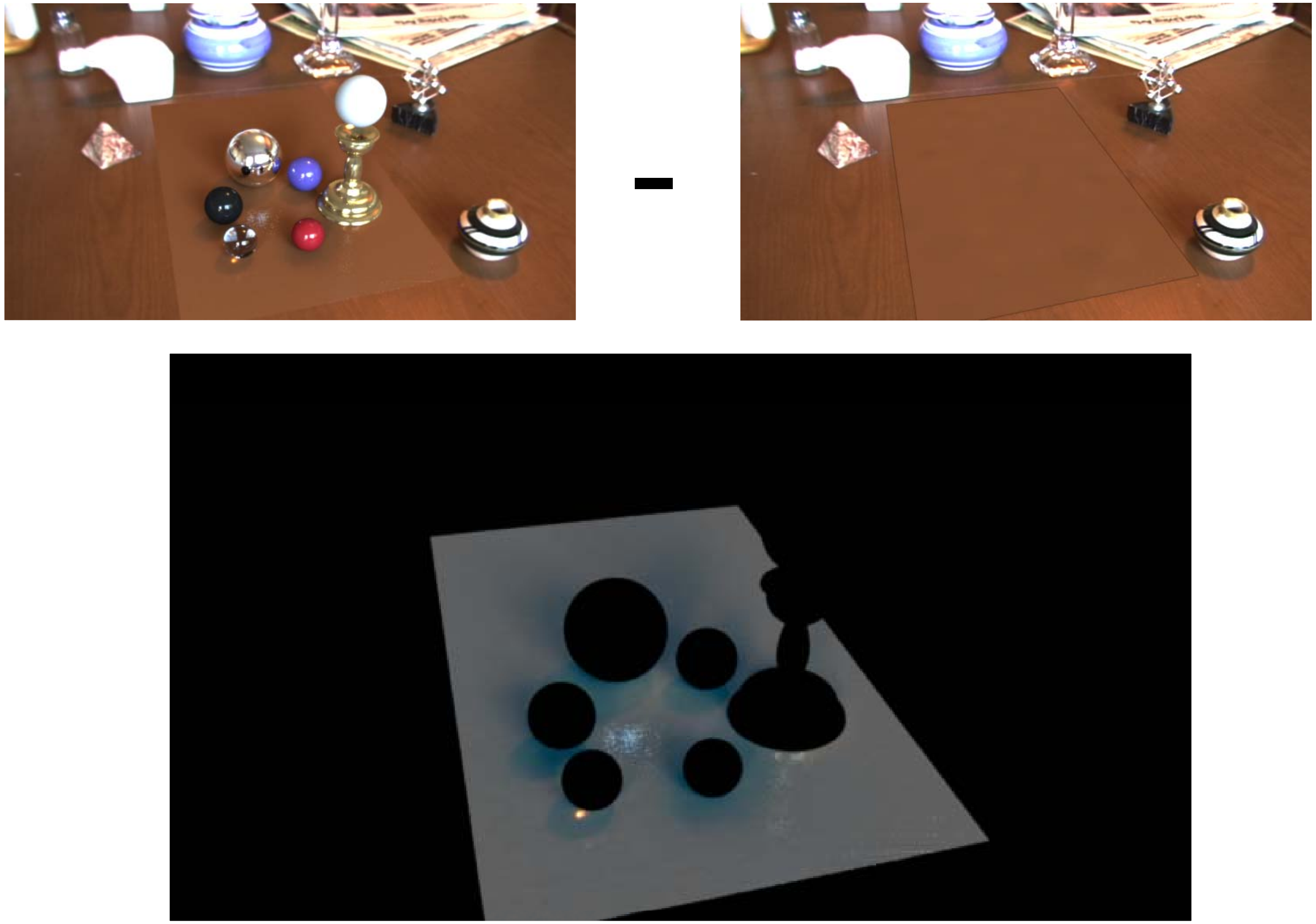
Rendering into the scene

- Objects and Local Scene matched to Scene

Differential rendering

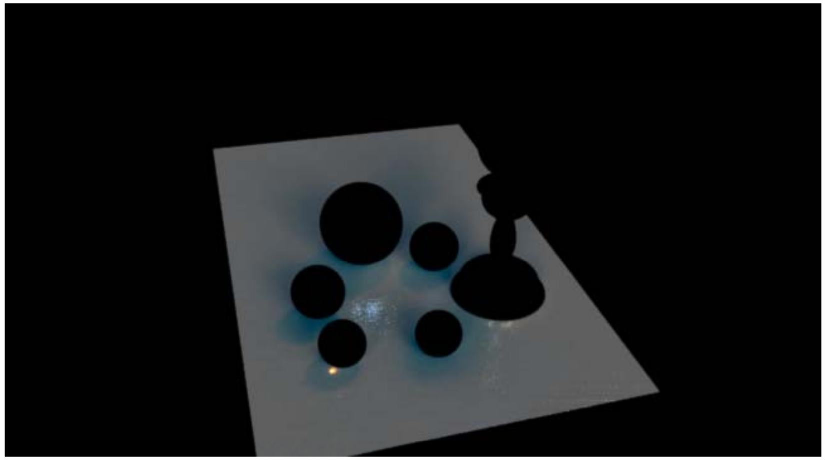
- Local scene w/o objects, illuminated by model

Differential rendering



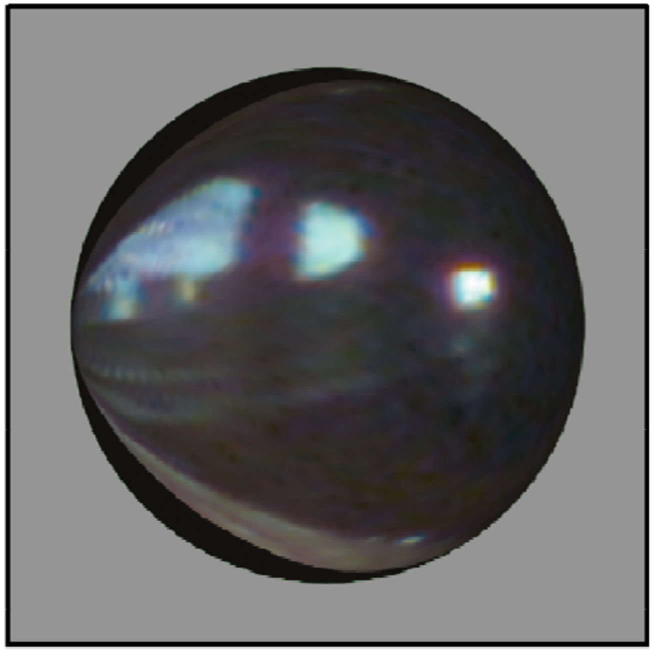
Differential rendering

+

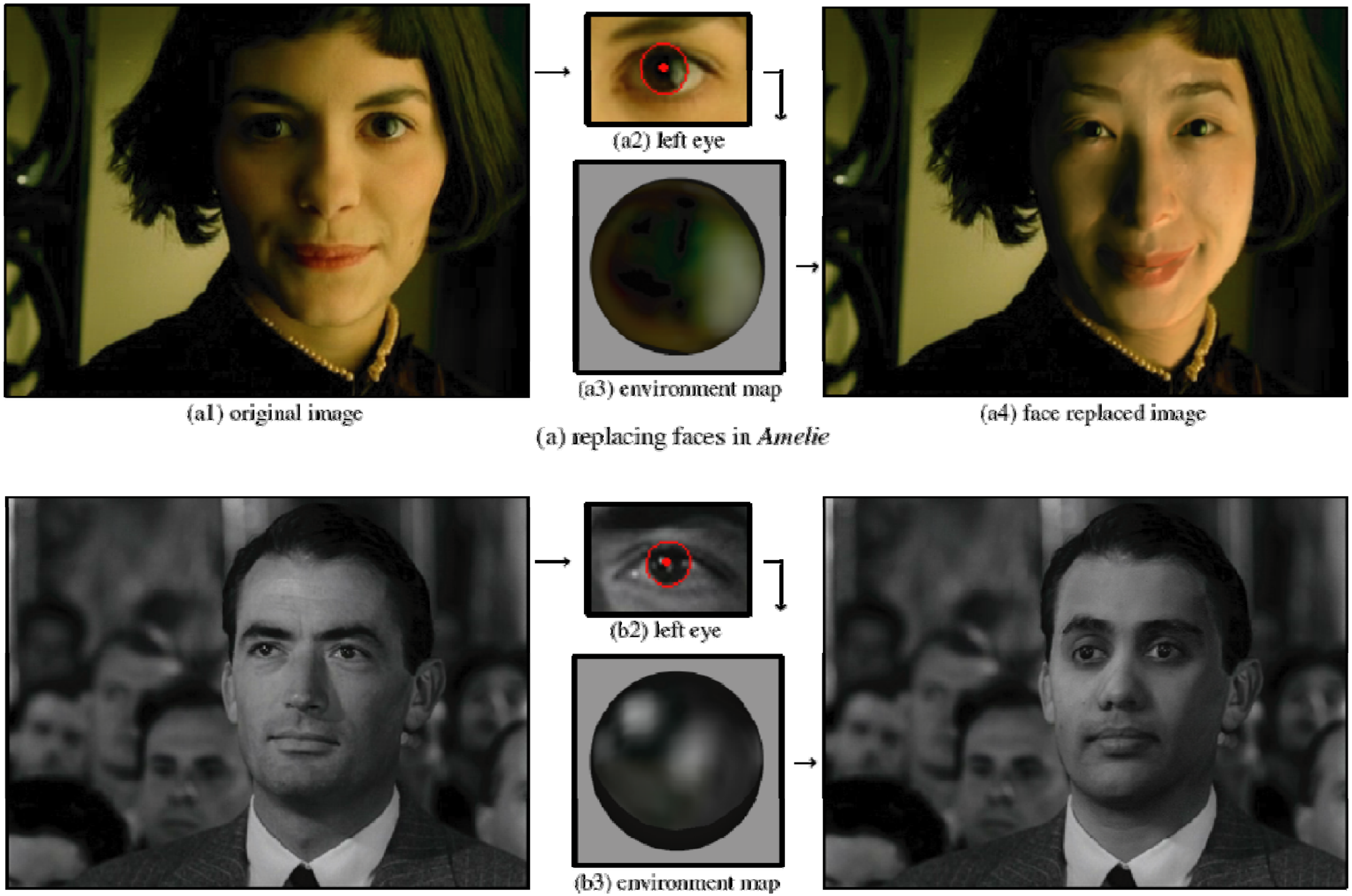


Environment map from single image?

Eye as light probe! (Nayar et al)



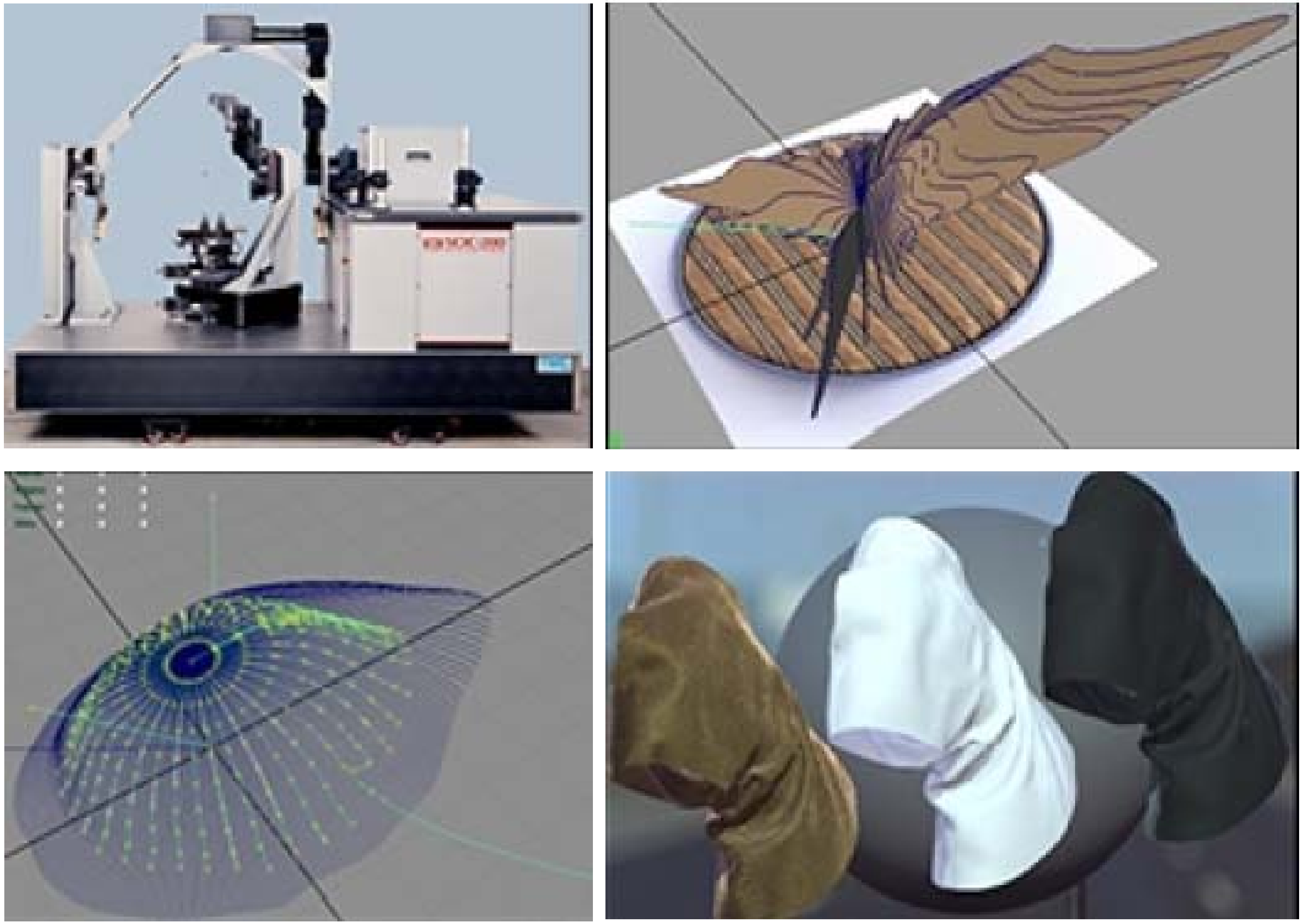
Results



(a) replacing faces in *Amelie*

Application in “Superman returns”

Capturing reflectance



Application in "The Matrix Reloaded"

3D acquisition for faces

Cyberware scanners

face & head scanner

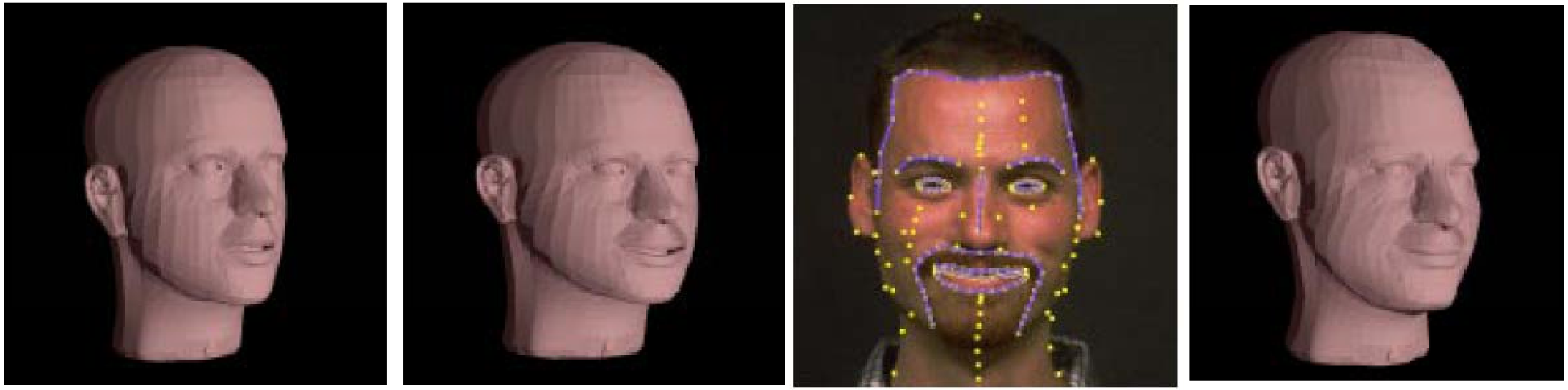
whole body scanner

Making facial expressions from photos

- Similar to Façade, use a generic face model and view-dependent texture mapping
- Procedure
  1. Take multiple photographs of a person
  2. Establish corresponding feature points
  3. Recover 3D points and camera parameters
  4. Deform the generic face model to fit points
  5. Extract textures from photos

Reconstruct a 3D model

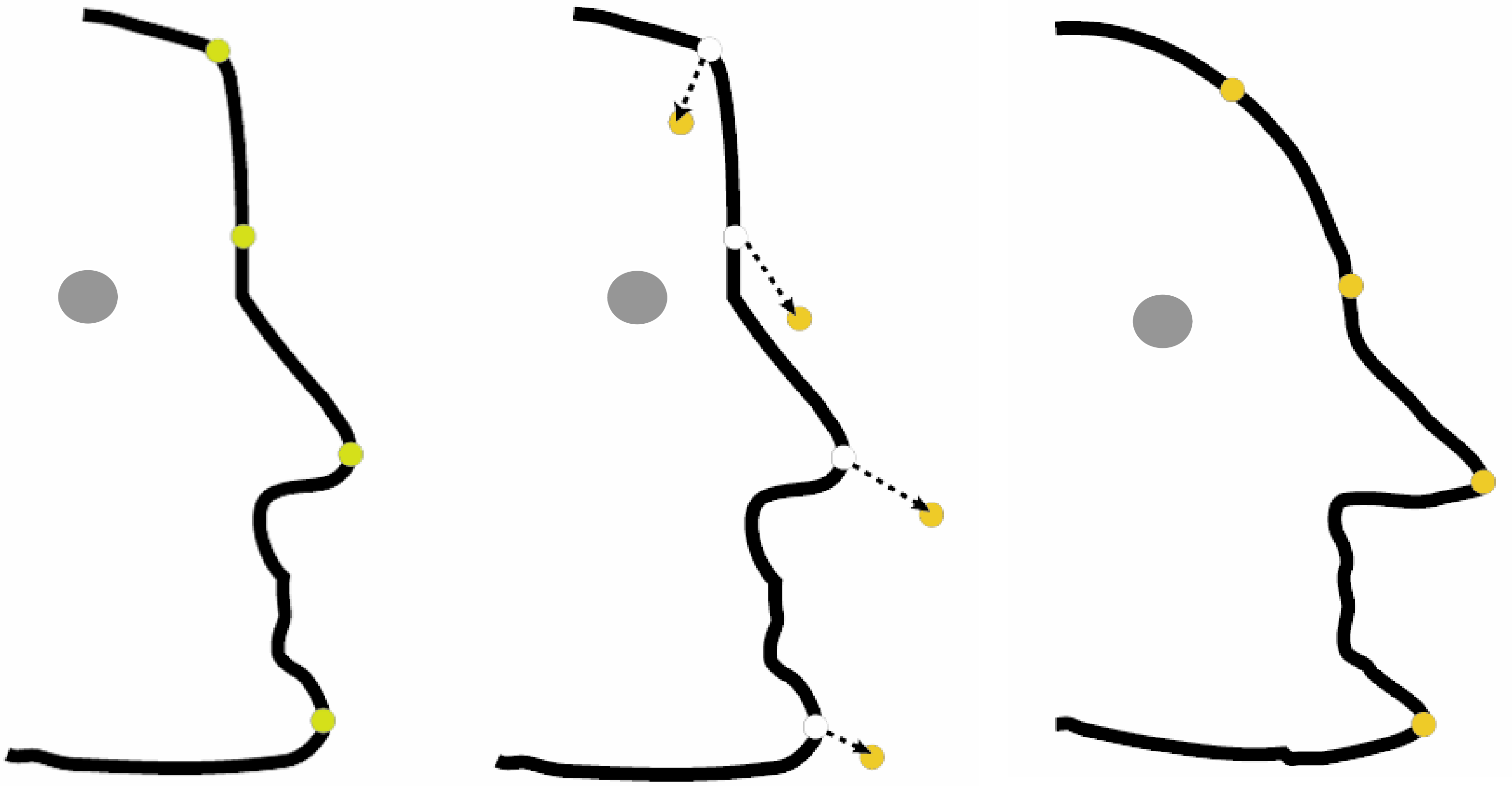
input photographs



generic 3D face model | pose estimation | more features | deformed model

Mesh deformation

- Compute displacement of feature points
- Apply scattered data interpolation

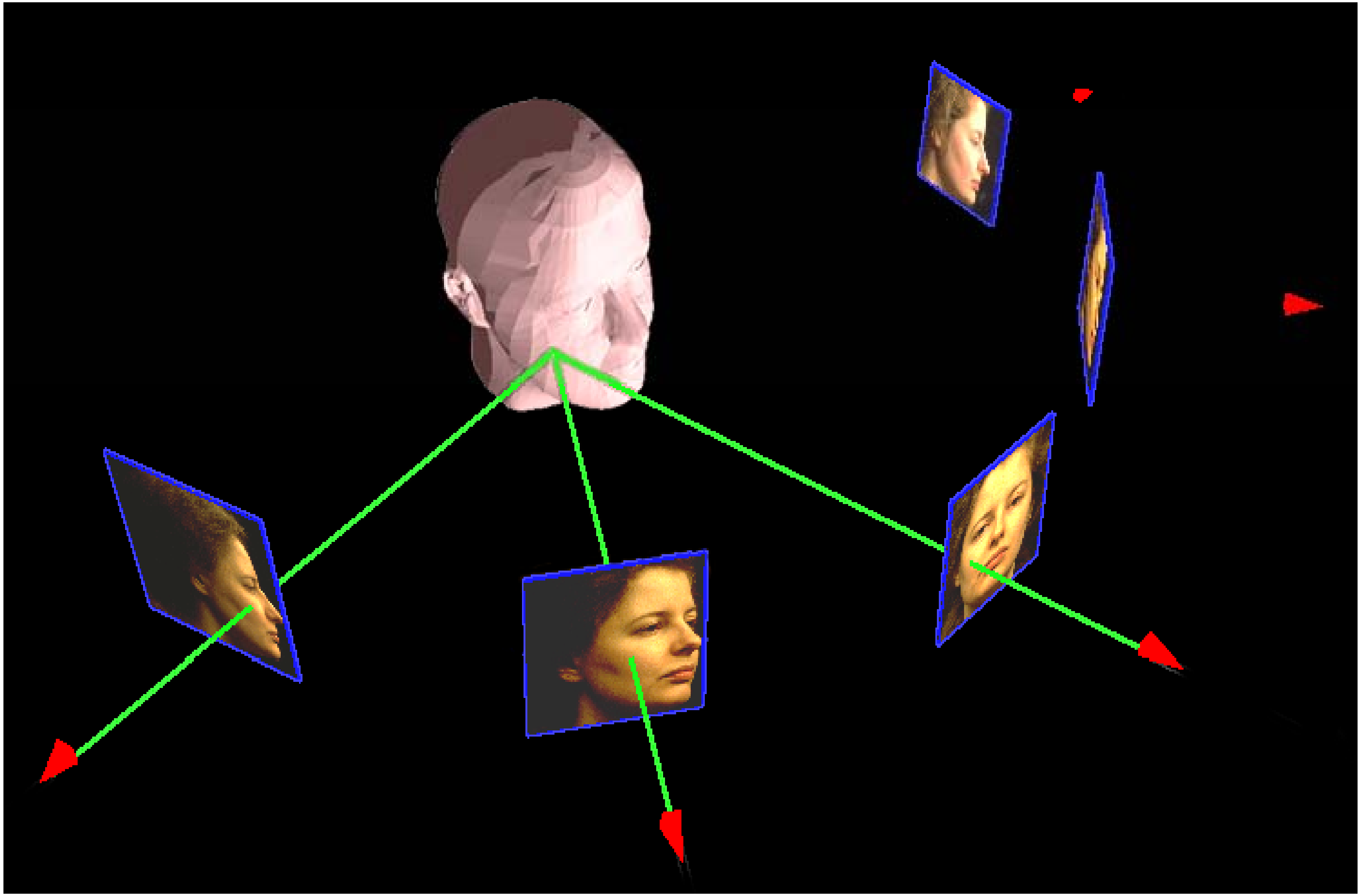


generic model displacement deformed model

Texture extraction

- The color at each point is a weighted combination of the colors in the photos
- Texture can be:
  - view-independent
  - view-dependent
- Considerations for weighting
  - occlusion
  - smoothness
  - positional certainty
  - view similarity

Texture extraction



Texture extraction

Texture extraction

view-independent

view-dependent

Model reconstruction

Use images to adapt a generic face model.

