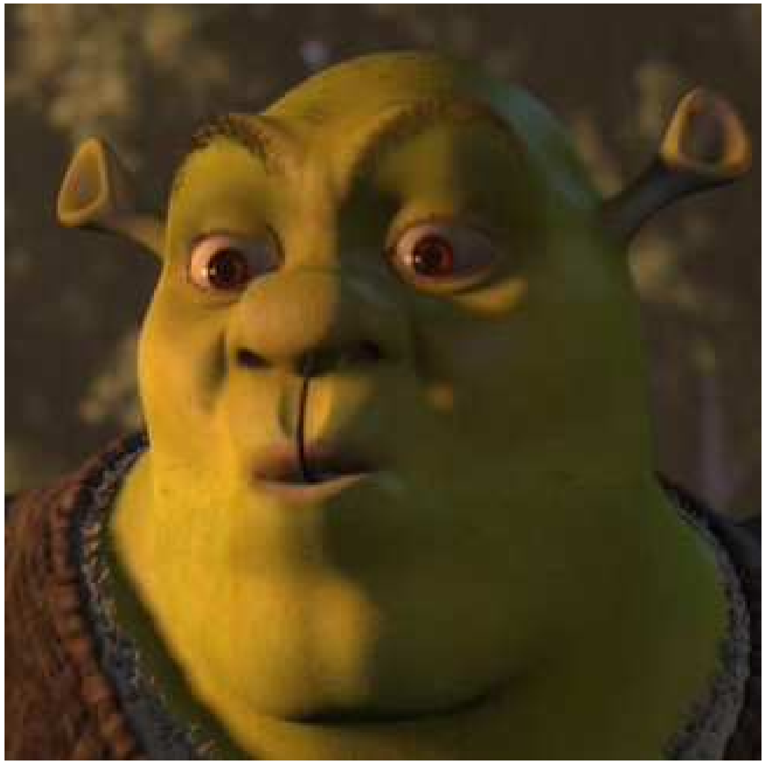
Facial Animation

Roman Berka
http://vyuka.iim.cz/a4m39mma:a4m39mma

1. History and application
2. Briefly about the anatomy
3. Modeling methods overview
4. Animation techniques
5. Parametrization