Creating new expressions

- In addition to global blending we can use:
  - Regional blending
  - Painterly interface

Creating new expressions

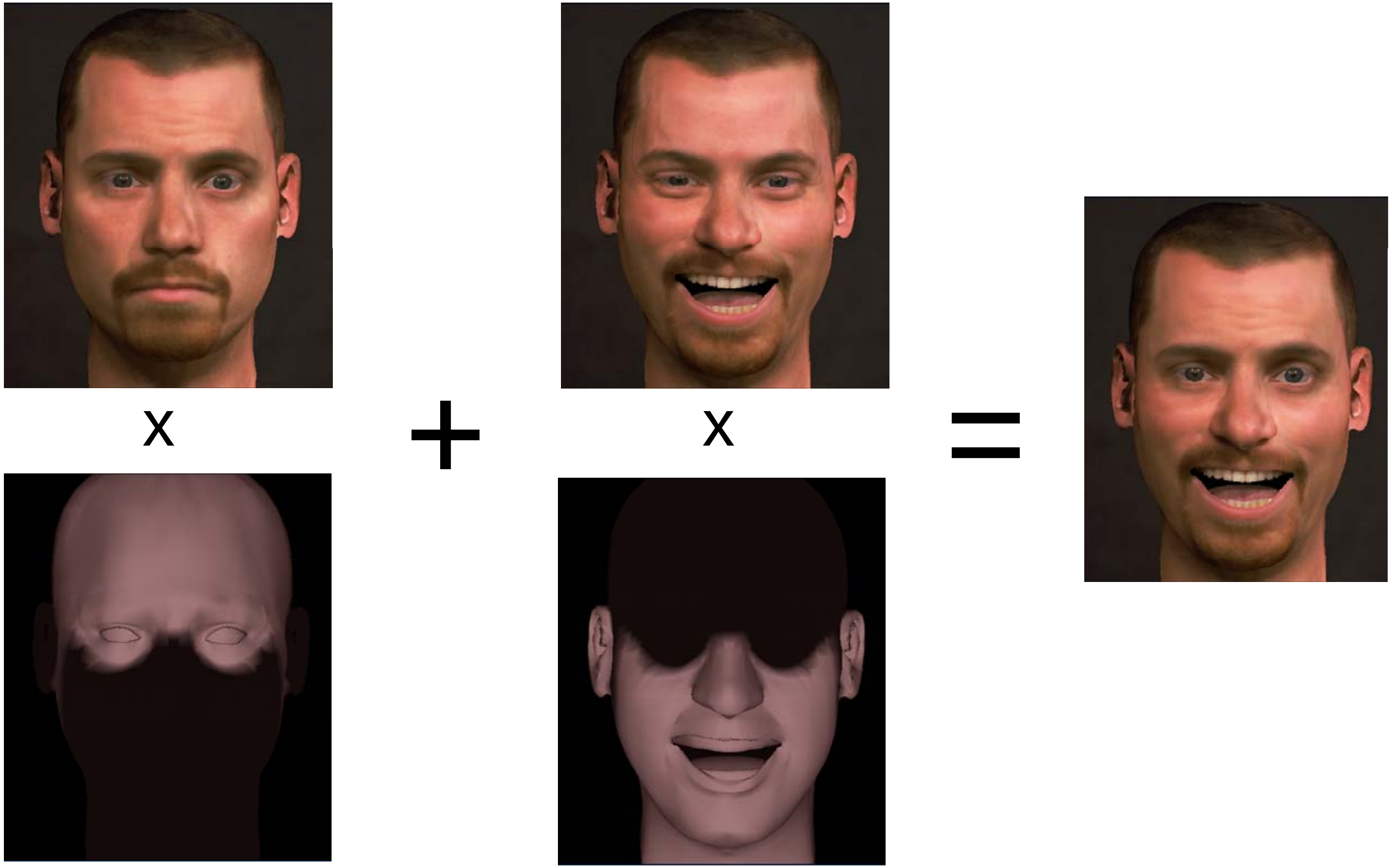
New expressions are created with 3D morphing:

/2 +

/2 =

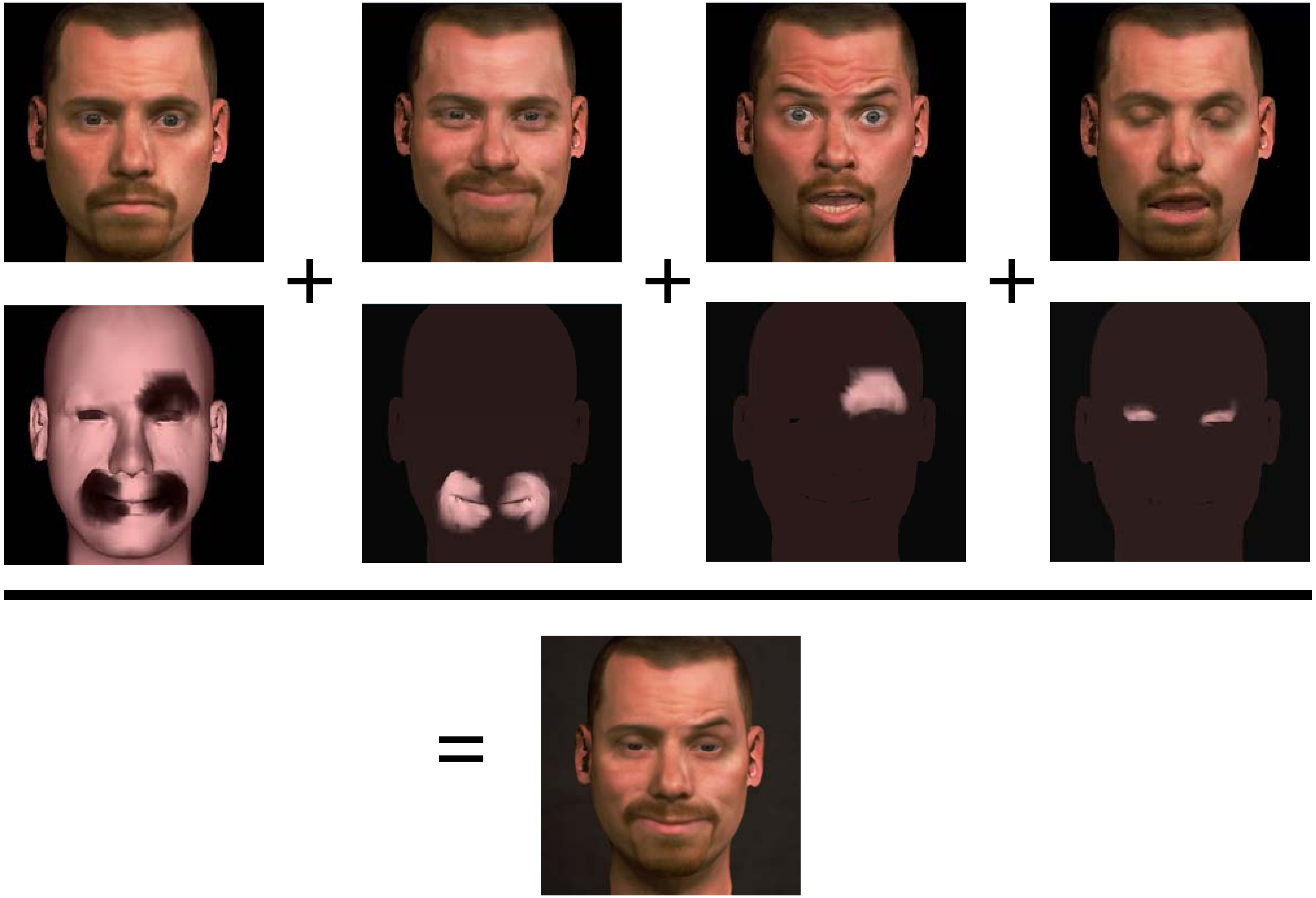
Applying a global blend

Creating new expressions



Applying a region-based blend

Creating new expressions



Using a painterly interface

Drunken smile

Animating between expressions

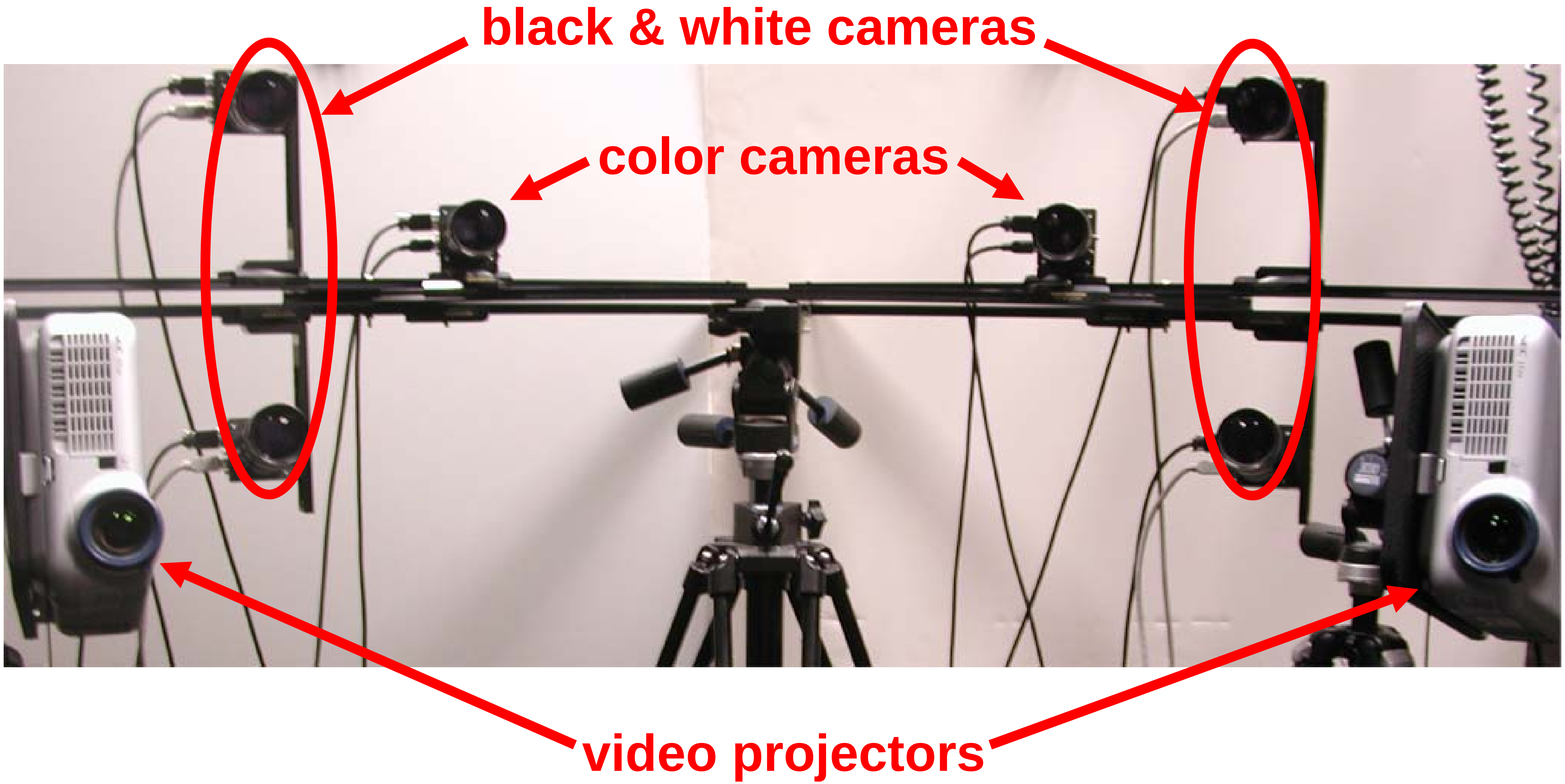
Morphing over time creates animation:

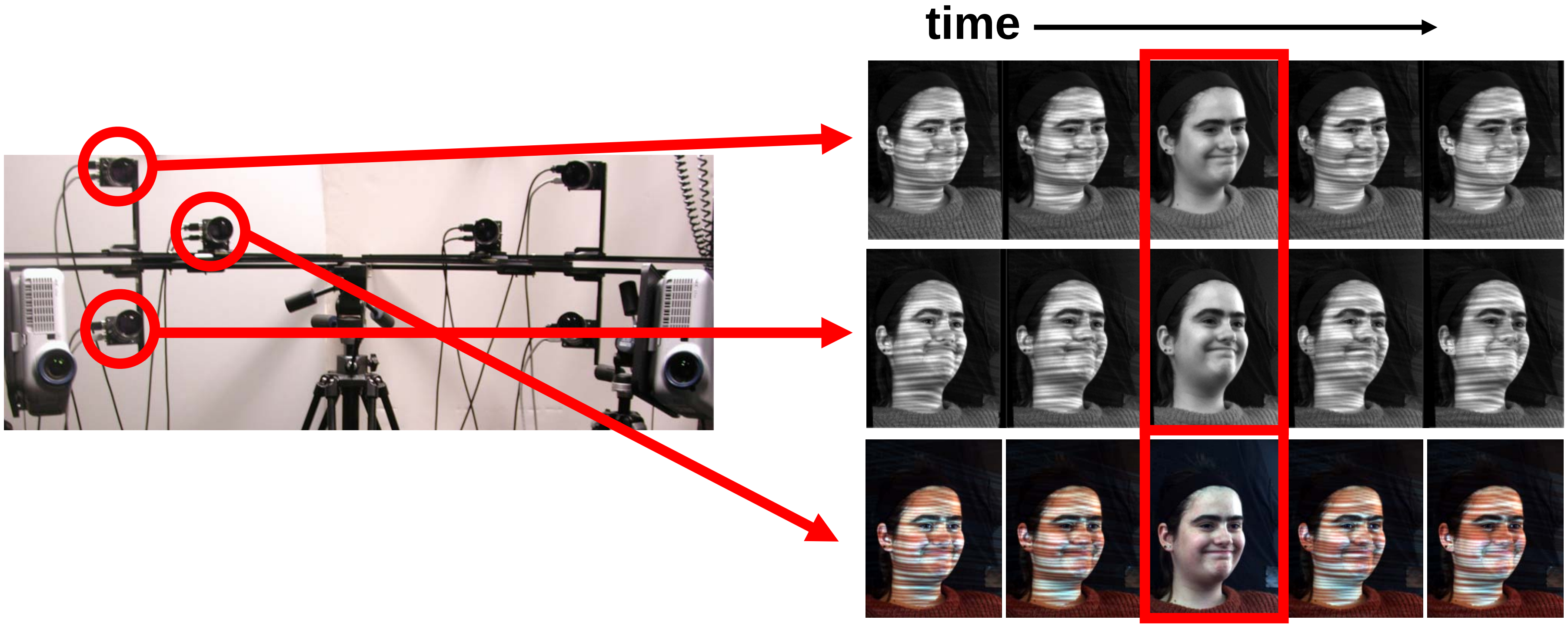
"neutral" → "joy"

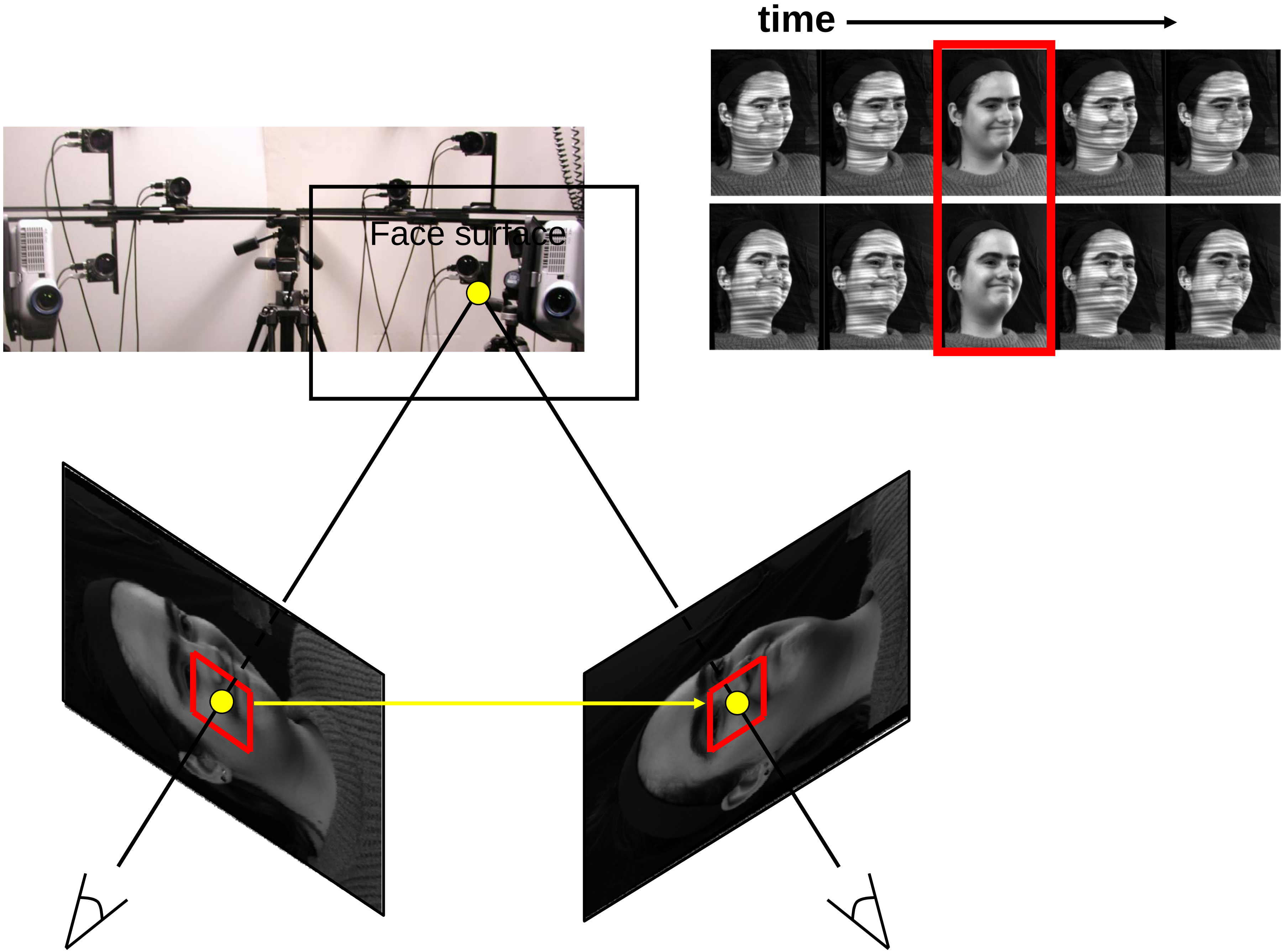
Video

Spacetime faces

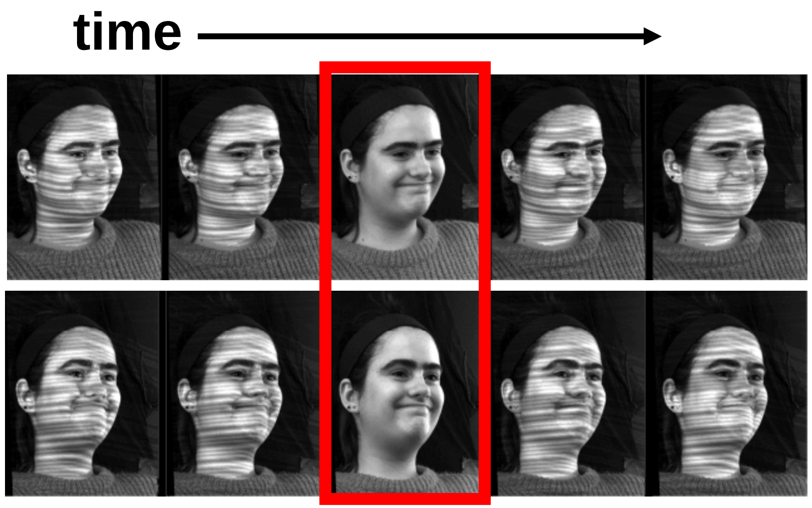
Spacetime faces

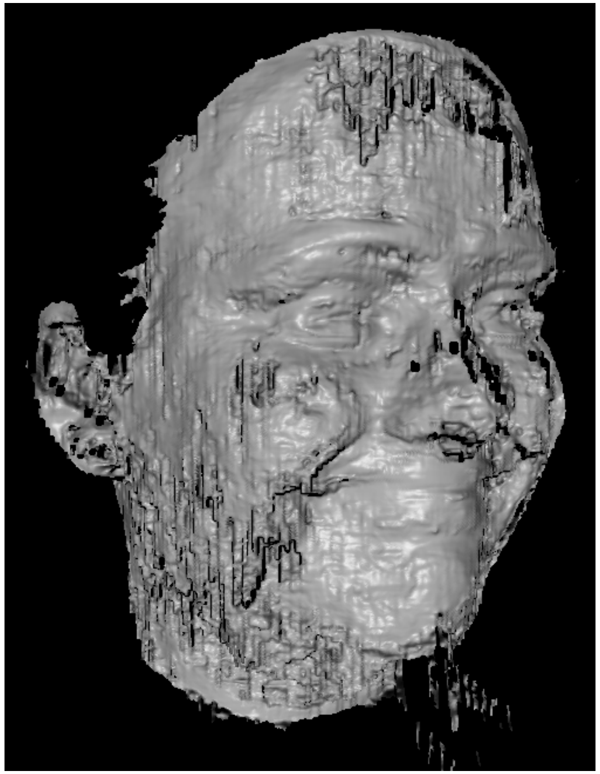



time

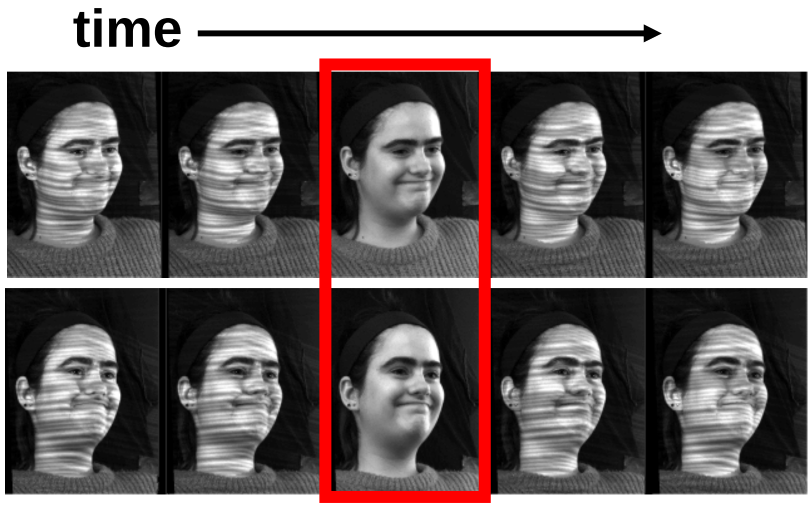


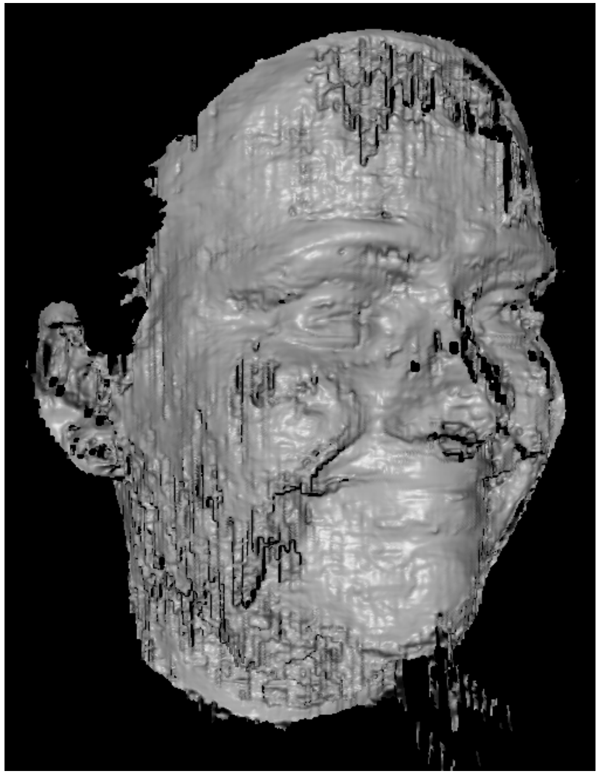
time
Face surface

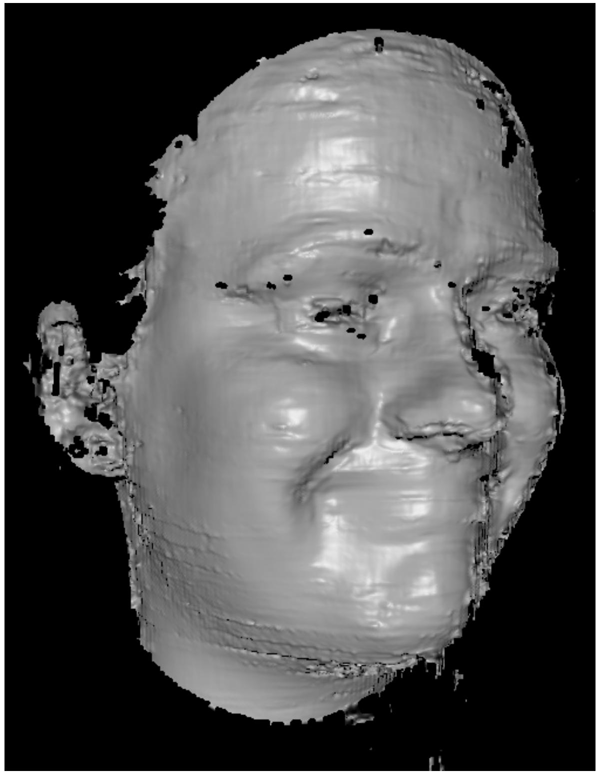


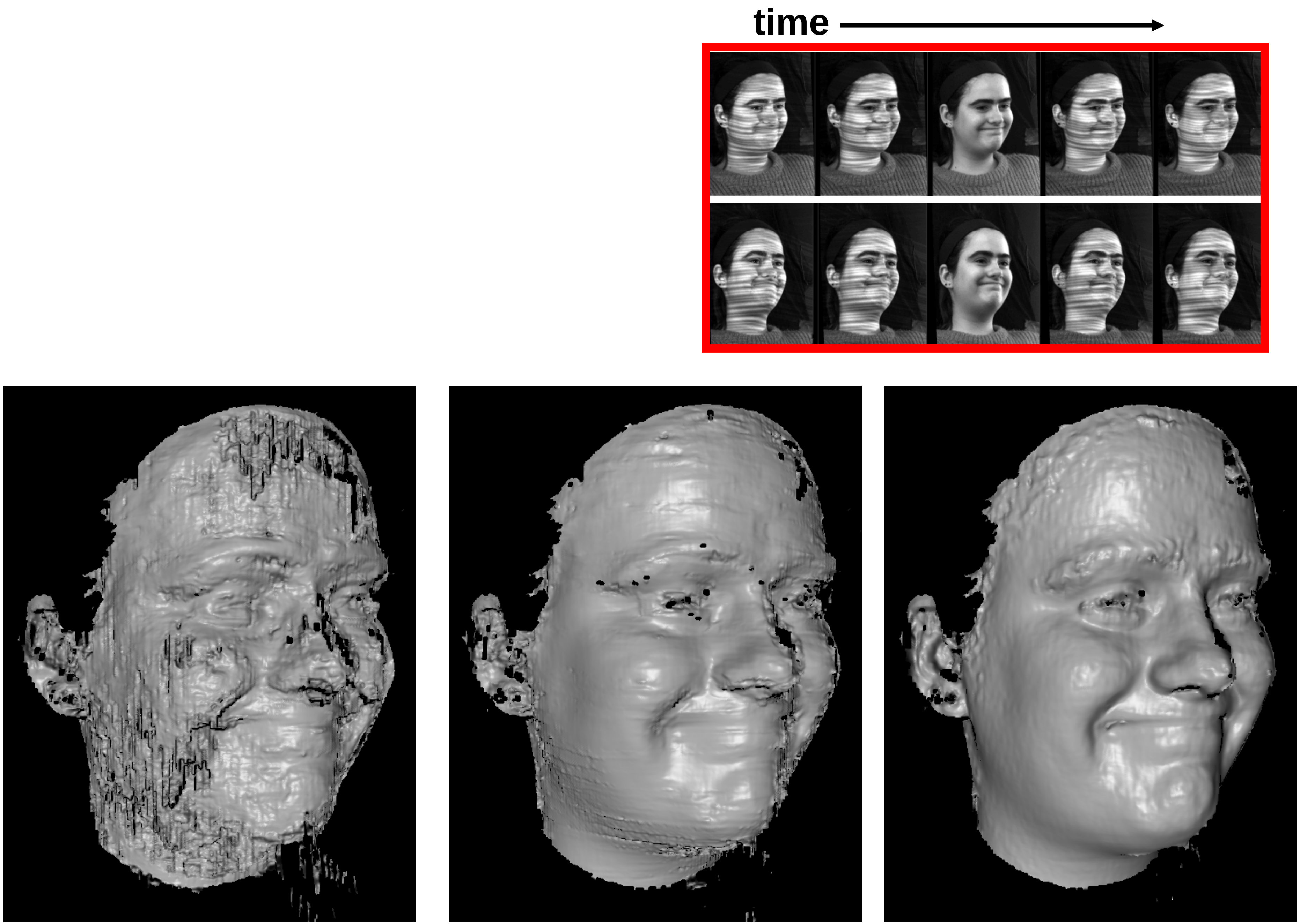


stereo



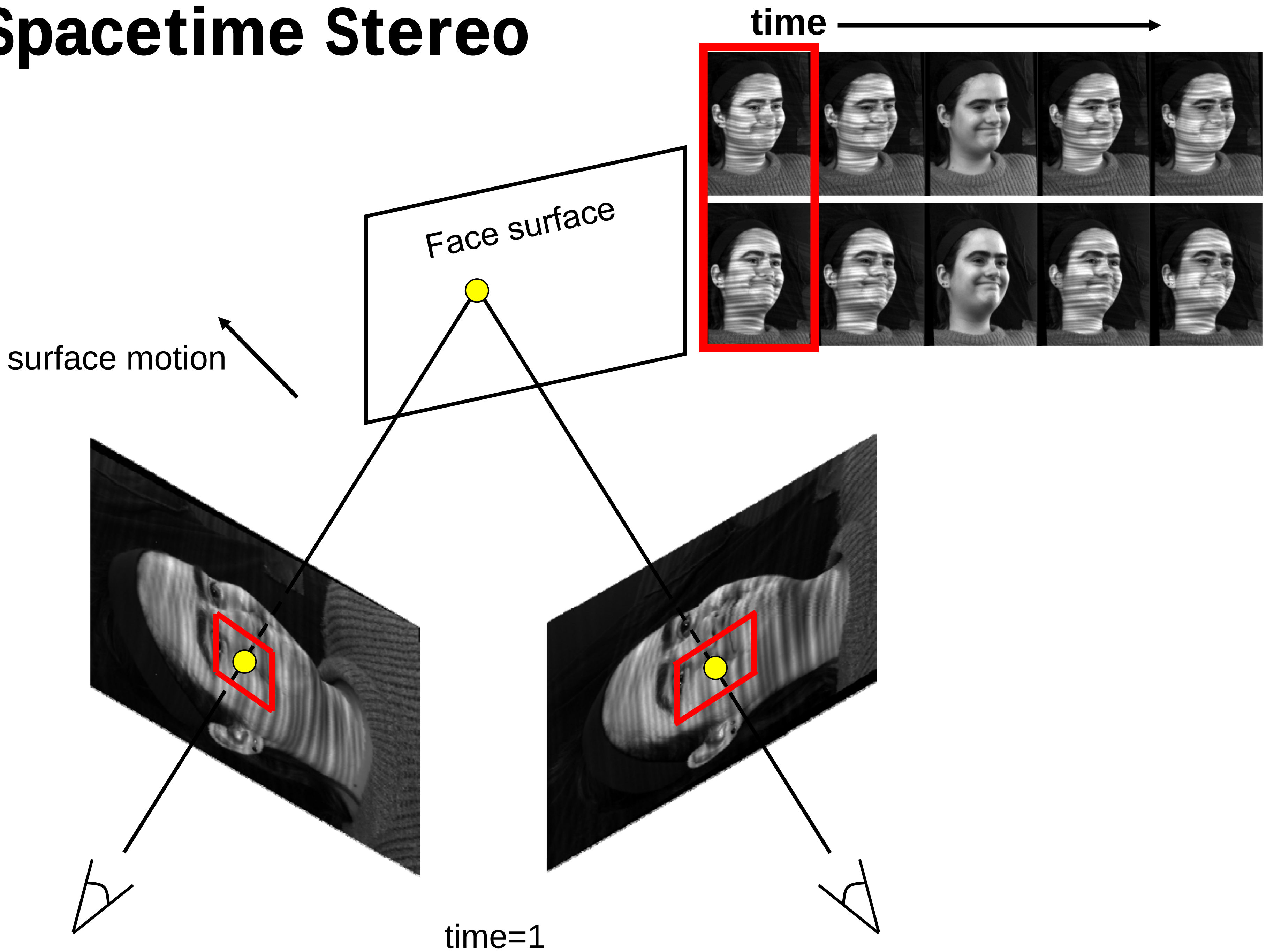

stereo


active stereo

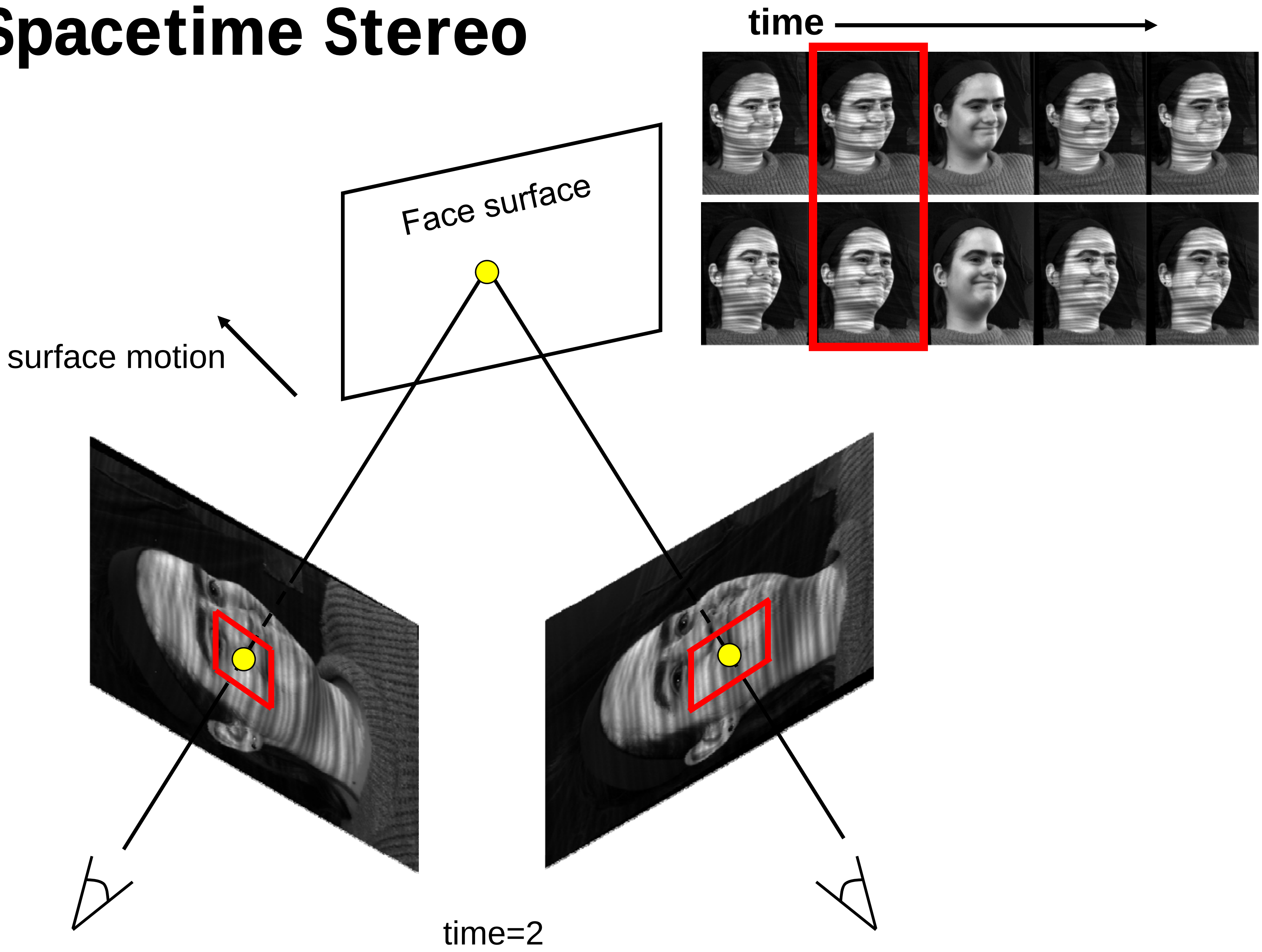


stereo active stereo spacetime stereo

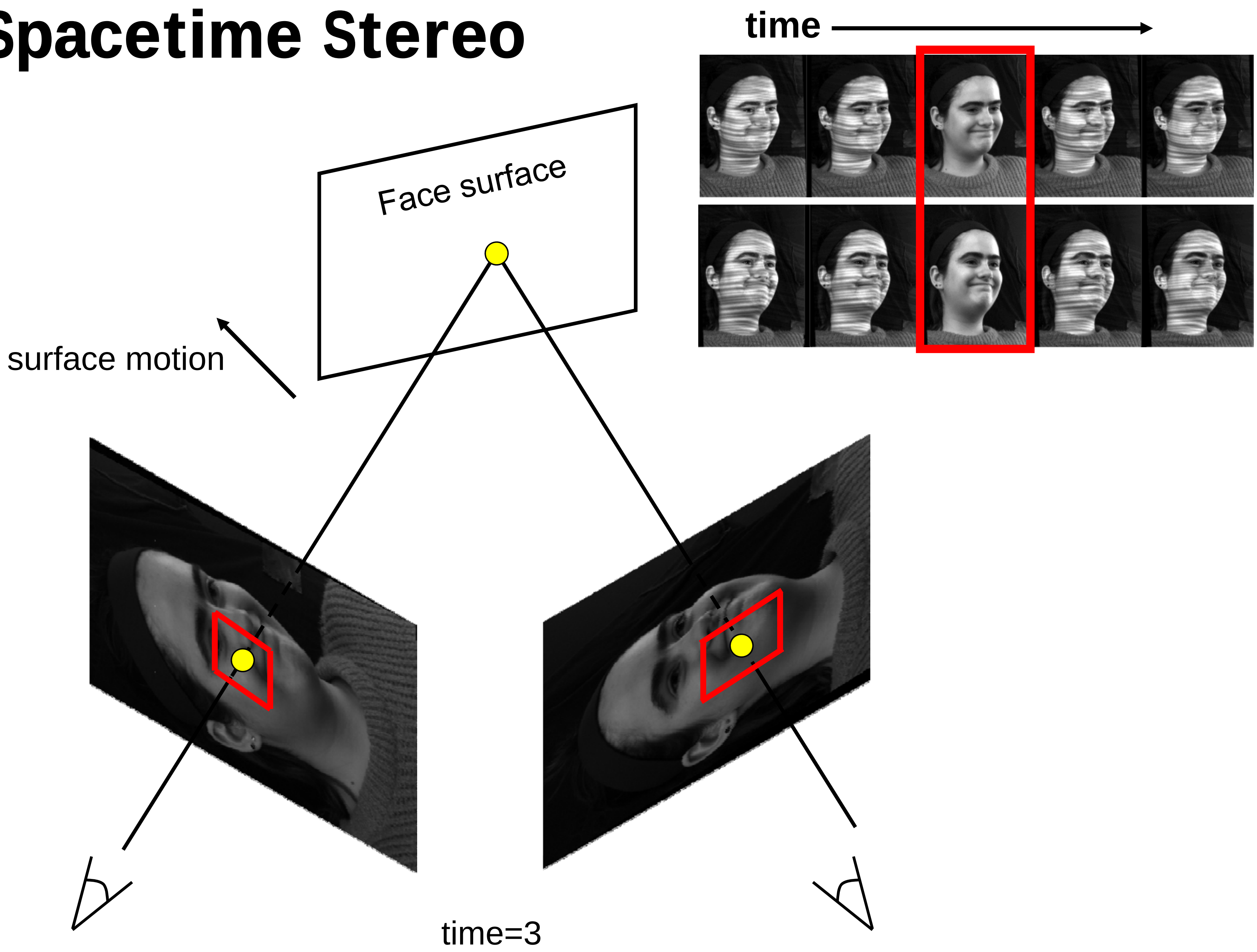
Spacetime Stereo



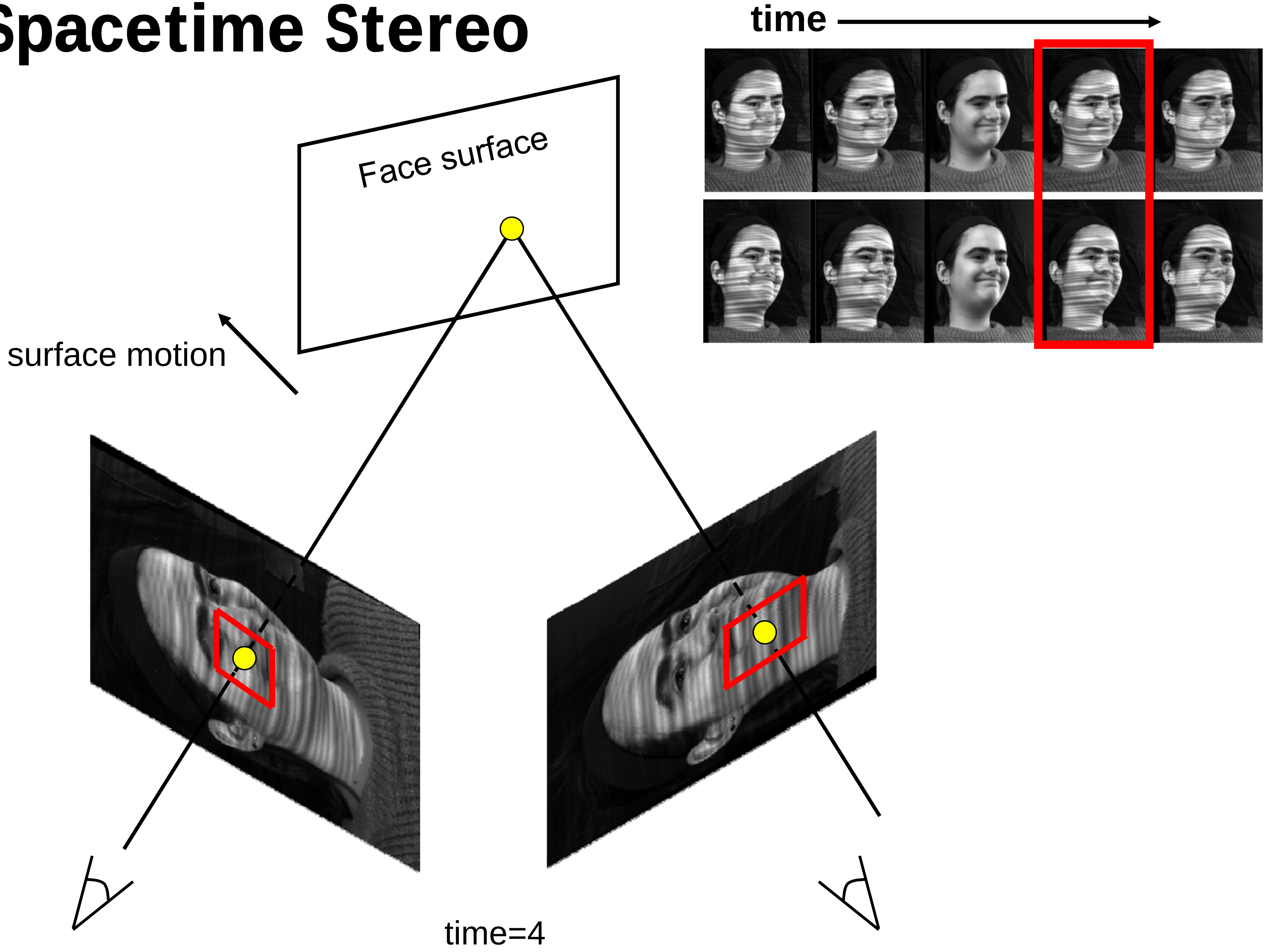
Spacetime Stereo



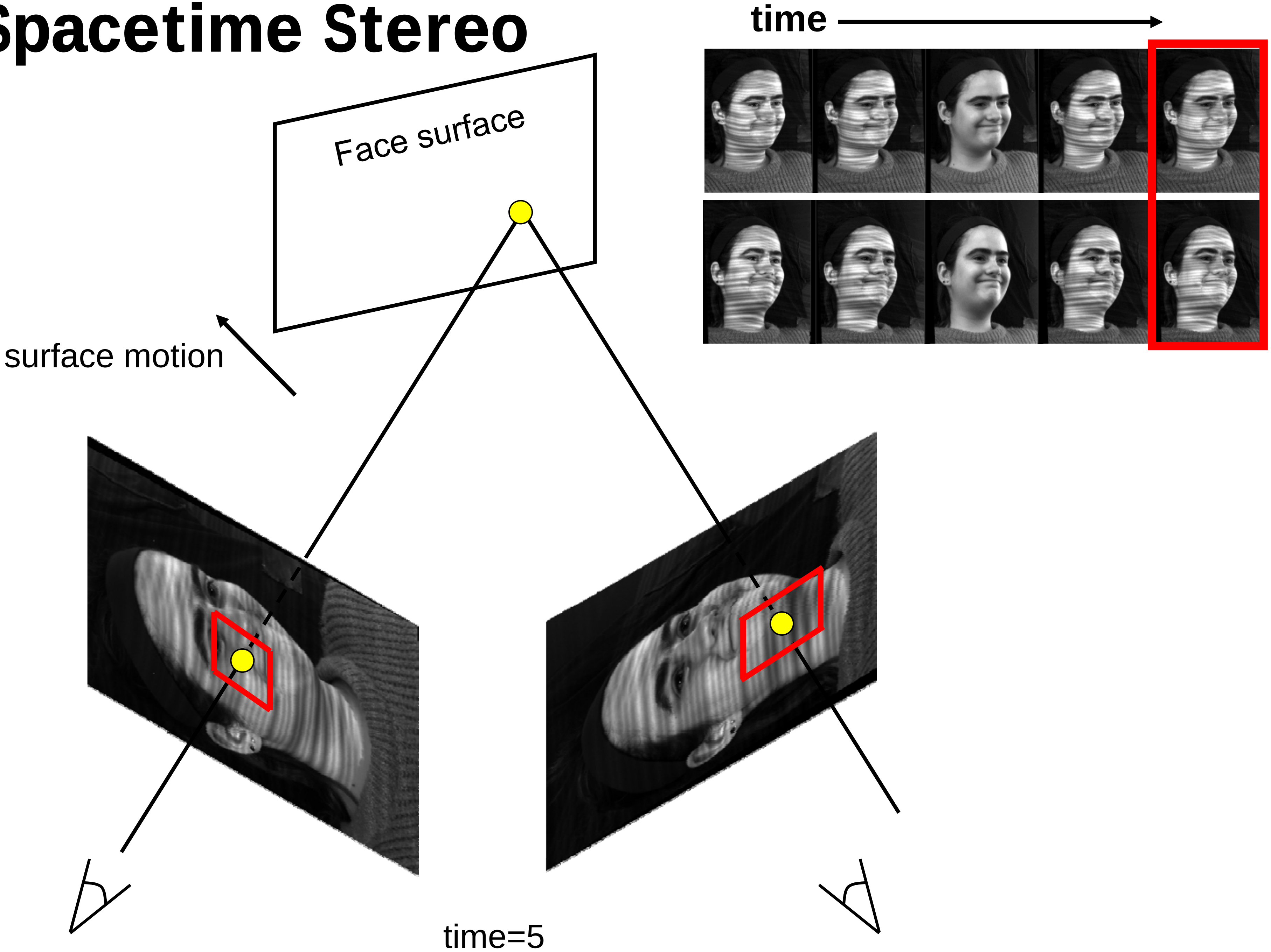
Spacetime Stereo



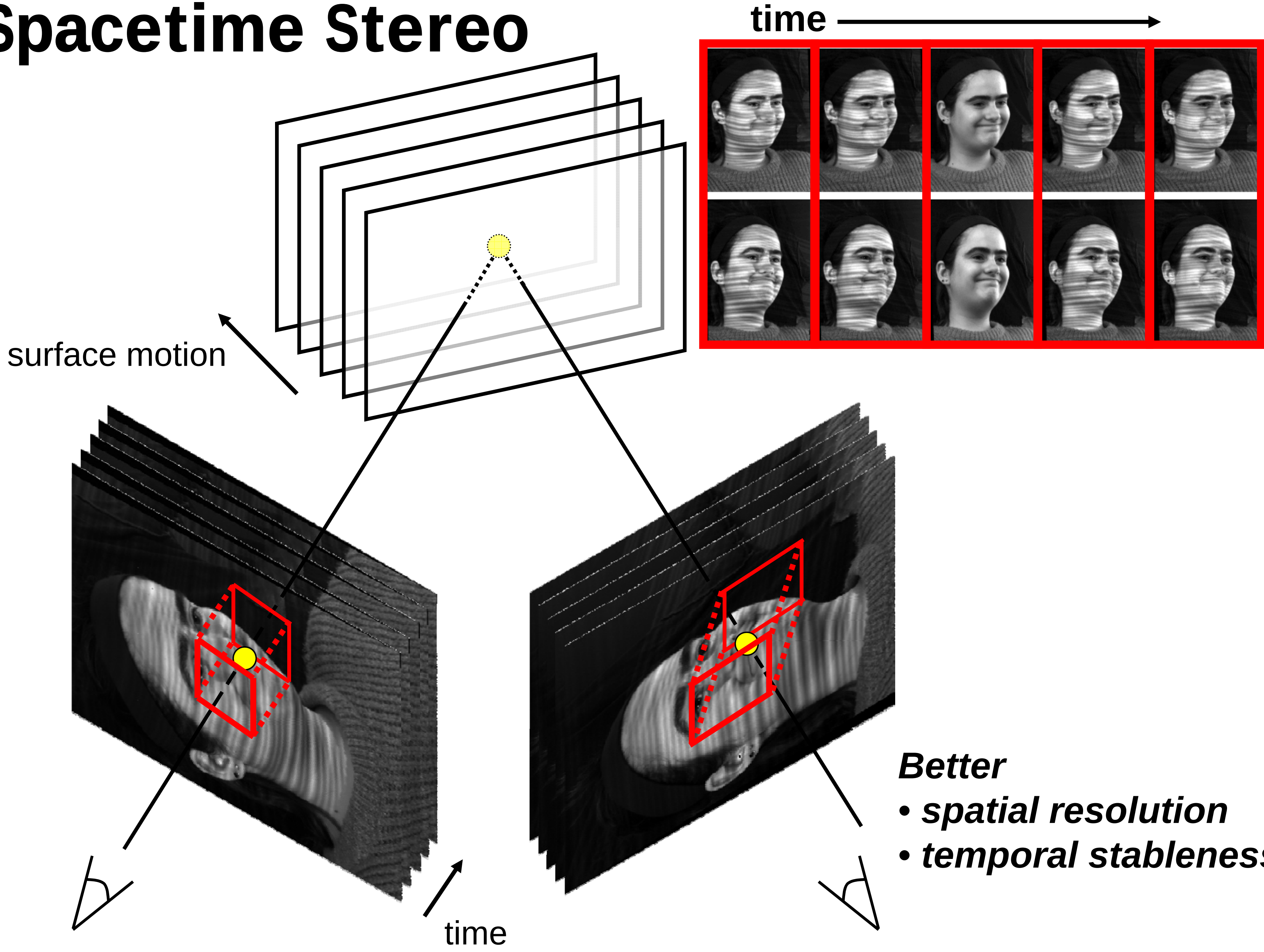
Spacetime Stereo



Spacetime Stereo

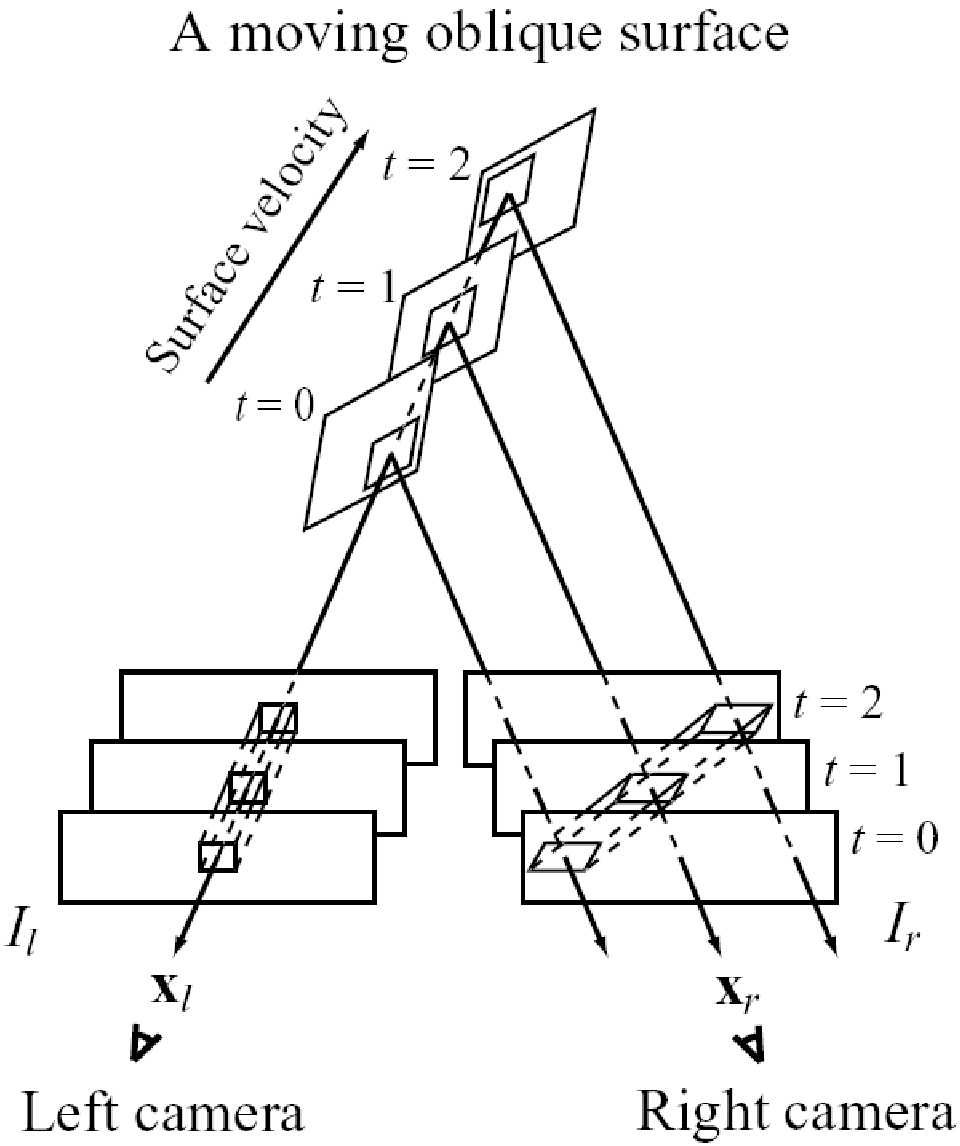


Spacetime Stereo

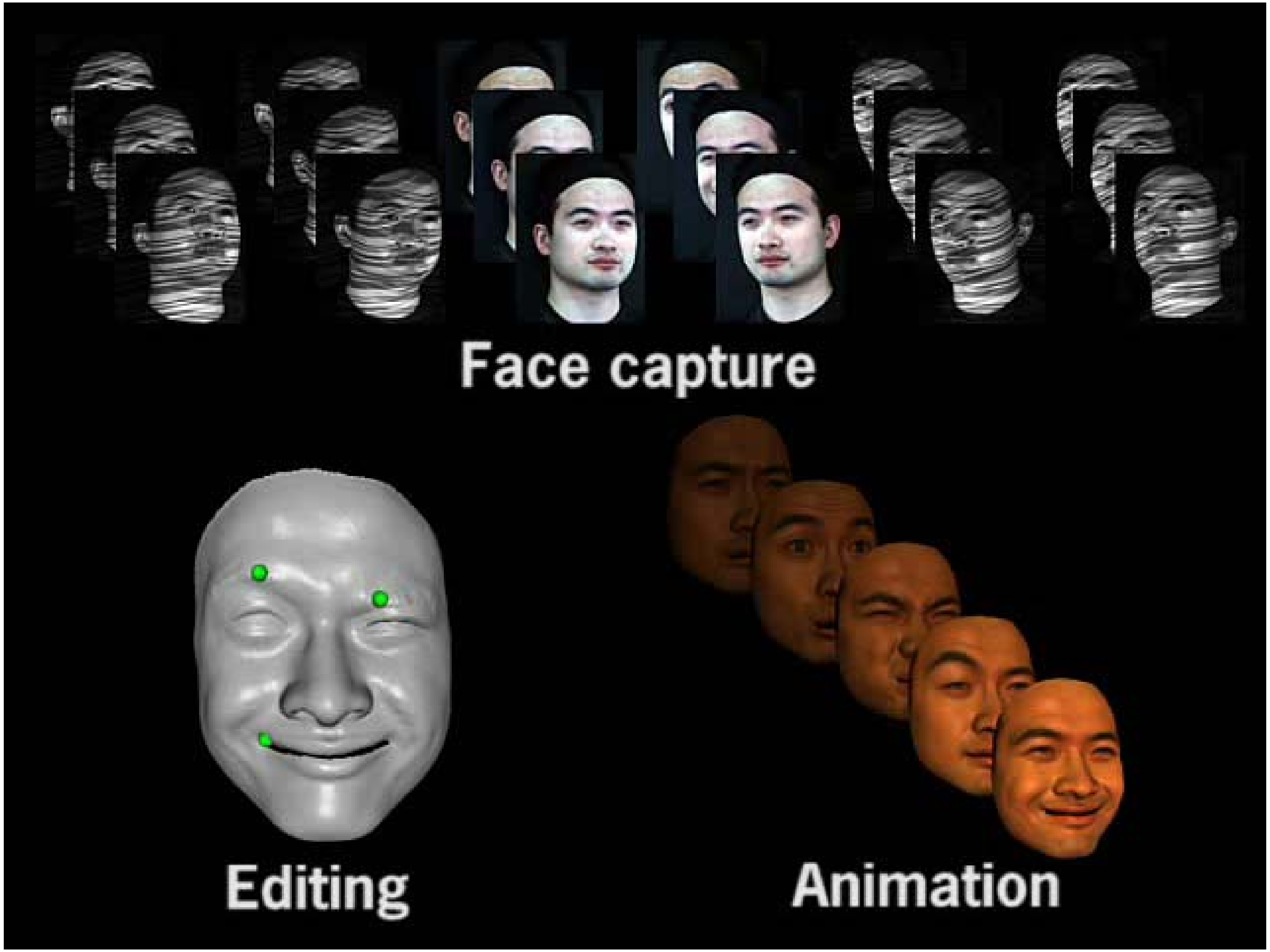


Better
- ***spatial resolution***
- ***temporal stableness***

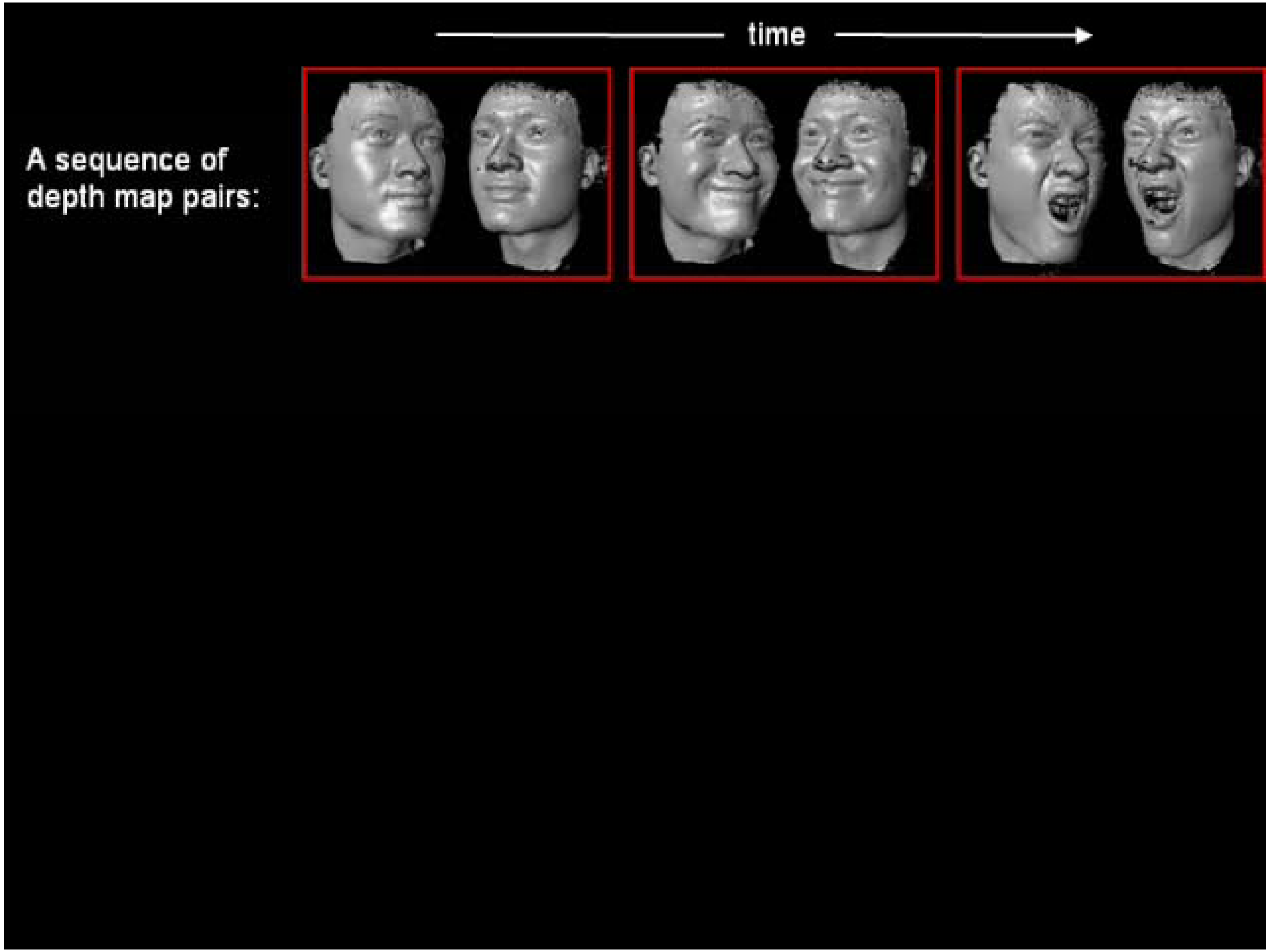
Spacetime stereo matching



Video

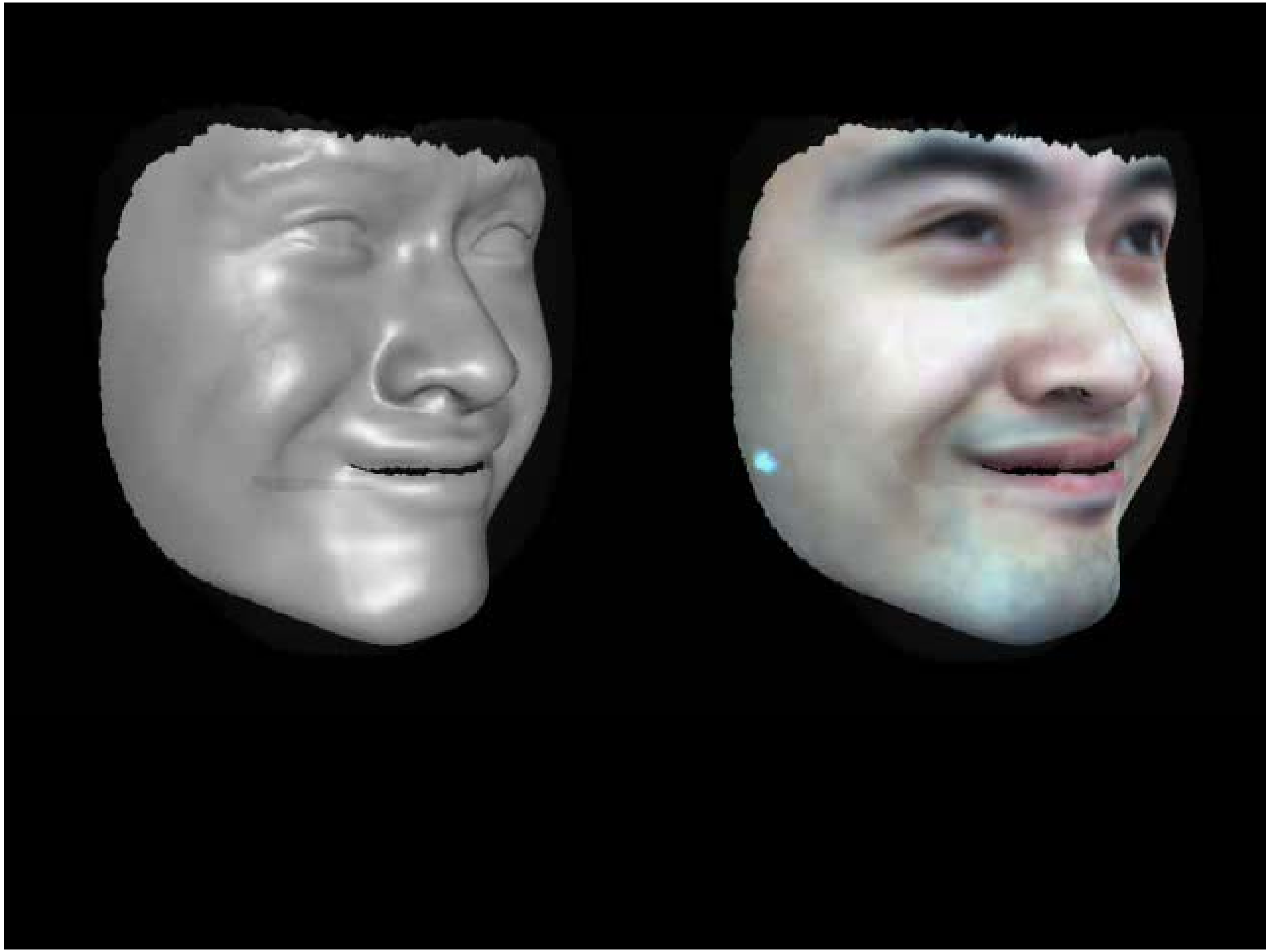


Fitting

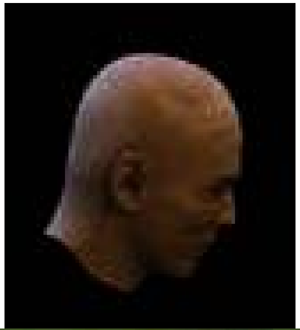


FaceIK

Animation



3D face applications: The one



3D face applications: Gladiator

extra 3M

Statistical methods

Statistical methods

para-meters $z \longrightarrow \boxed{f(z)+\varepsilon} \longrightarrow y$ observed signal

$$
\begin{aligned}
z^* &= \max_z P(z \mid y) \\
&= \max_z \frac{P(y \mid z)P(z)}{P(y)} \\
&= \min_z L(y \mid z) + L(z)
\end{aligned}
$$

Example:
super-resolution
de-noising
de-blocking
Inpainting
…

Statistical methods

para-
meters

$$z \longrightarrow \boxed{f(z)+\varepsilon} \longrightarrow y$$

observed
signal

$$z^* = \min_z L(y \mid z) + L(z)$$

data
evidence

$$\frac{\|y - f(z)\|^2}{\sigma^2}$$

a-priori
knowledge

Statistical methods

There are approximately 10^{240} possible 10×10 gray-level images. Even human being has not seen them all yet. There must be a strong statistical bias.

Takeo Kanade

Approximately $8X10^{11}$ blocks per day per person.

Generic priors

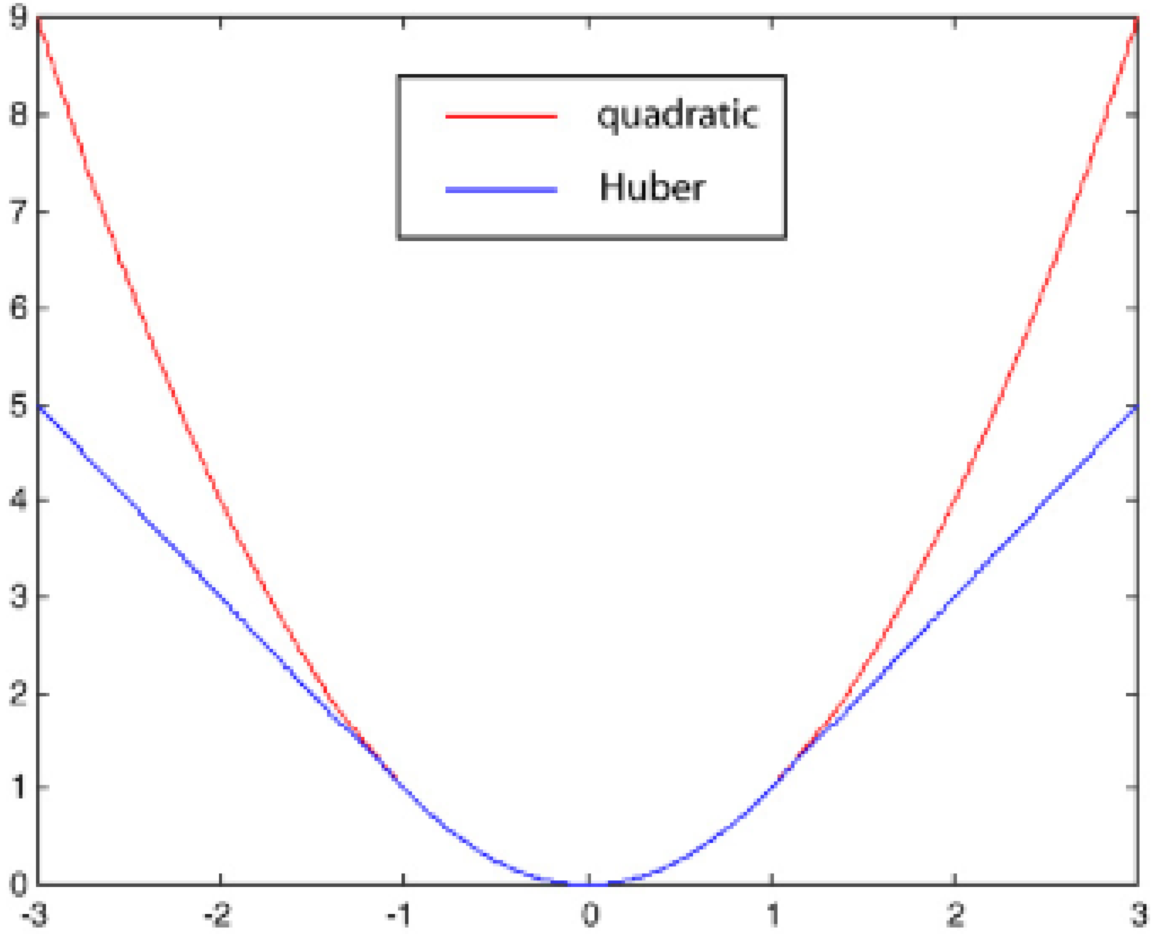
"Smooth images are good images."

$$L(z) = \sum_{x} \rho(V(x))$$

Gaussian MRF $\rho(d) = d^2$

Huber MRF $$\rho(d) = \begin{cases} d^2 & |d| \le T \\ T^2 + 2T(|d| - T) & d > T \end{cases}$$

Generic priors

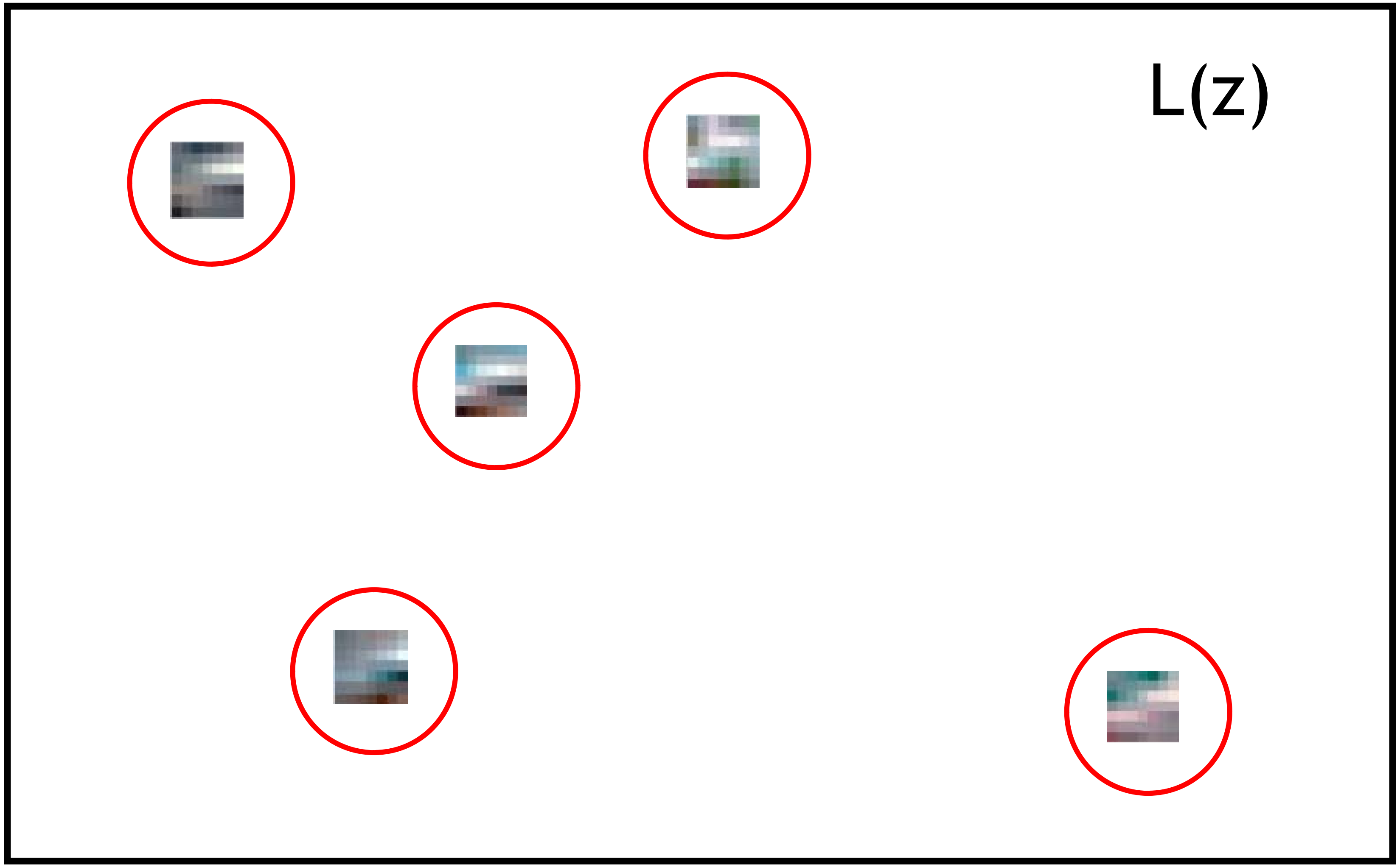


Example-based priors

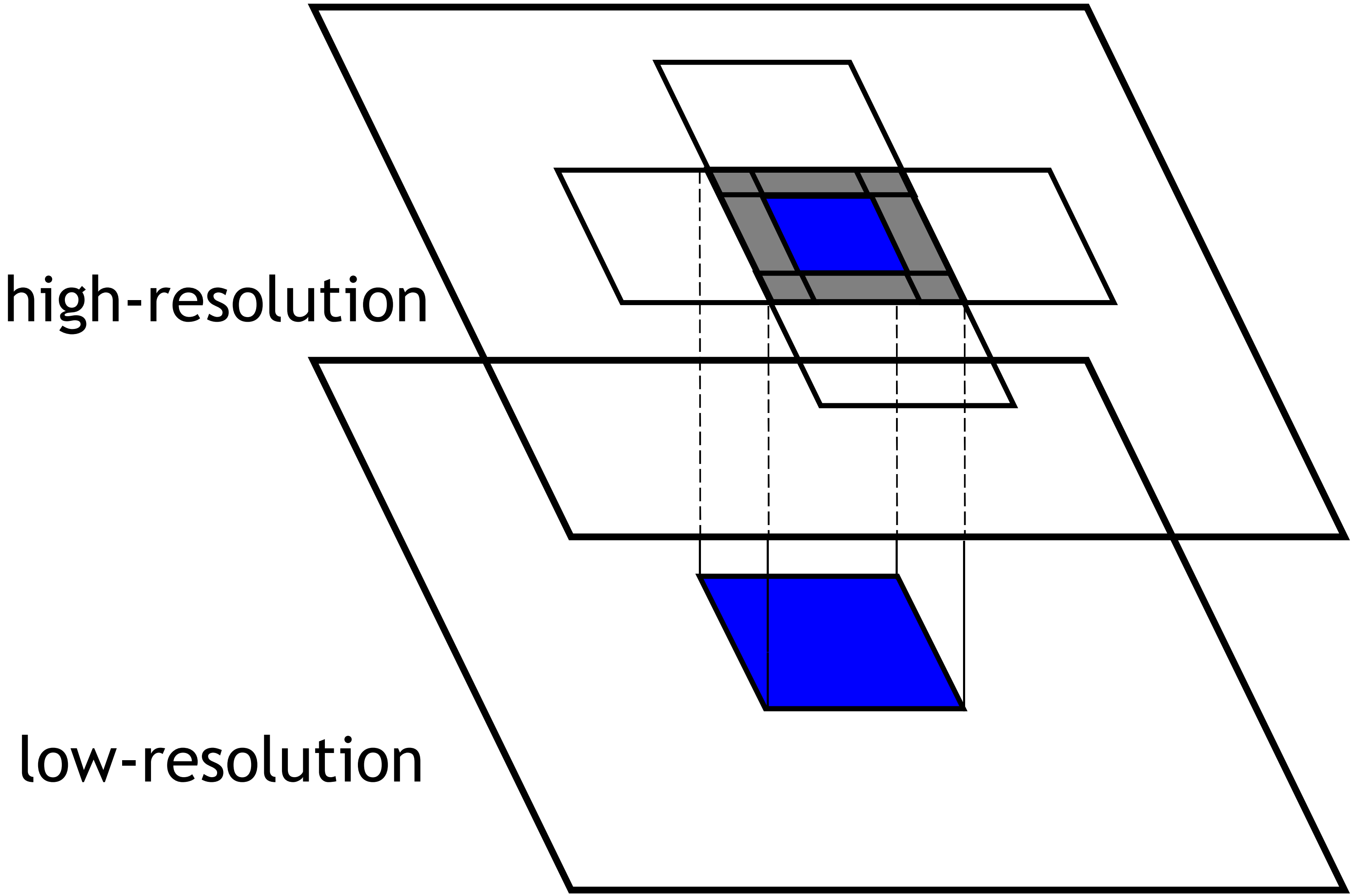
"Existing images are good images."

six 200×200 Images ⇒ 2,000,000 pairs

Example-based priors



Example-based priors



Model-based priors

"Face images are good images when working on face images …"

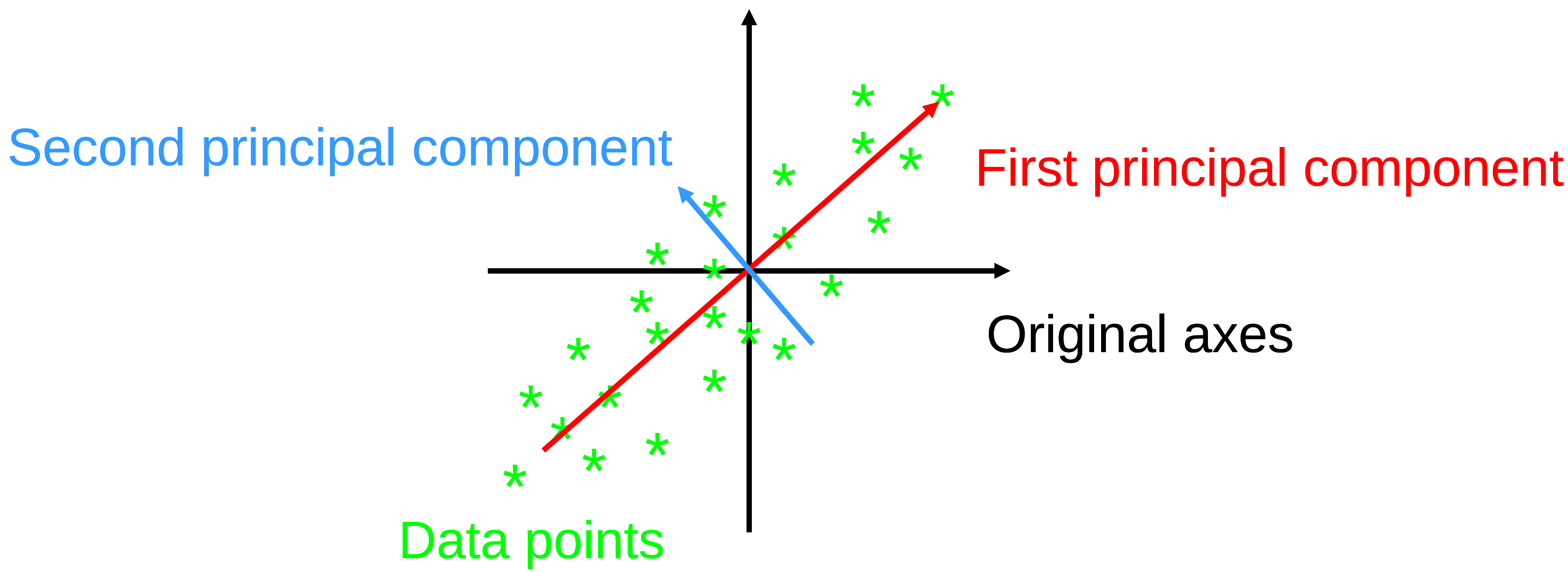
Parametric model $\quad Z=WX+\mu \quad L(X)$

$$z^* = \min_z L(y \mid z) + L(z)$$

$$\begin{cases} X^* = \min_X L(y \mid WX + \mu) + L(X) \\ z^* = WX^* + \mu \end{cases}$$

PCA

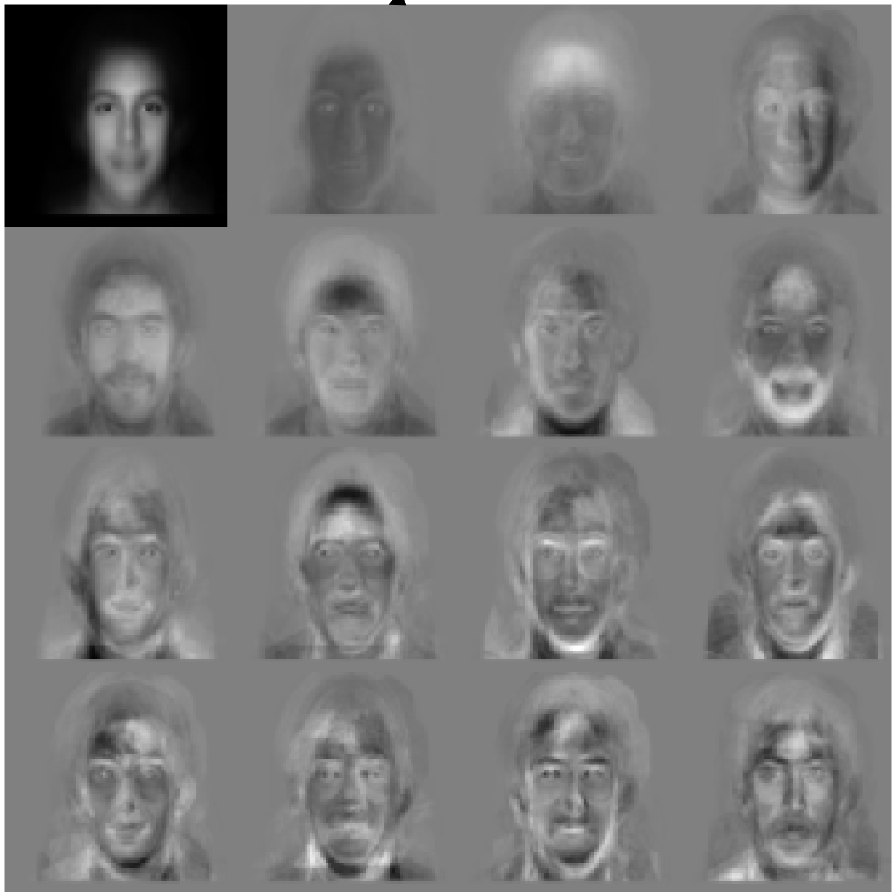
- Principal Components Analysis (PCA): approximating a high-dimensional data set with a lower-dimensional subspace



PCA on faces: "eigenfaces"

First principal component

Average face



Other components

For all except average,
"gray" = 0,
"white" > 0,
"black" < 0

Model-based priors

"Face images are good images when working on face images …"

Parametric model $\quad Z=WX+\mu \quad L(X)$

$$z^* = \min_z L(y \mid z) + L(z)$$

$$\begin{cases} X^* = \min_X L(y \mid WX + \mu) + L(X) \\ z^* = WX^* + \mu \end{cases}$$

Super-resolution

(a) (b) (c) (d) (e) (f)

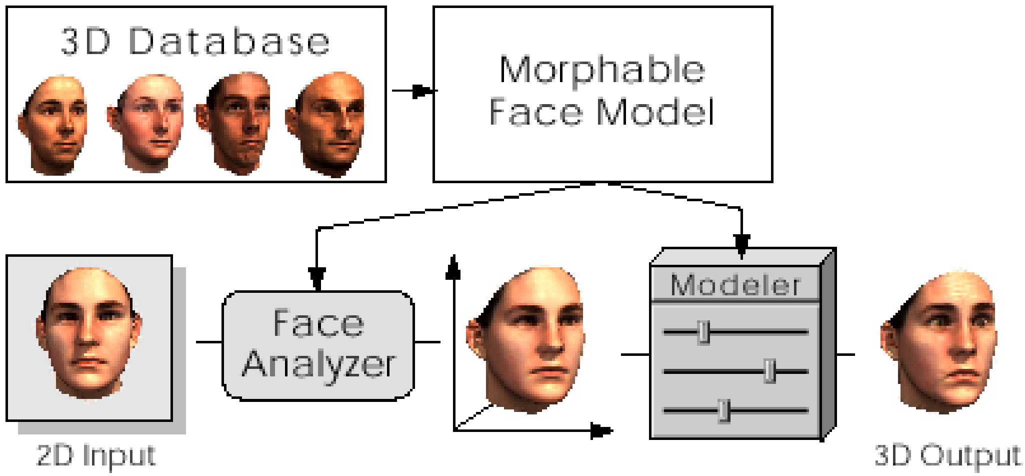
(a) Input low 24×32 (b) Our results (c) Cubic B-Spline

(d) Freeman et al. (e) Baker et al. (f) Original high 96×128

Face models from single images

Morphable model of 3D faces

- Start with a catalogue of 200 aligned 3D Cyberware scans



- Build a model of *average* shape and texture, and principal *variations* using PCA

Morphable model

shape examplars texture examplars

$$S_{model} = \overline{S} + \sum_{i=1}^{m-1} \alpha_i s_i \,, \quad T_{model} = \overline{T} + \sum_{i=1}^{m-1} \beta_i t_i \,, \qquad (1)$$

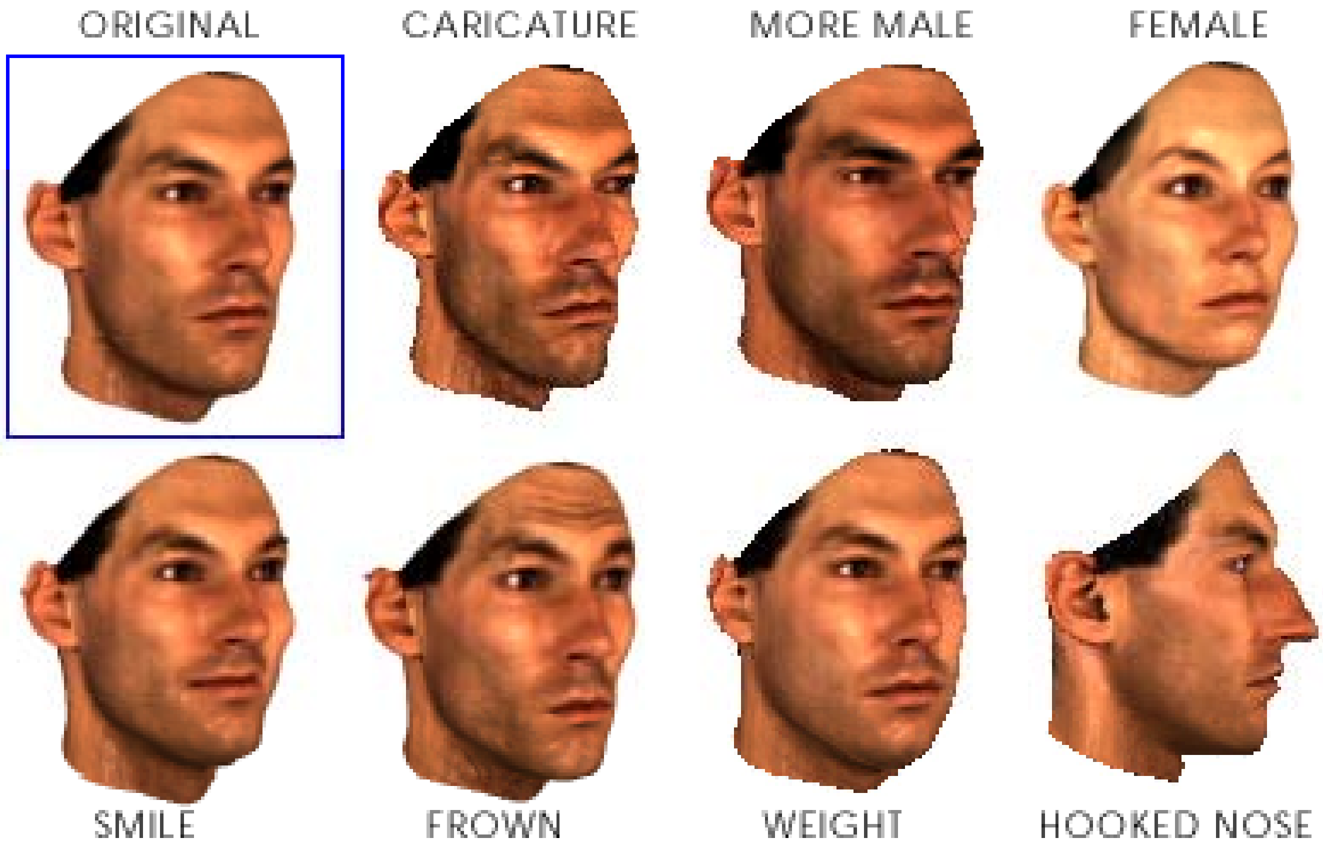
$\vec{\alpha}, \vec{\beta} \in \Re^{m-1}$. The probability for coefficients $\vec{\alpha}$ is given by

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

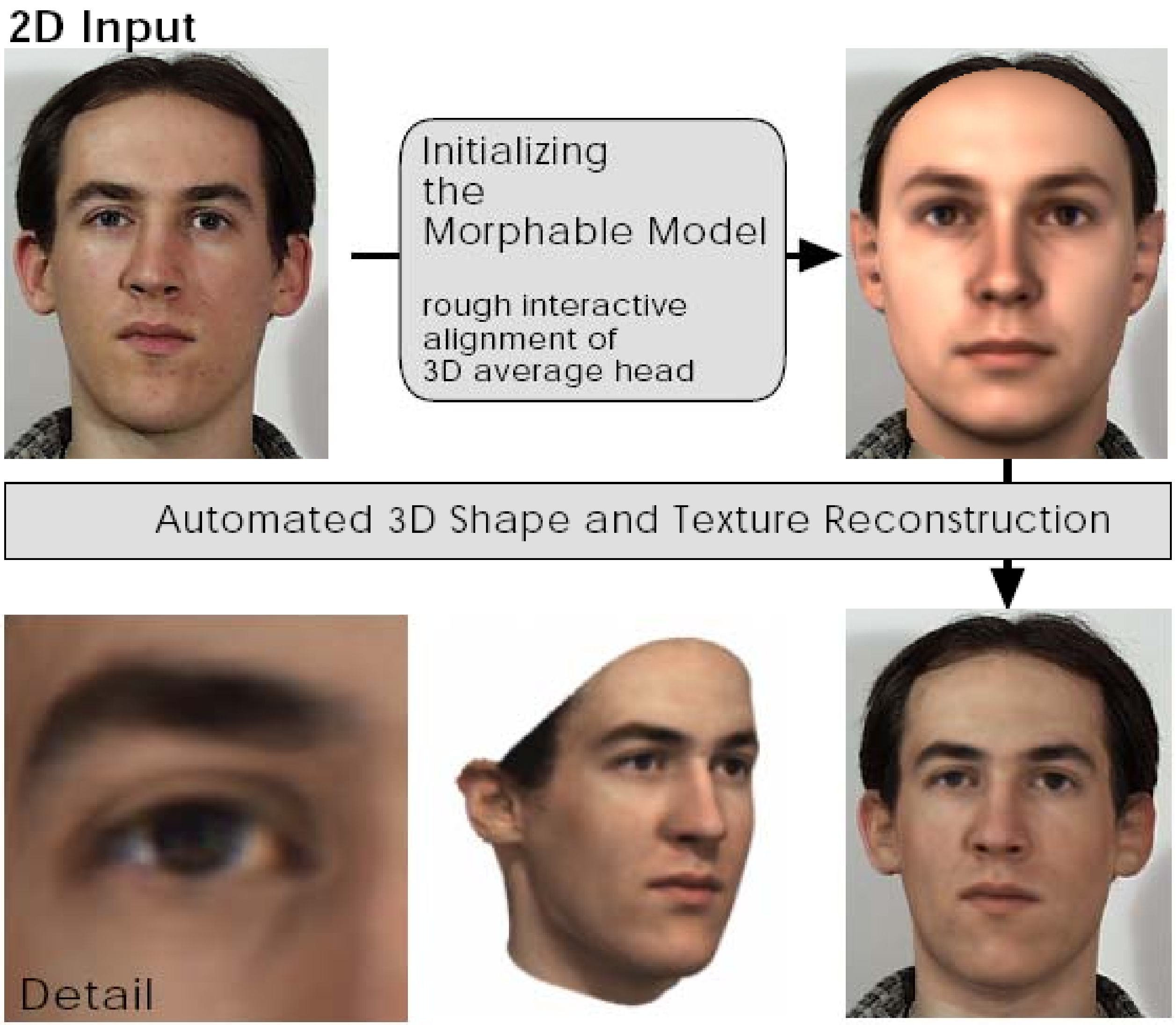
Morphable model of 3D faces

DigiVFX

- Adding some variations



Reconstruction from single image



$\alpha_j \quad \beta_j$

Rendering must be similar to the input if we guess right

Reconstruction from single image

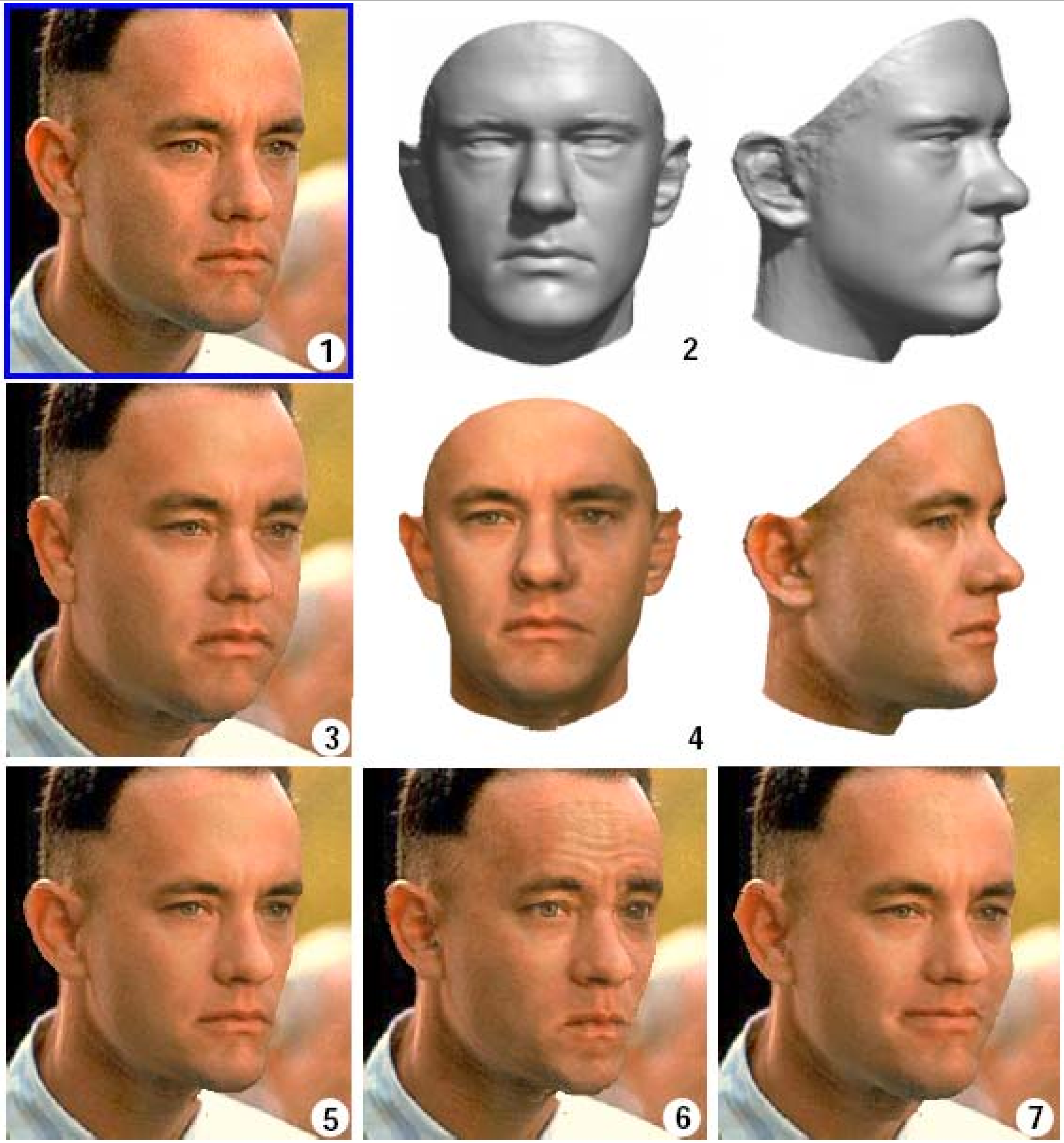
$$E = \frac{1}{\sigma_N^2}\boxed{E_I} + \boxed{\sum_{j=1}^{m-1}\boxed{\frac{\alpha_j^2}{\sigma_{S,j}^2}} + \sum_{j=1}^{m-1}\boxed{\frac{\beta_j^2}{\sigma_{T,j}^2}} + \sum_j \boxed{\frac{(\rho_j - \bar{\rho}_j)^2}{\sigma_{\rho,j}^2}}}$$ prior

$$E_I = \sum_{x,y} \|\mathbf{I}_{input}(x,y) - \mathbf{I}_{model}(x,y)\|^2$$

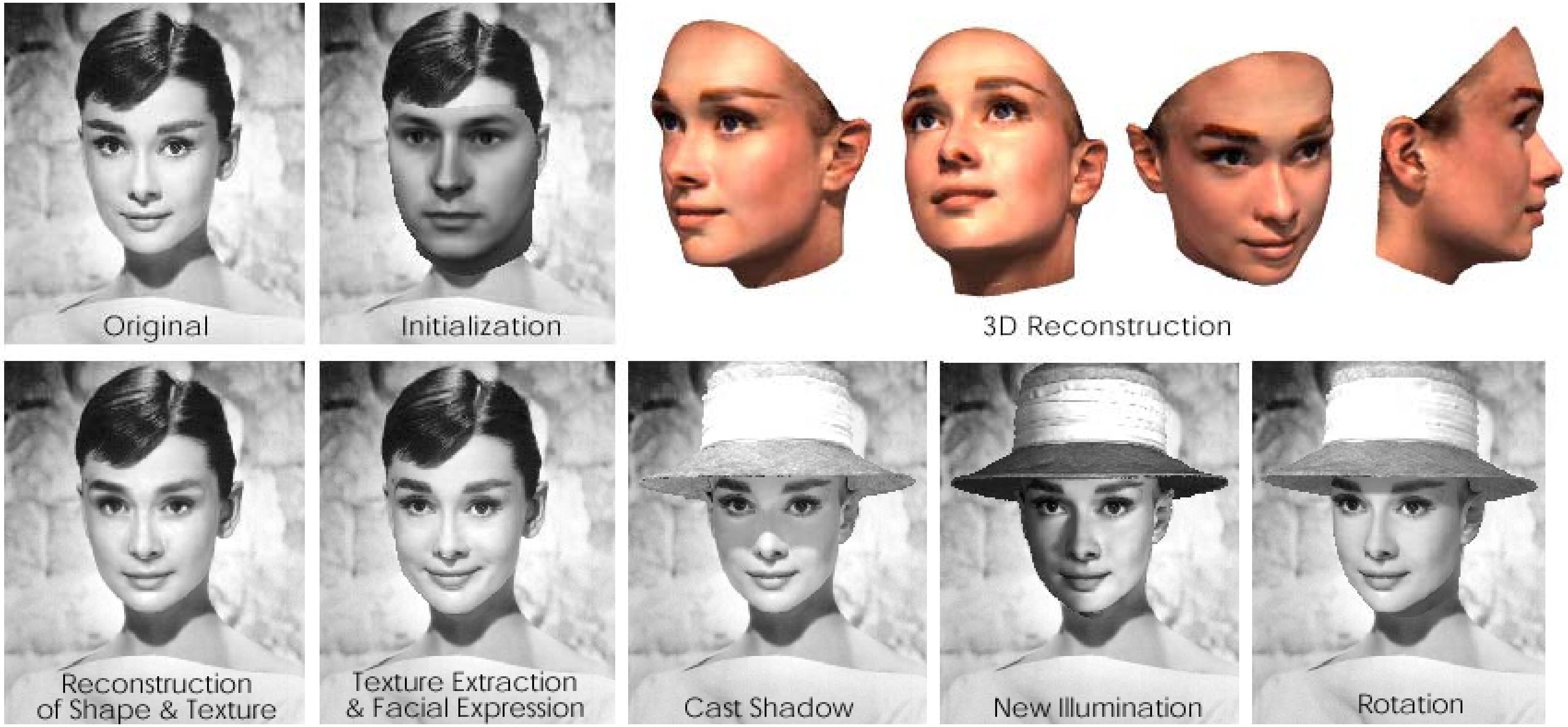
shape and texture priors are learnt from database

ρ is the set of parameters for shading including camera pose, lighting and so on

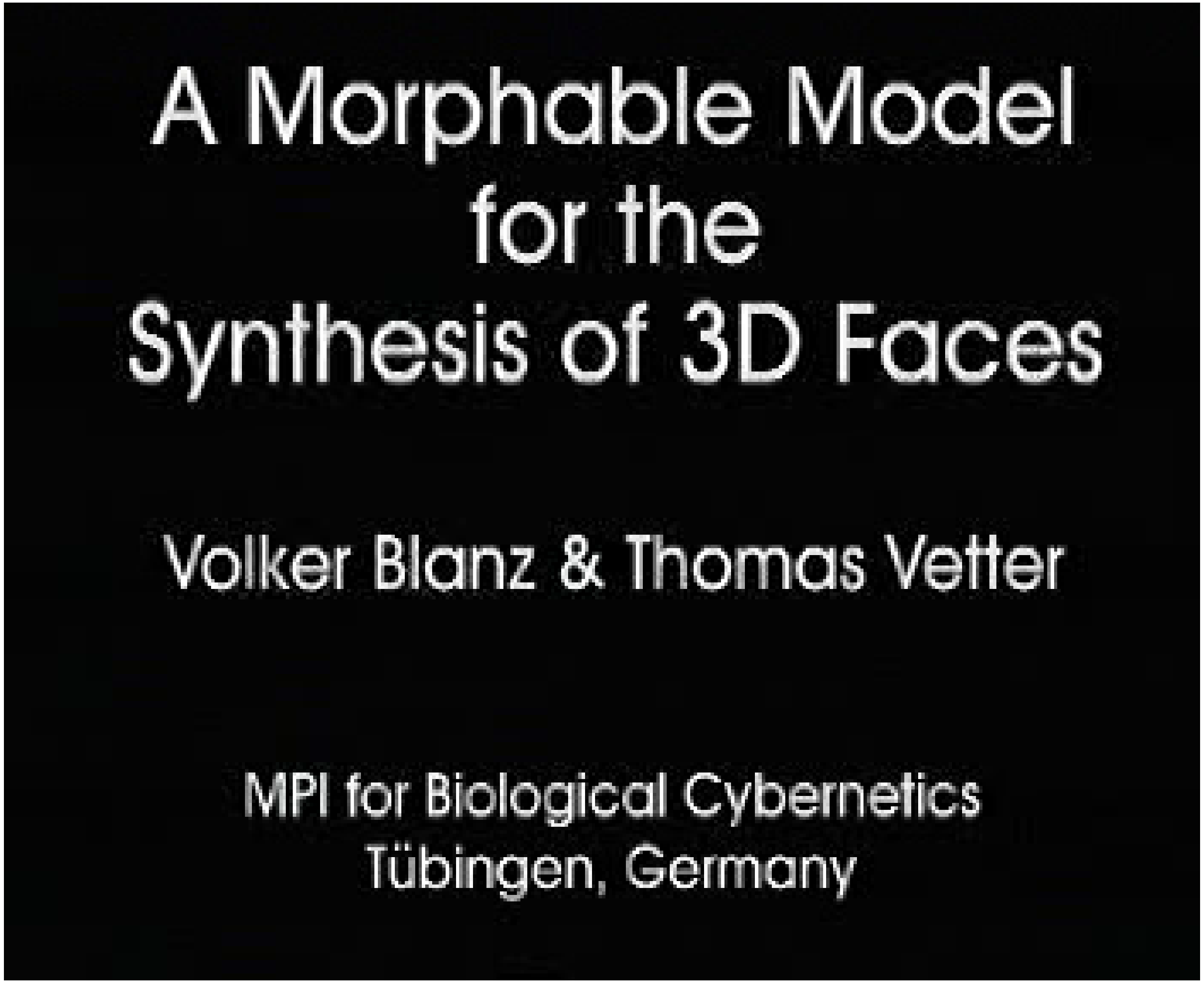
Modifying a single image



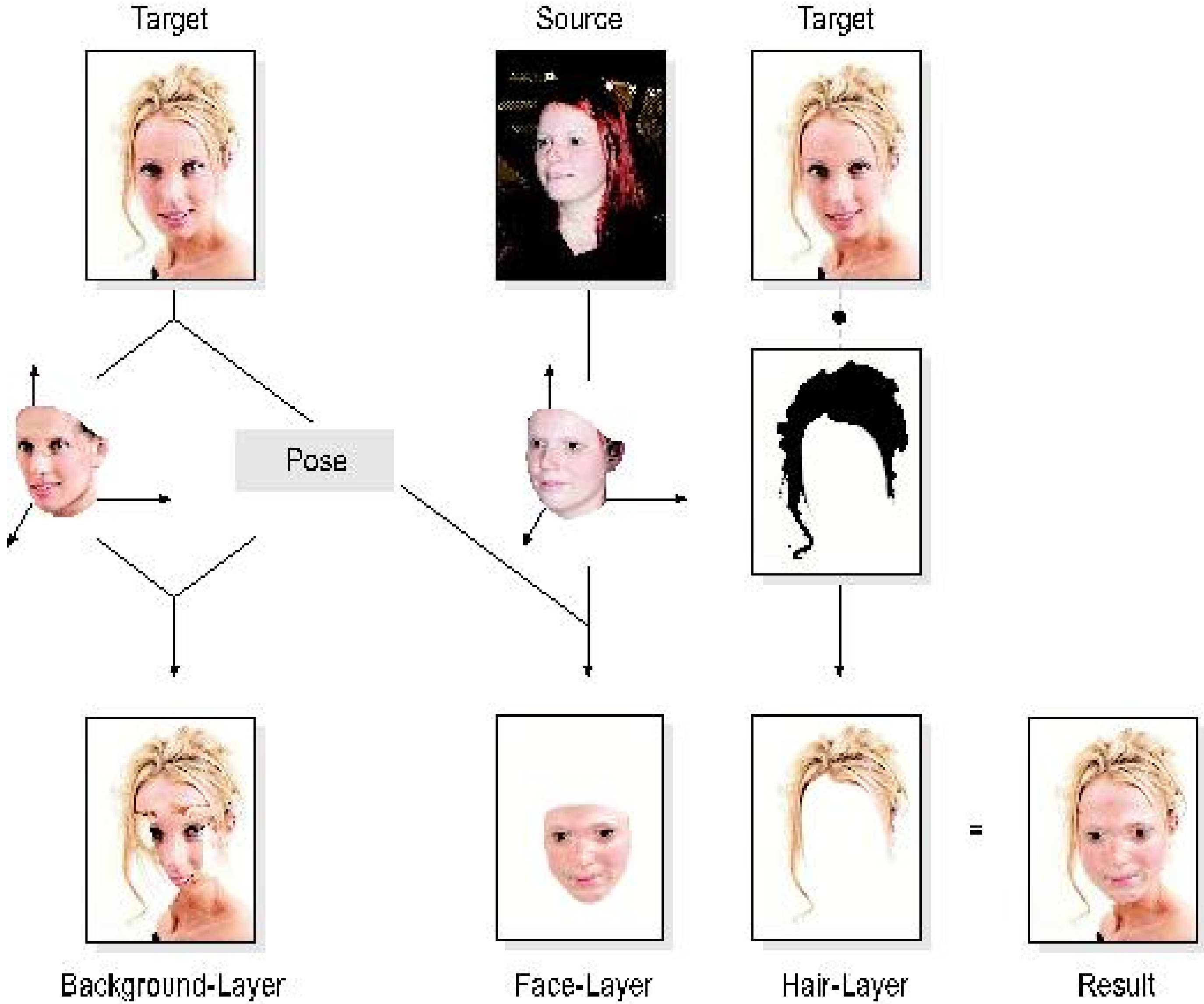
Animating from a single image



Video


A Morphable Model
for the
Synthesis of 3D Faces
Volker Blanz & Thomas Vetter
MPI for Biological Cybernetics
Tübingen, Germany

Exchanging faces in images



Exchange faces in images

Exchange faces in images

Exchange faces in images

Exchange faces in images

Morphable model for human body

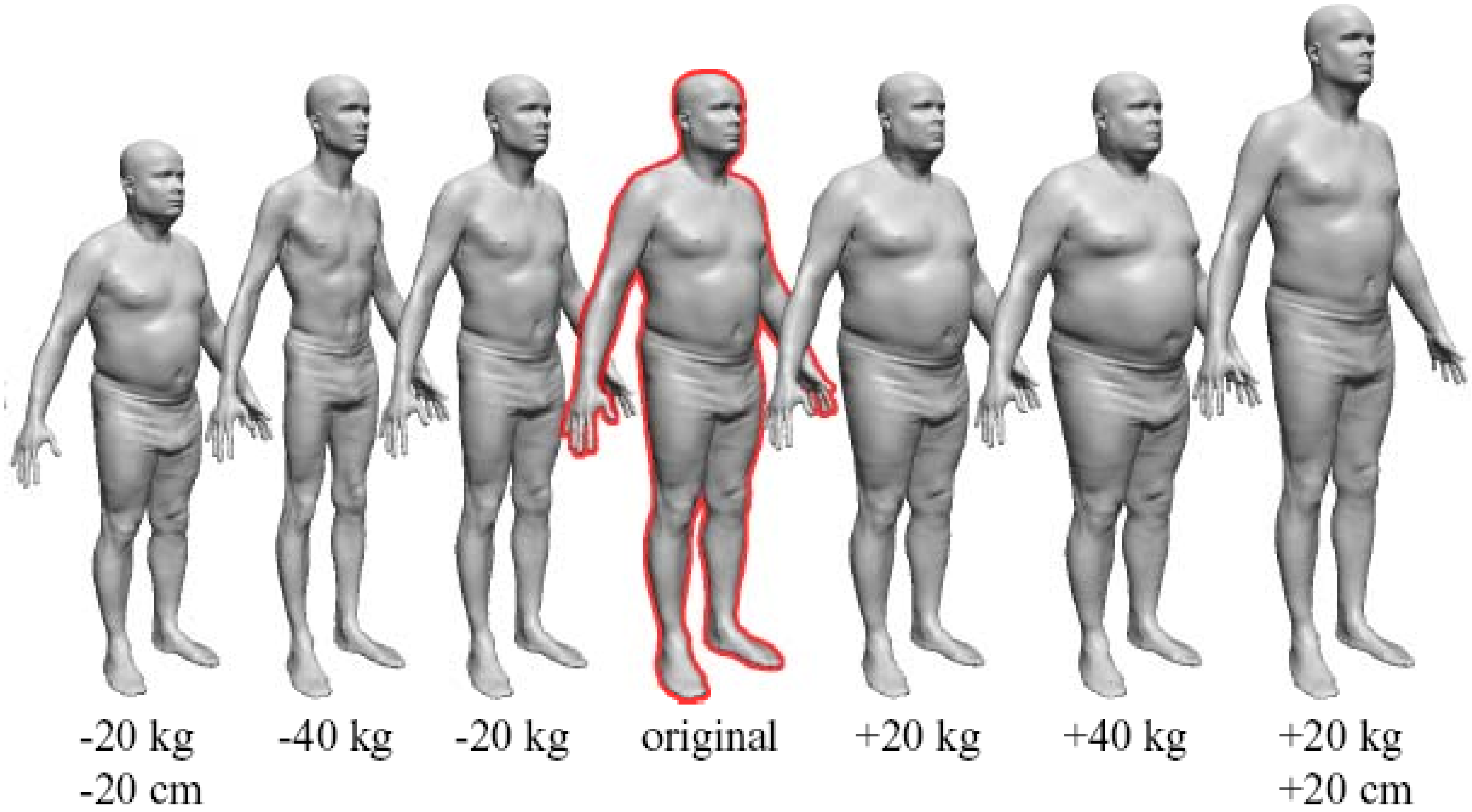
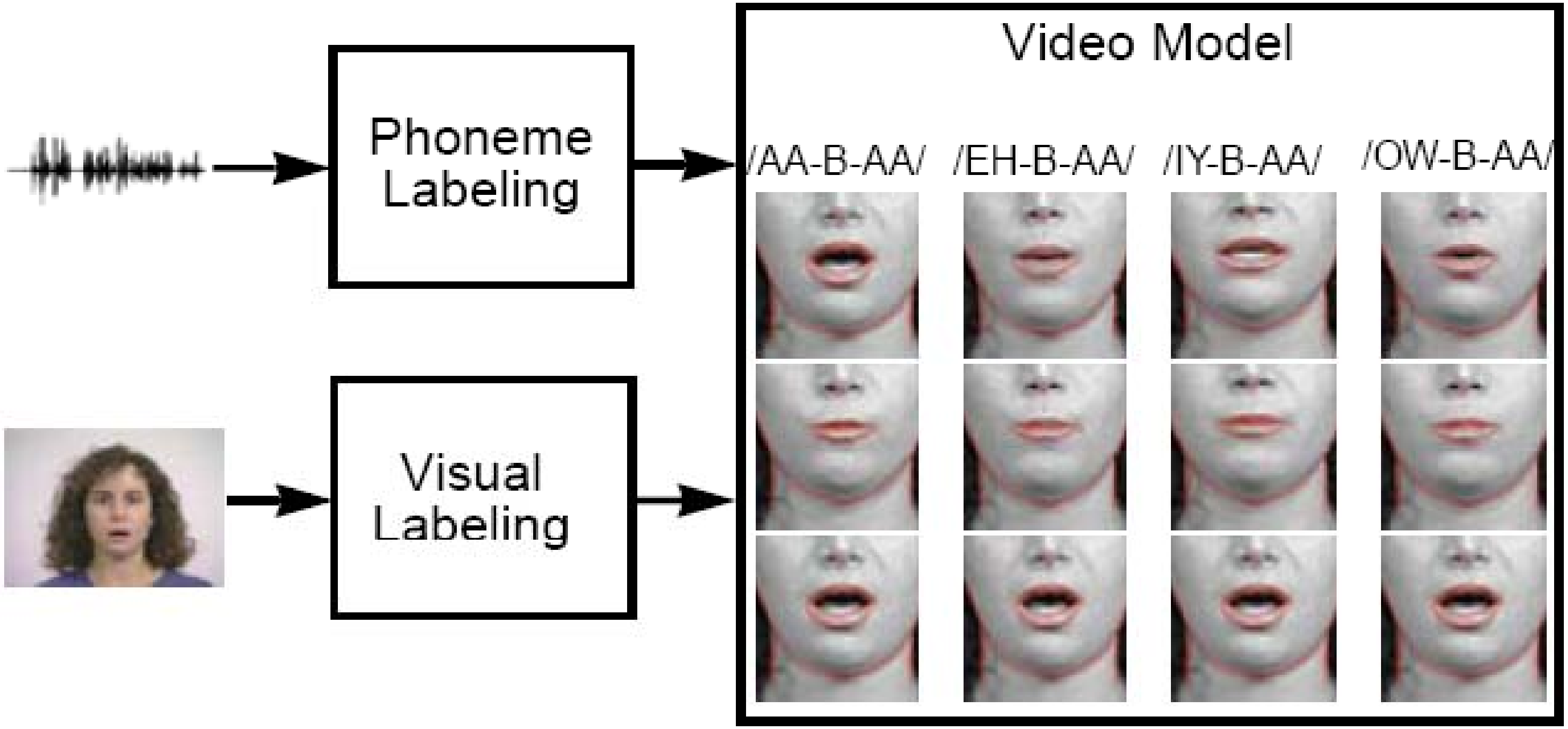
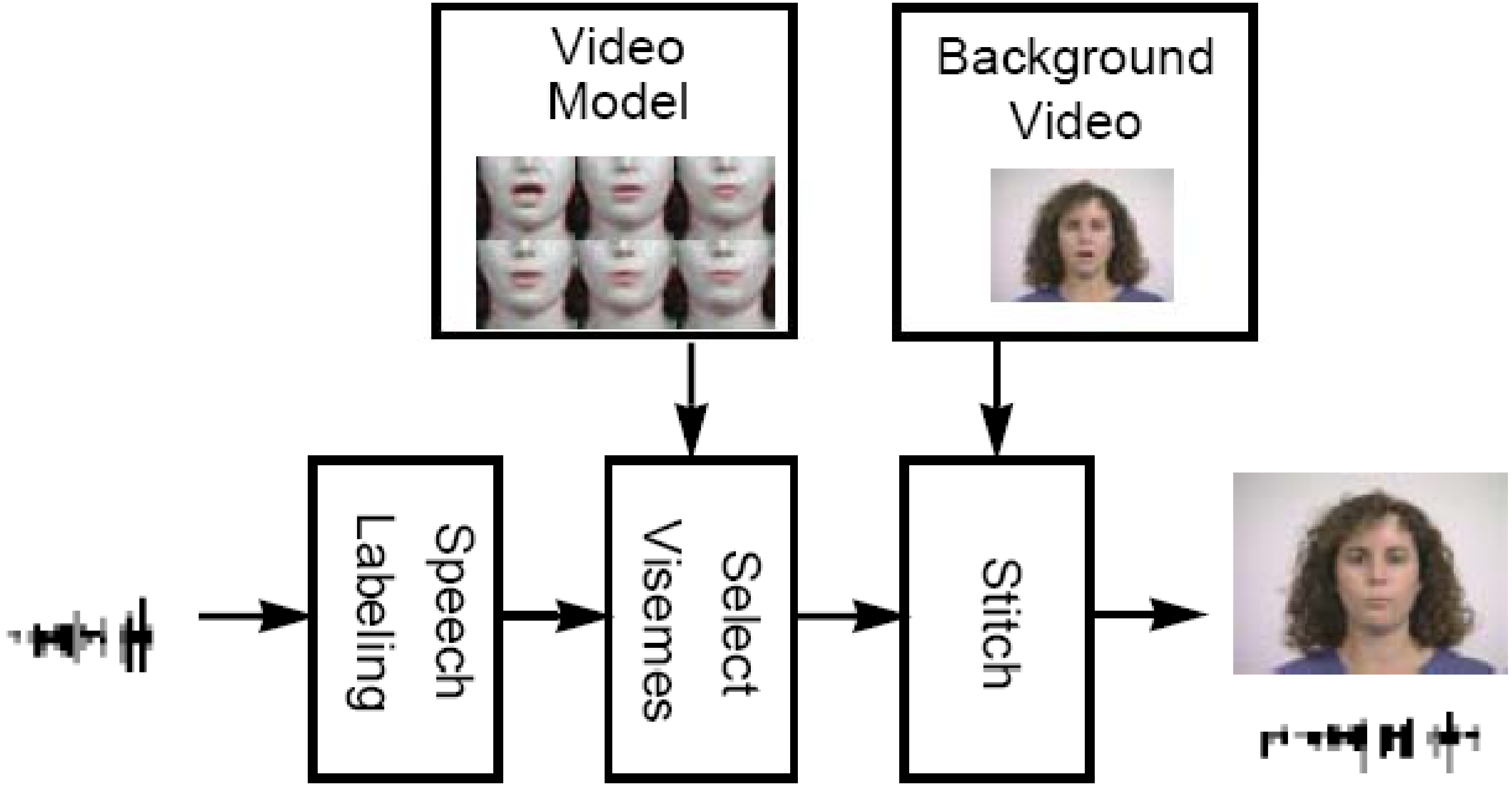


Image-based faces
(lip sync.)

Video rewrite (analysis)



Video rewrite (synthesis)



Results

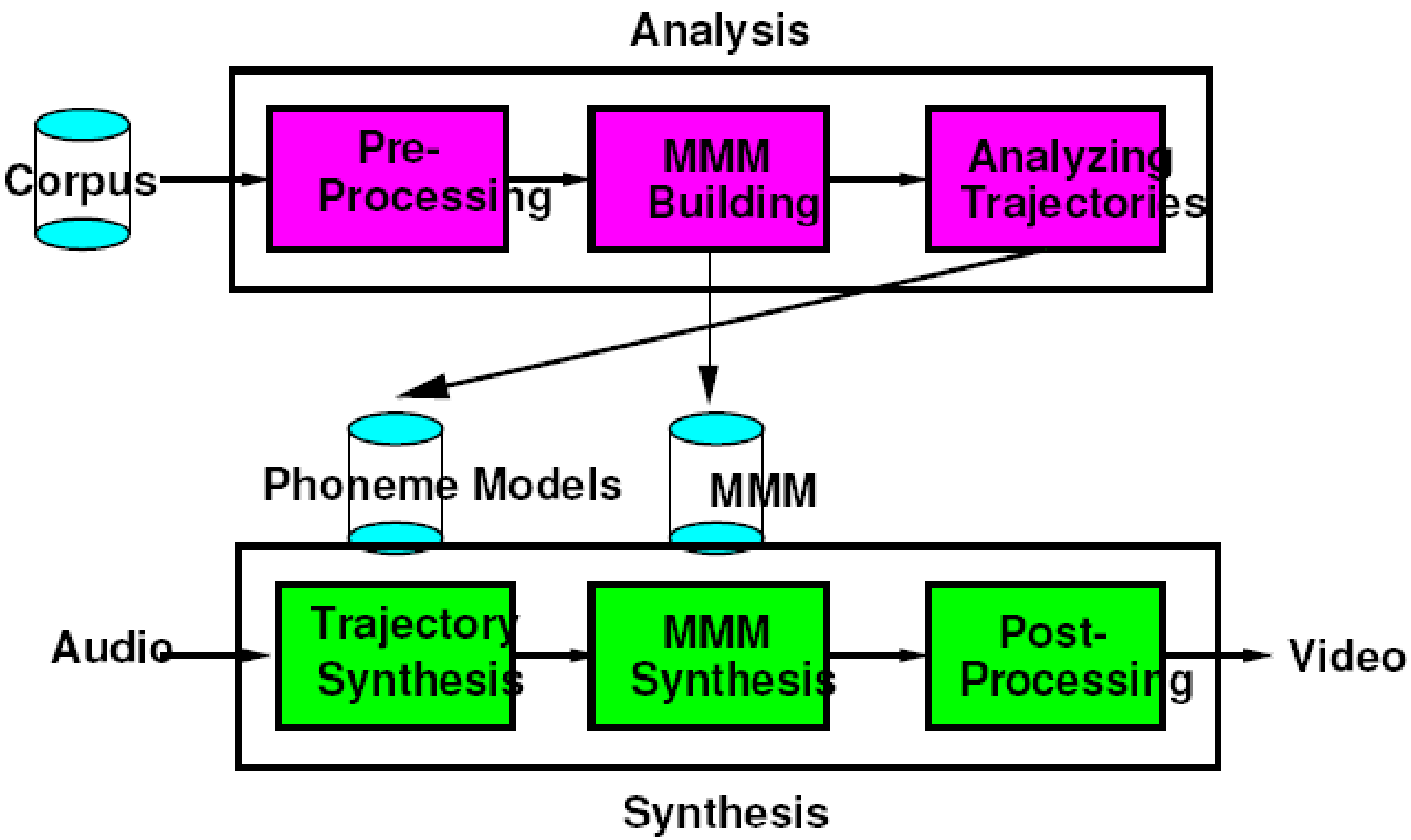
- Video database
  - 2 minutes of JFK
    - Only half usable
    - Head rotation

training video

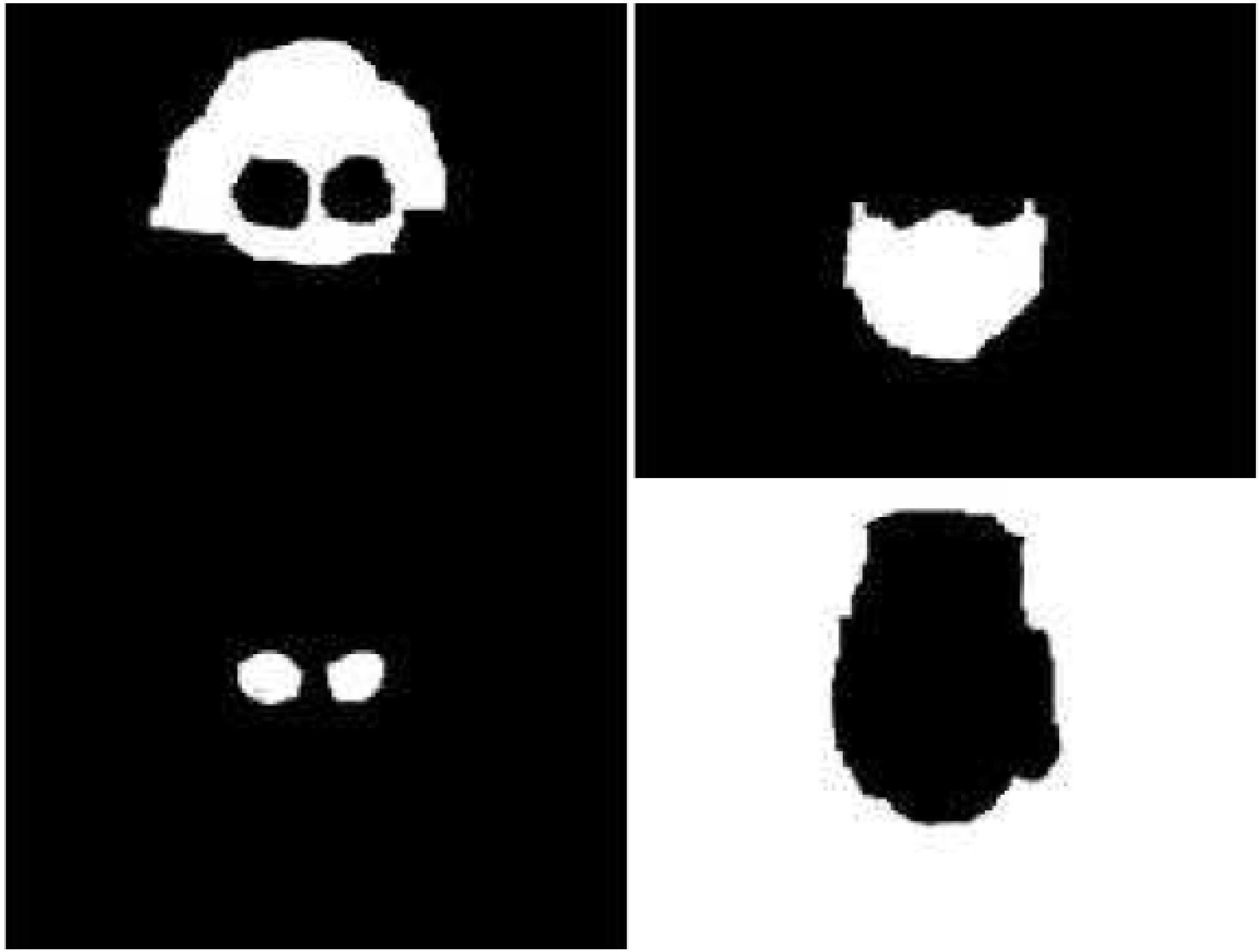
Read my lips.

I never met Forest Gump.

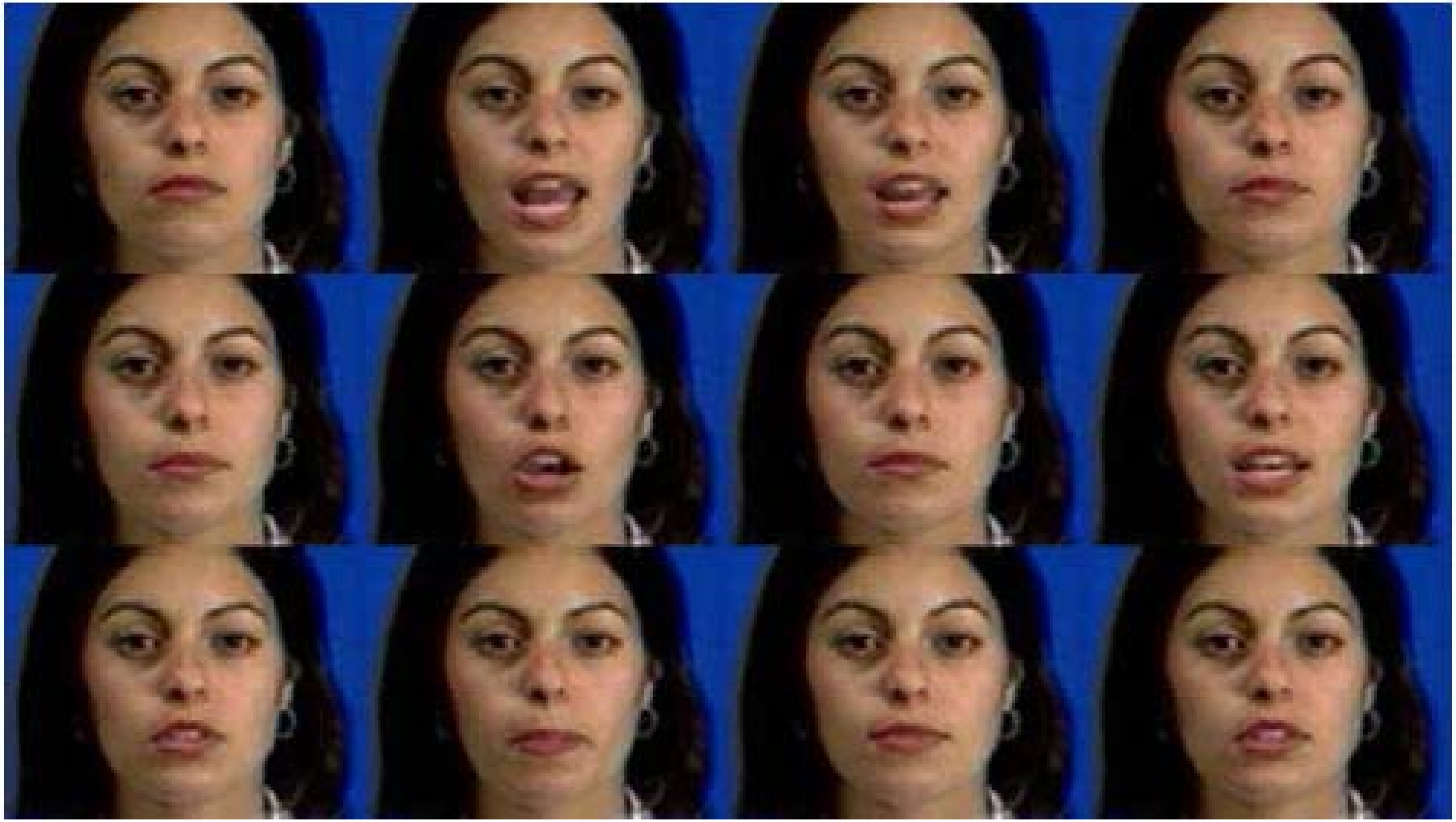
Morphable speech model



Preprocessing



Prototypes (PCA+k-mean clustering)



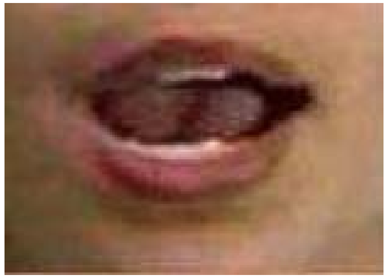
We find I_i and C_i for each prototype image.

Morphable model

$$I^{morph}(\alpha,\beta)=\sum_{i=1}^{N}\beta_i \mathbf{W}(I_i,\mathbf{W}(\sum_{j=1}^{N}\alpha_j C_j - C_i, C_i))$$

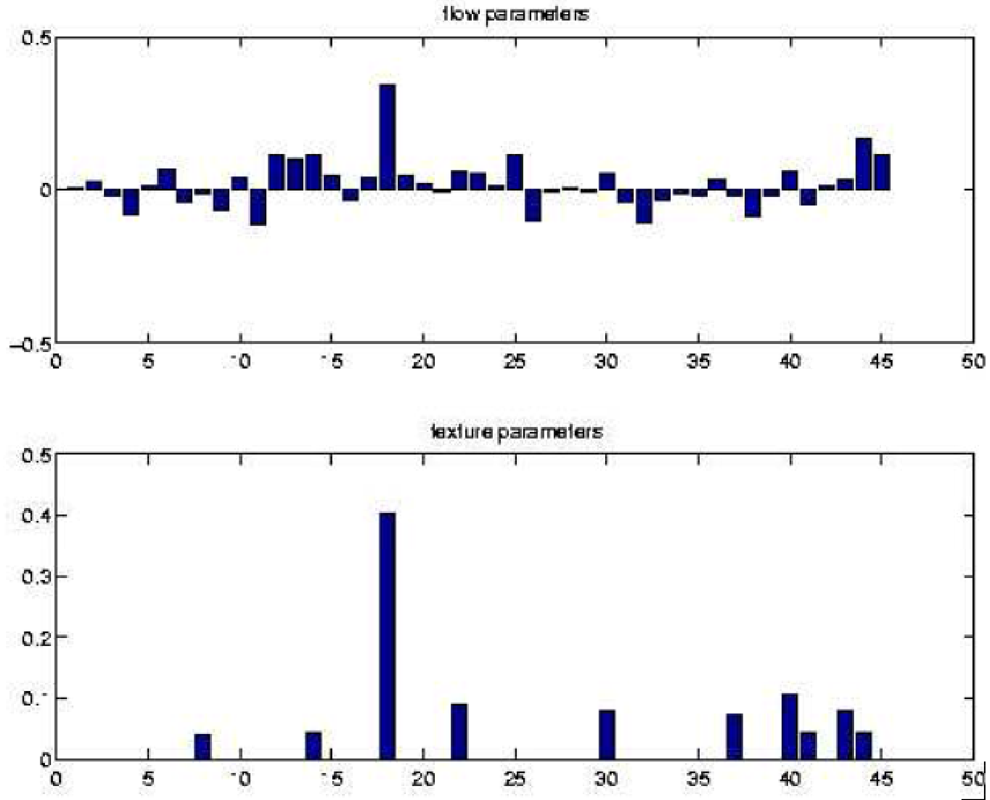
$$\mathbf{I} \underset{\text{synthesis}}{\overset{\text{analysis}}{\rightleftarrows}} \boldsymbol{\alpha}\ \boldsymbol{\beta}$$

Morphable model



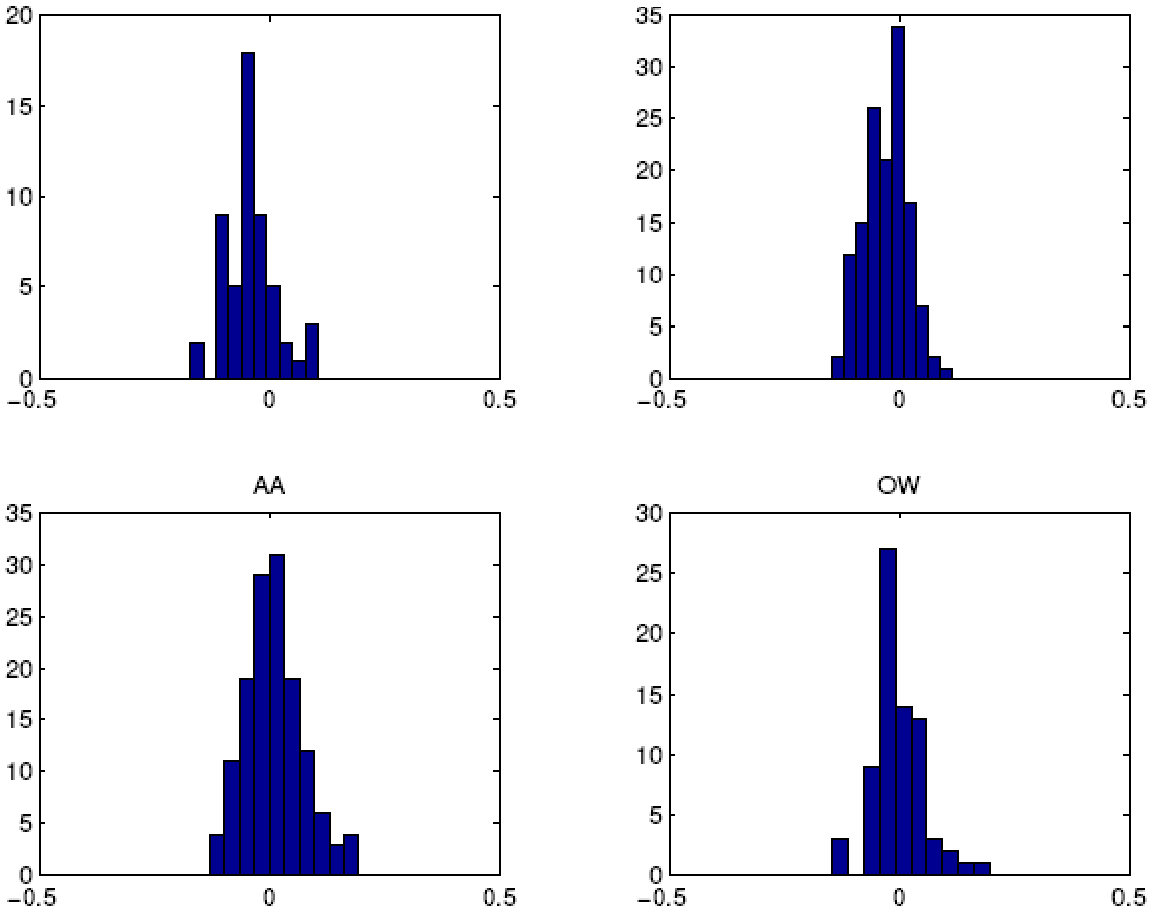
analysis

synthesis



Synthesis

$$E = \underbrace{(y-\mu)^T D^T \Sigma^{-1} D(y-\mu)}_{target\ term} + \lambda \underbrace{y^T W^T W y}_{smoothness}$$

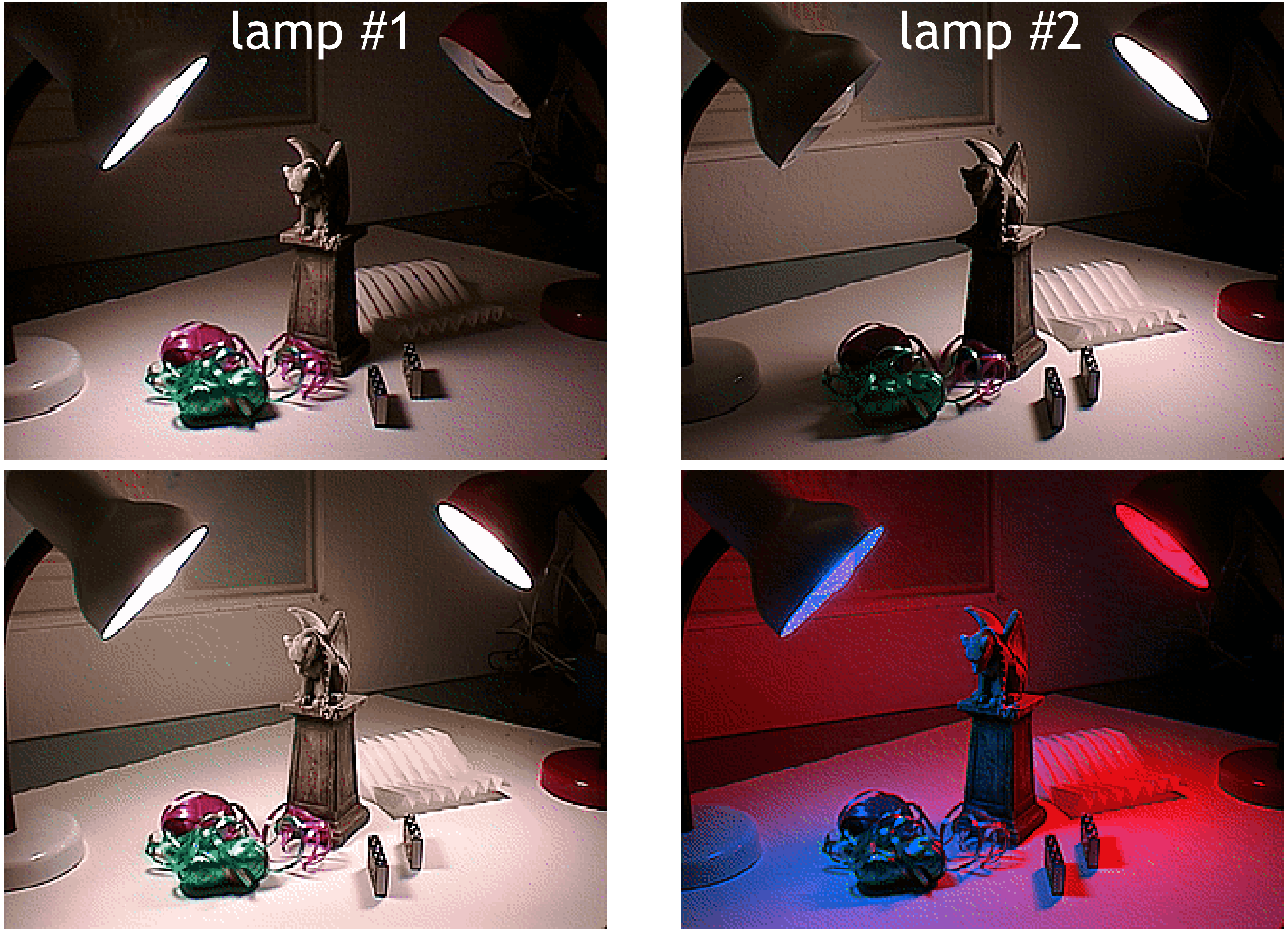


Results

Results

Relighting faces

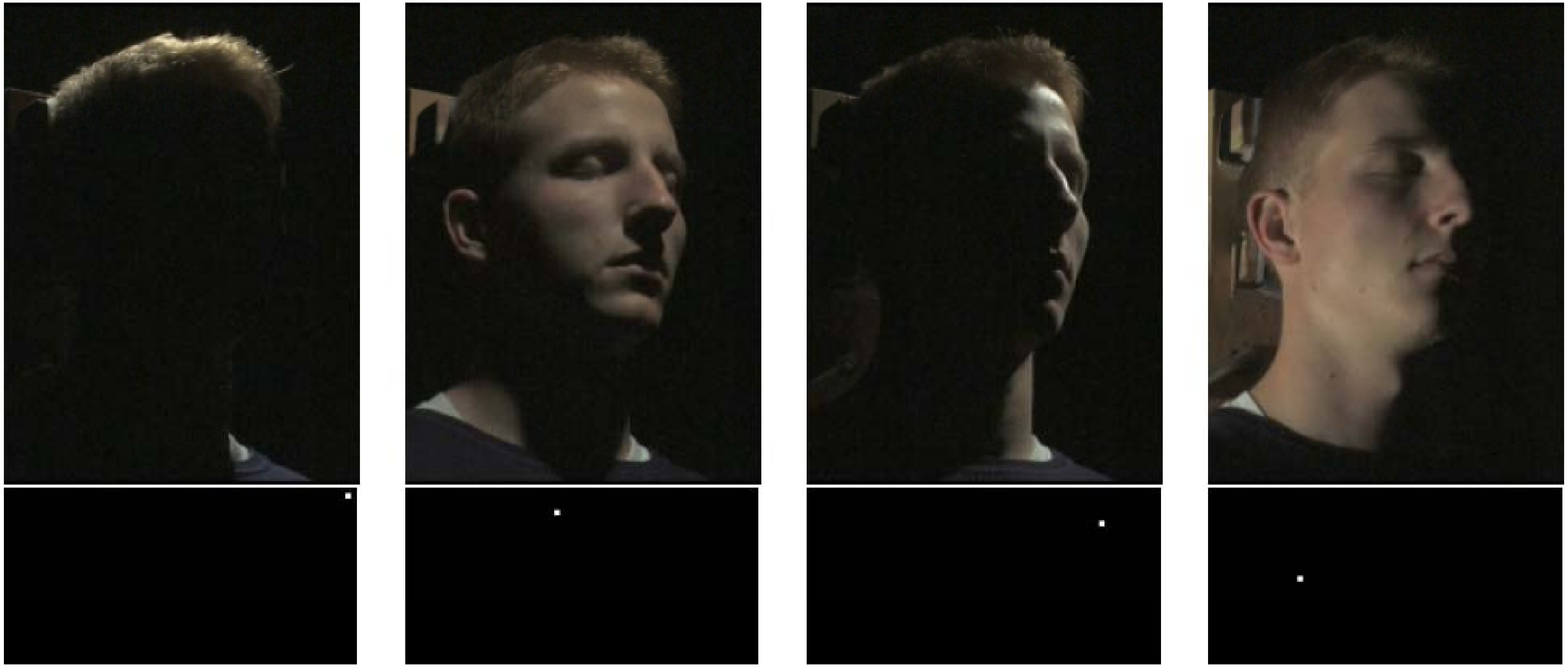
Light is additive



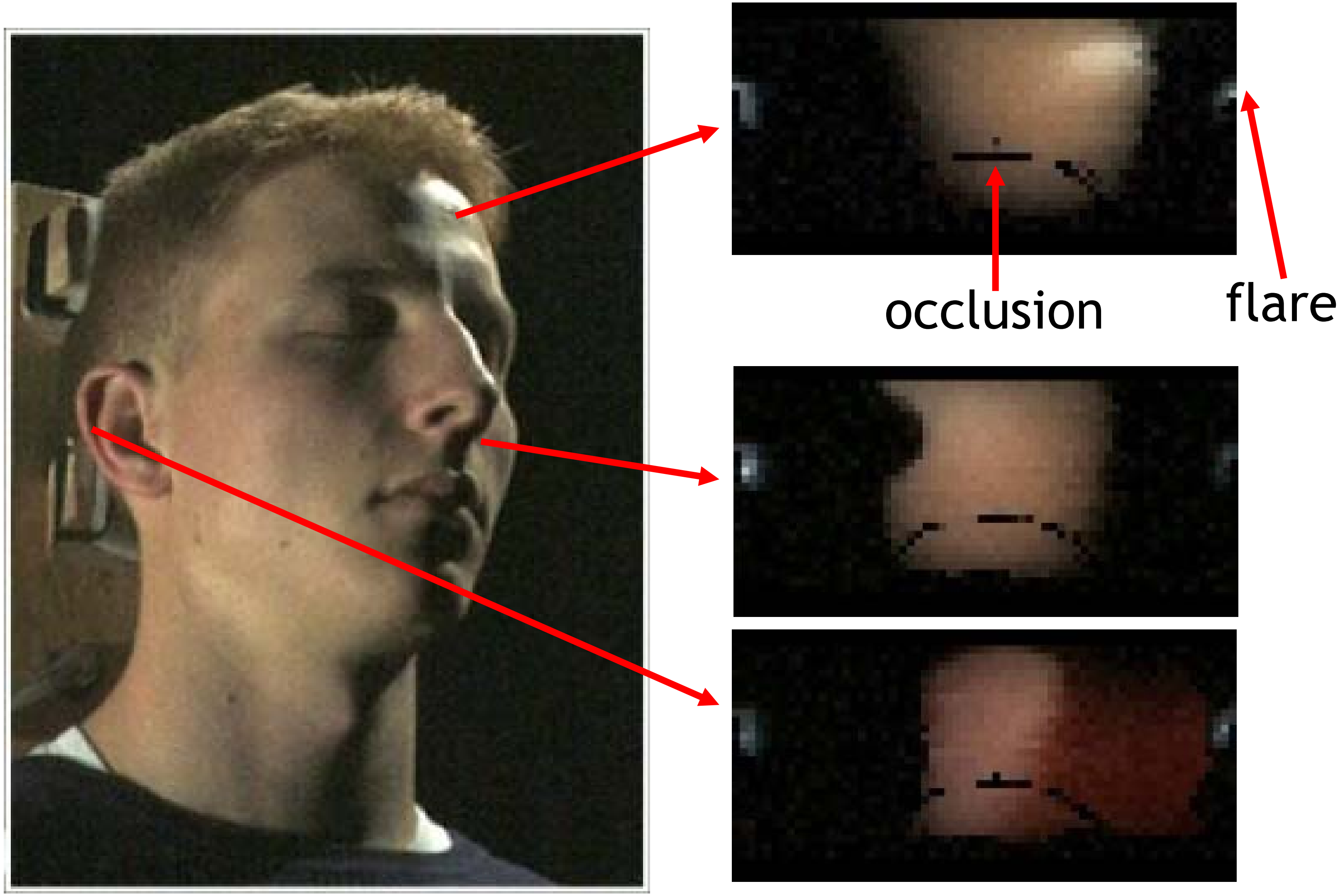
Light stage 1.0

Light stage 1.0

Input images



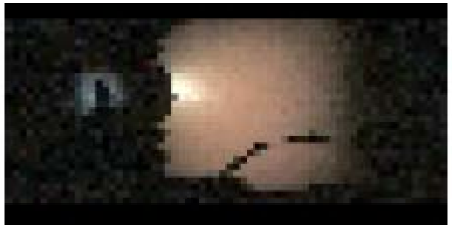
Reflectance function



Relighting

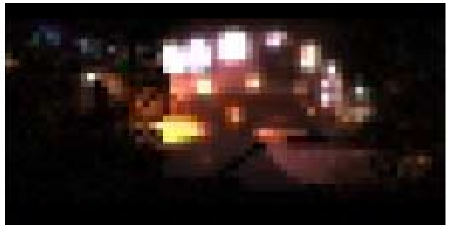
normalized light map

$\times$

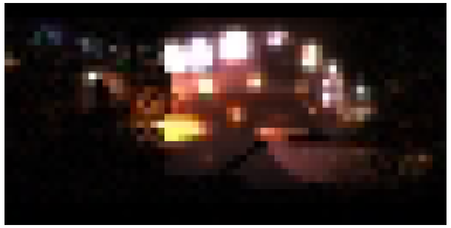


reflectance function

$=$

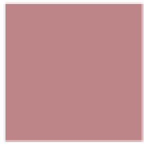


lighting product

$\|$  $\|_1$

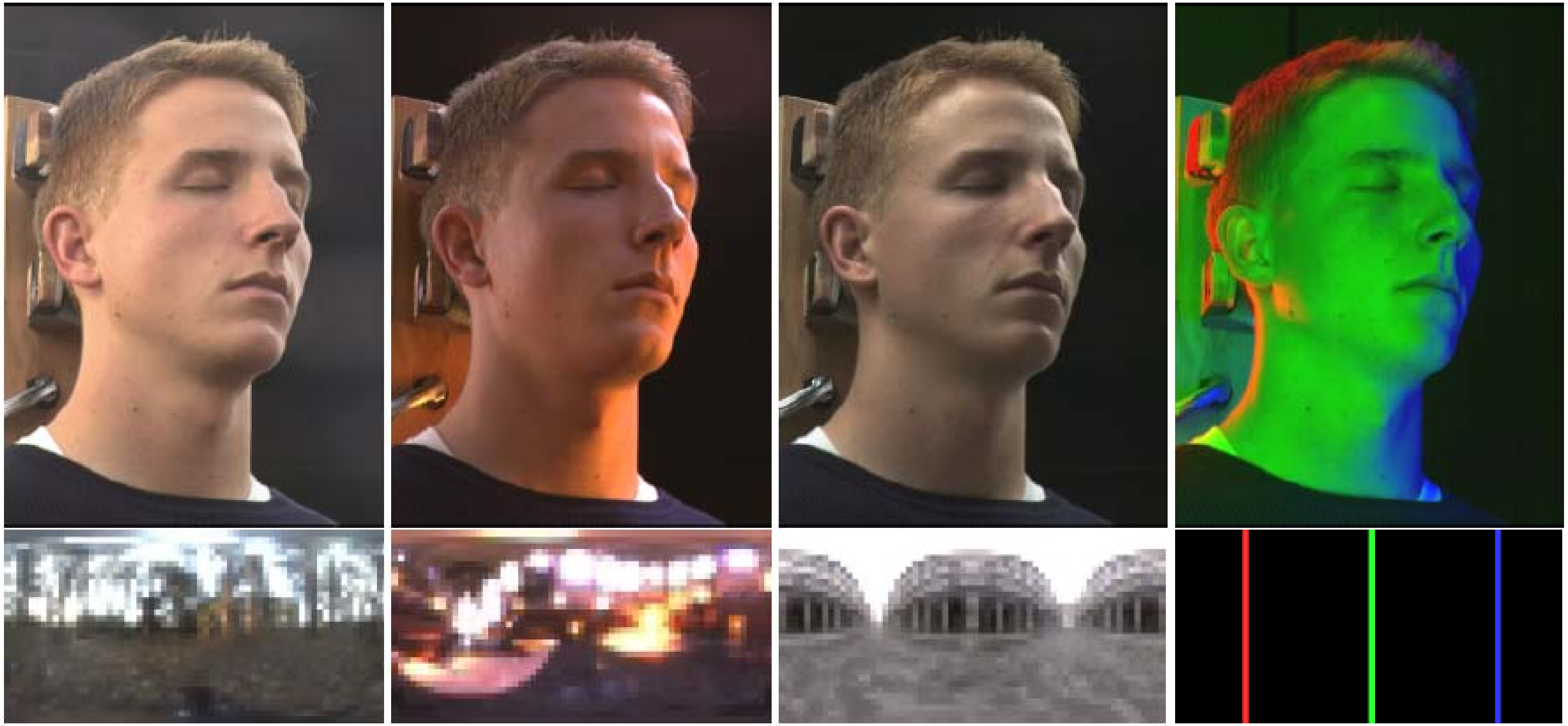
lighting product

$=$

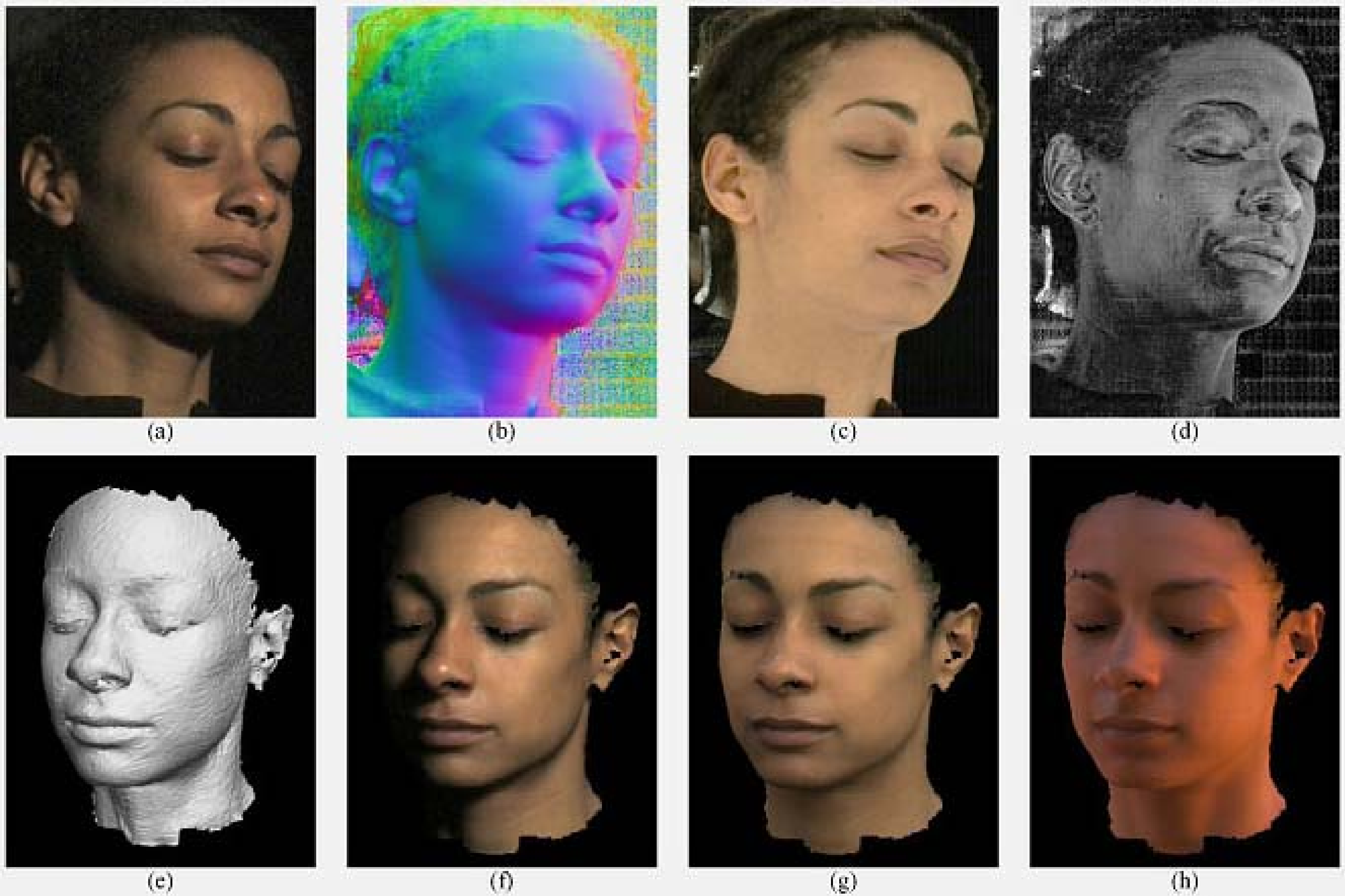


rendered pixel

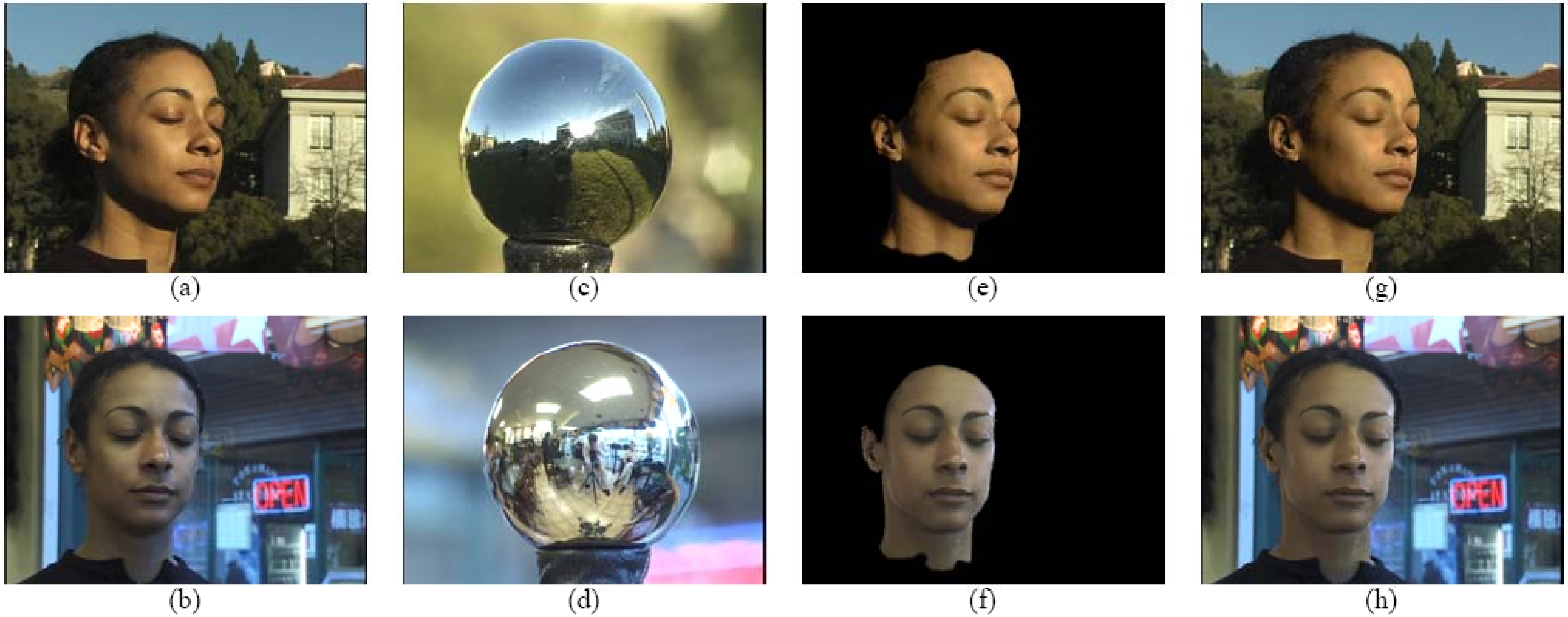
Results



Changing viewpoints

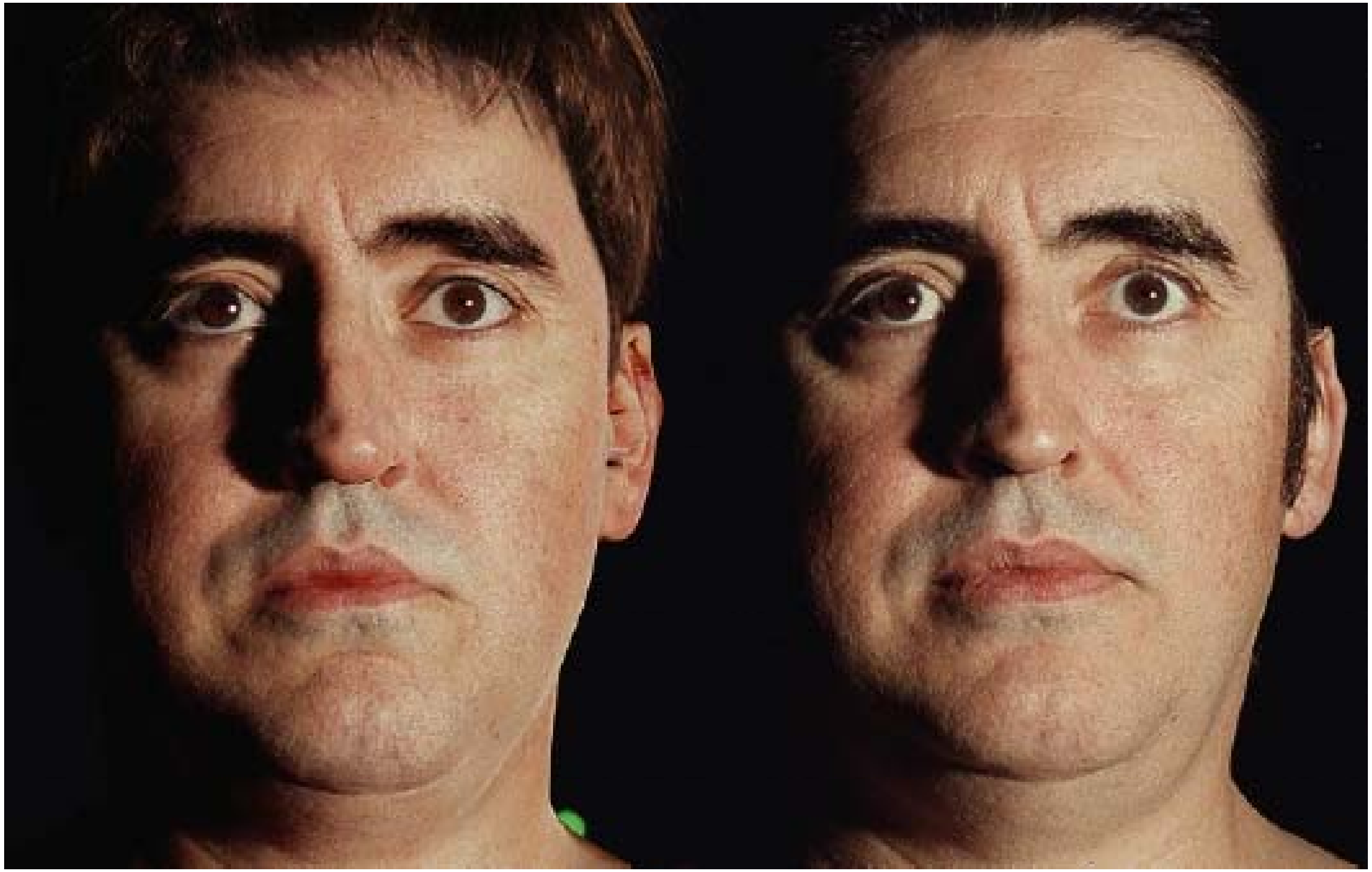


Results



3D face applications: Spiderman 2

Spiderman 2

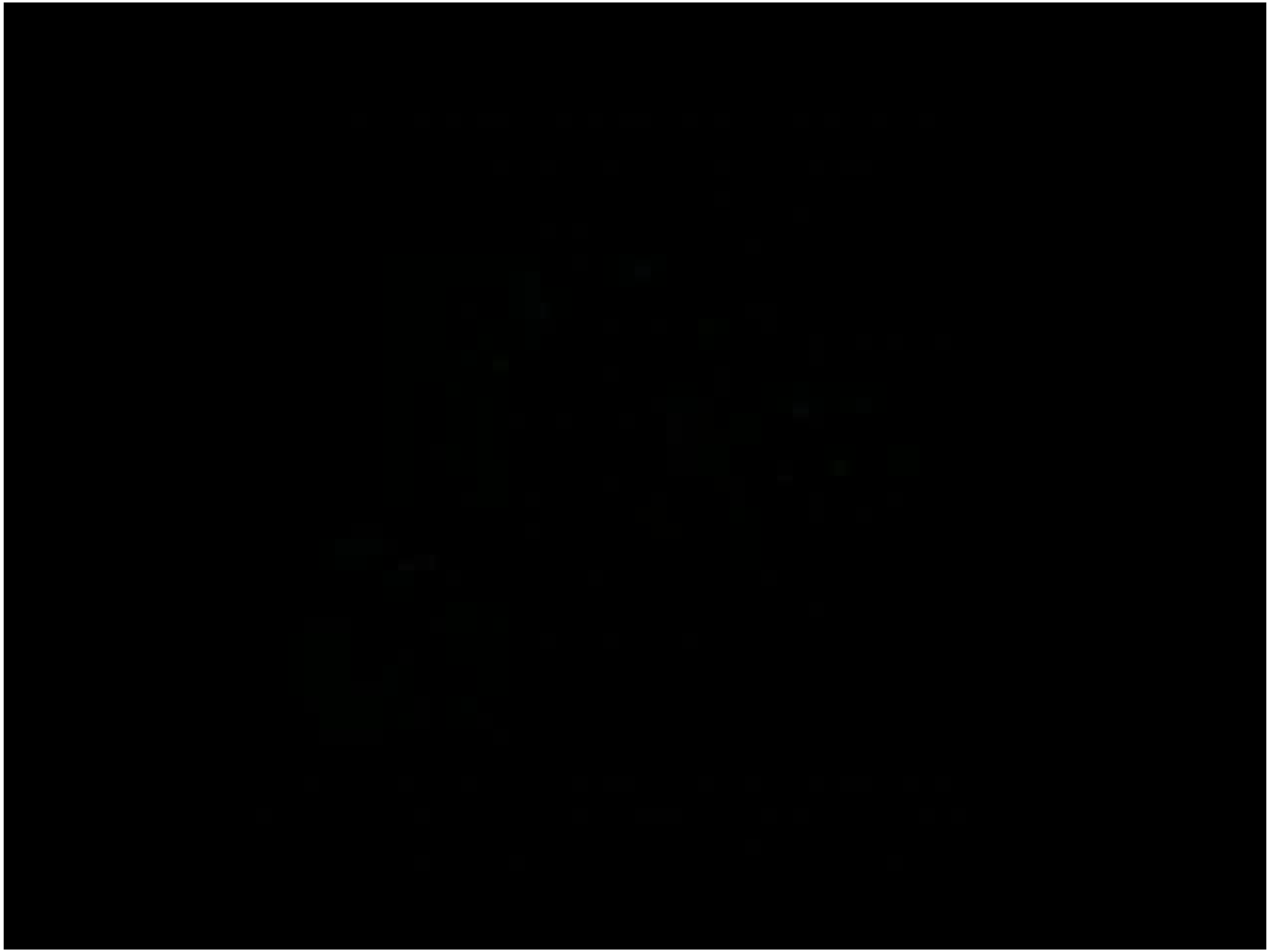


real synthetic

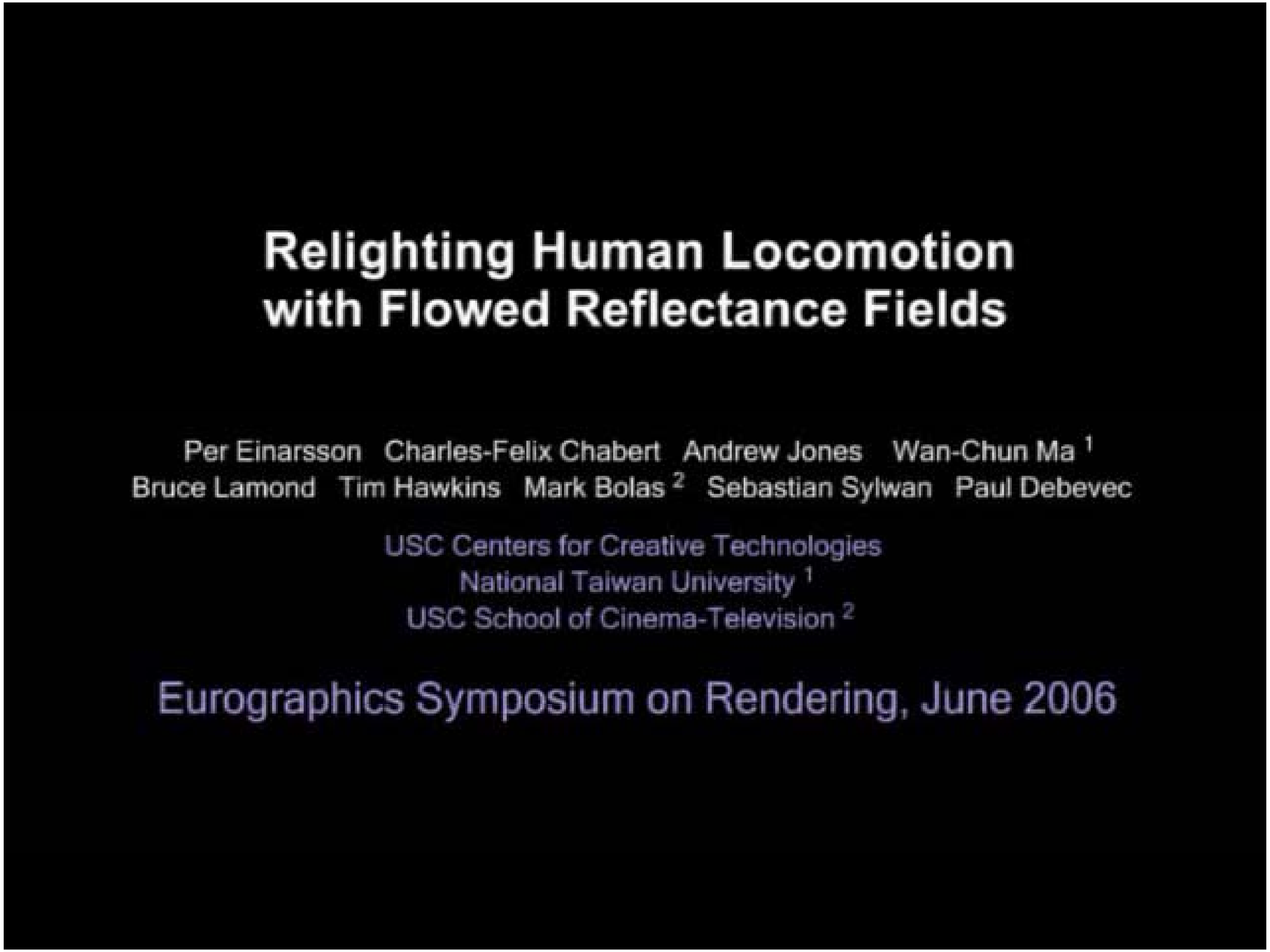
Spiderman 2

video

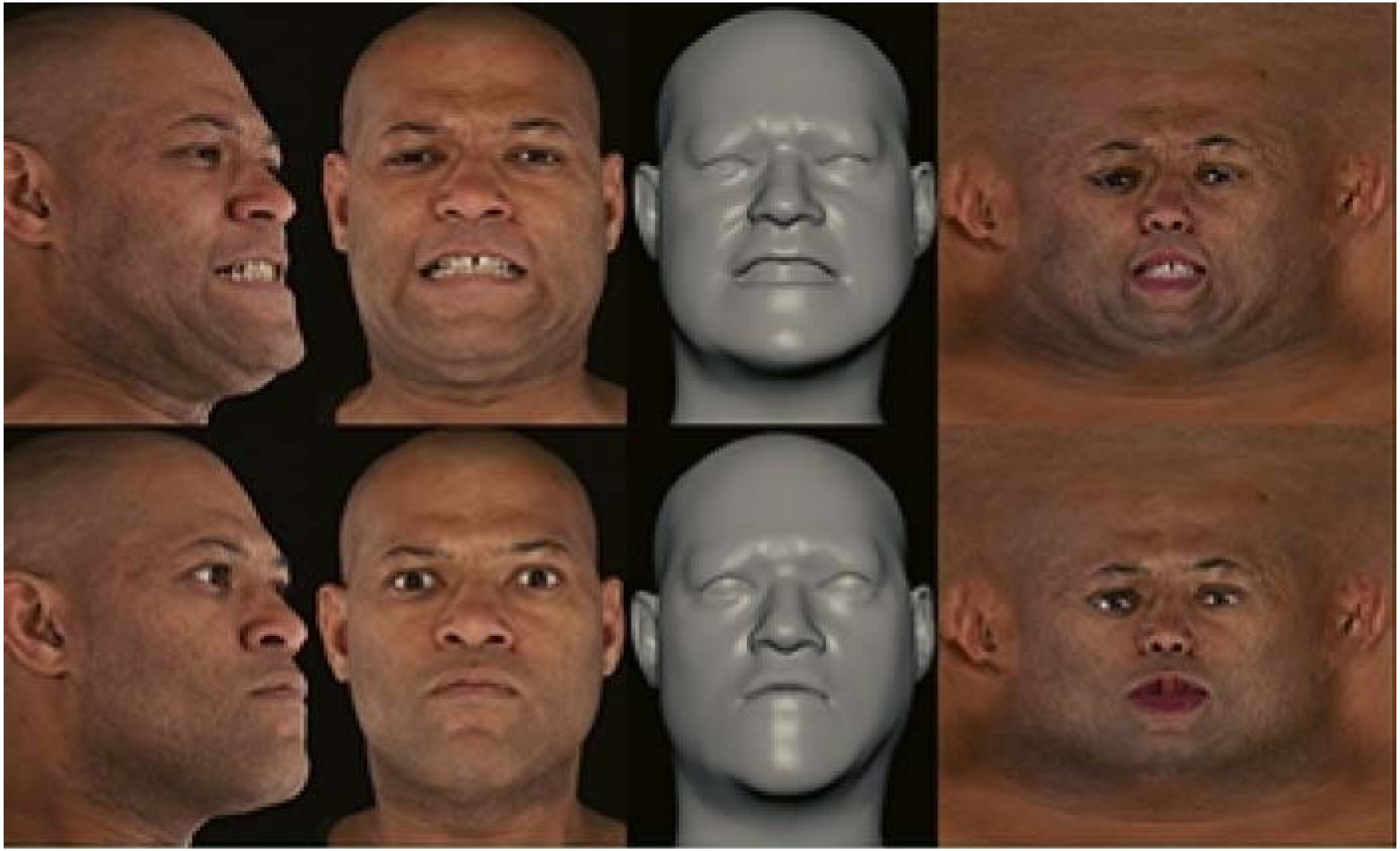
Light stage 3



Light stage 6



Application: The Matrix Reloaded



Application: The Matrix Reloaded

References

- Paul Debevec, Rendering Synthetic Objects into Real Scenes: Bridging Traditional and Image-based Graphics with Global Illumination and High Dynamic Range Photography, SIGGRAPH 1998.
- F. Pighin, J. Hecker, D. Lischinski, D. H. Salesin, and R. Szeliski. Synthesizing realistic facial expressions from photographs. SIGGRAPH 1998, pp75-84.
- Li Zhang, Noah Snavely, Brian Curless, Steven M. Seitz, Spacetime Faces: High Resolution Capture for Modeling and Animation, SIGGRAPH 2004.
- Blanz, V. and Vetter, T., A Morphable Model for the Synthesis of 3D Faces, SIGGRAPH 1999, pp187-194.
- Paul Debevec, Tim Hawkins, Chris Tchou, Haarm-Pieter Duiker, Westley Sarokin, Mark Sagar, Acquiring the Reflectance Field of a Human Face, SIGGRAPH 2000.
- Christoph Bregler, Malcolm Slaney, Michele Covell, Video Rewrite: Driving Visual Speeach with Audio, SIGGRAPH 1997.
- Tony Ezzat, Gadi Geiger, Tomaso Poggio, Trainable Videorealistic Speech Animation, SIGGRAPH 2002.