Motivation

1 2
3 4
5 6
7 8
9 10
11 12
13 14
15 16
17 18
19 20
21 22
23 24
25 26
27 28
29 30



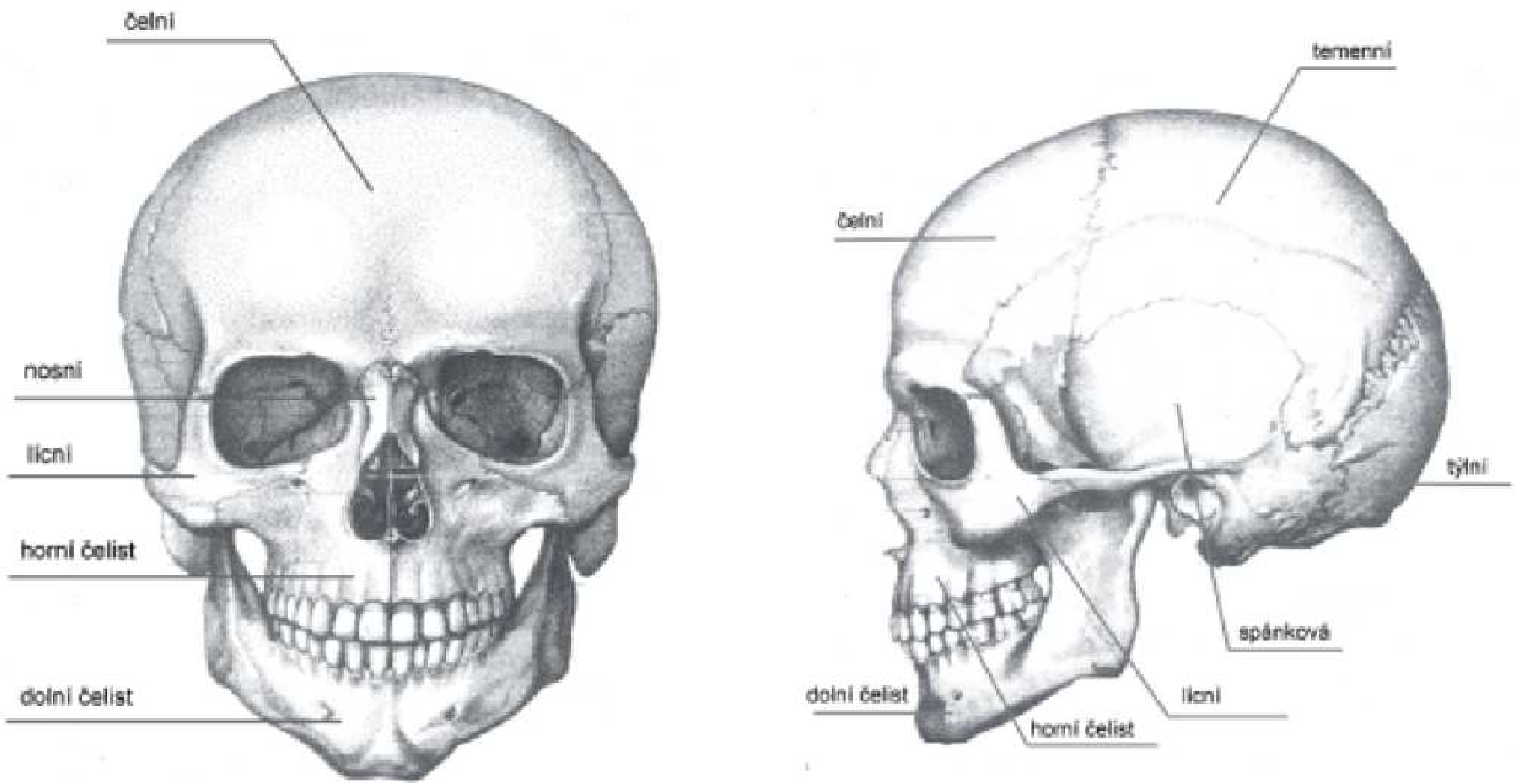
History and Application

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30

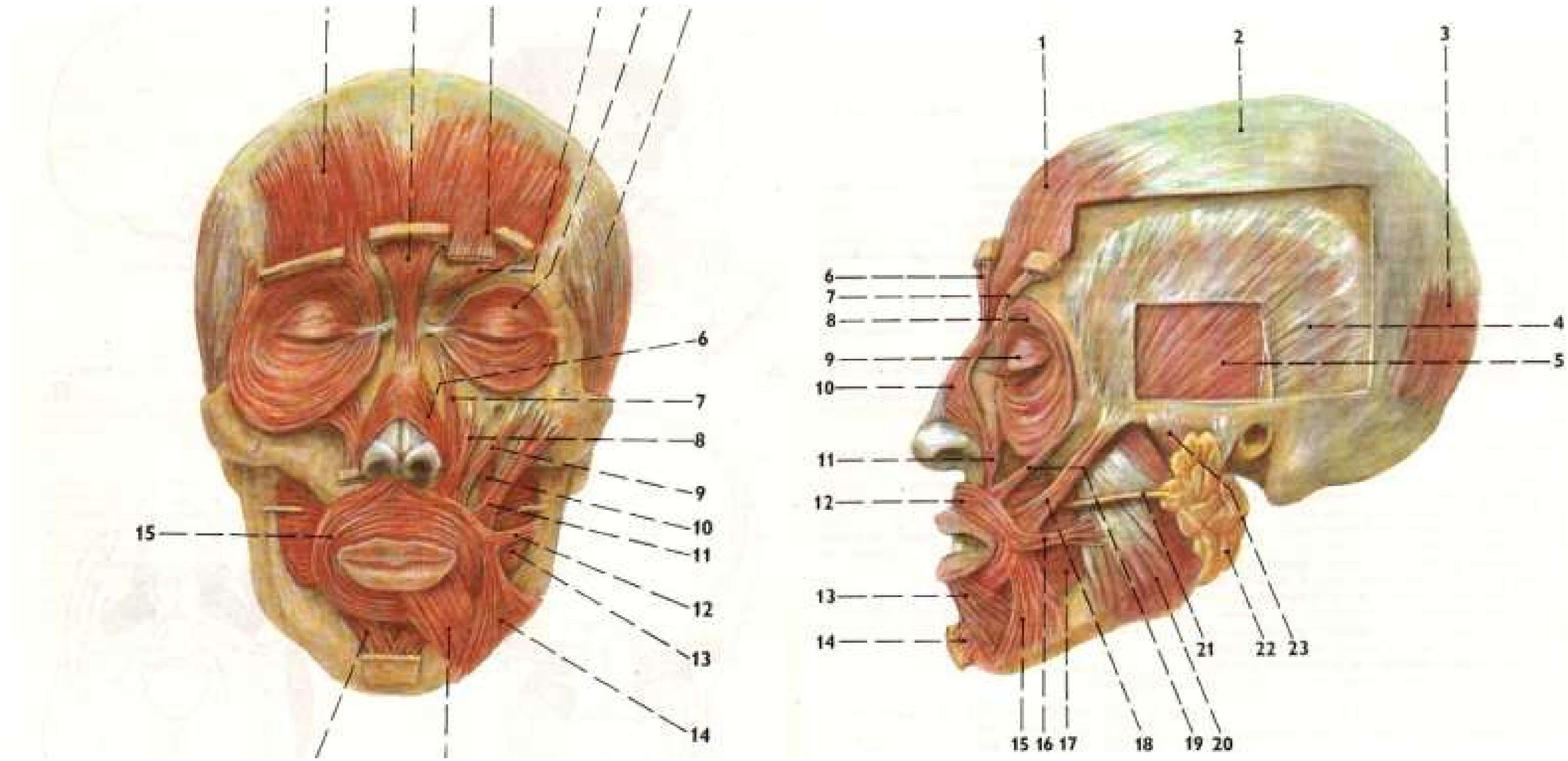
- Parke 1972 – polygonal representation (Gouraud)
- Parke 1974 – parametrized model
- 1974-78 New York Institute of Technology, Cornell University cartoons
- 1980 – Platt - Master thesis muscle controlled FA
- 1982 – MIT – Weil videodisk-based system
- 1987 – Waters - muscle model
- 1988 – N.M.Thalman - abstract muscle model – Levis - speach synchronisation
- 1990 – Optical scanners
- 1990 – image registered on facial model
- 1993 – individual face mapping

- Games
- Medicine
  - craniofacial surgical planning
  - facial tissue simulation
- Teleconferencing
- Social agents, virtual reality

www.cybertown.com

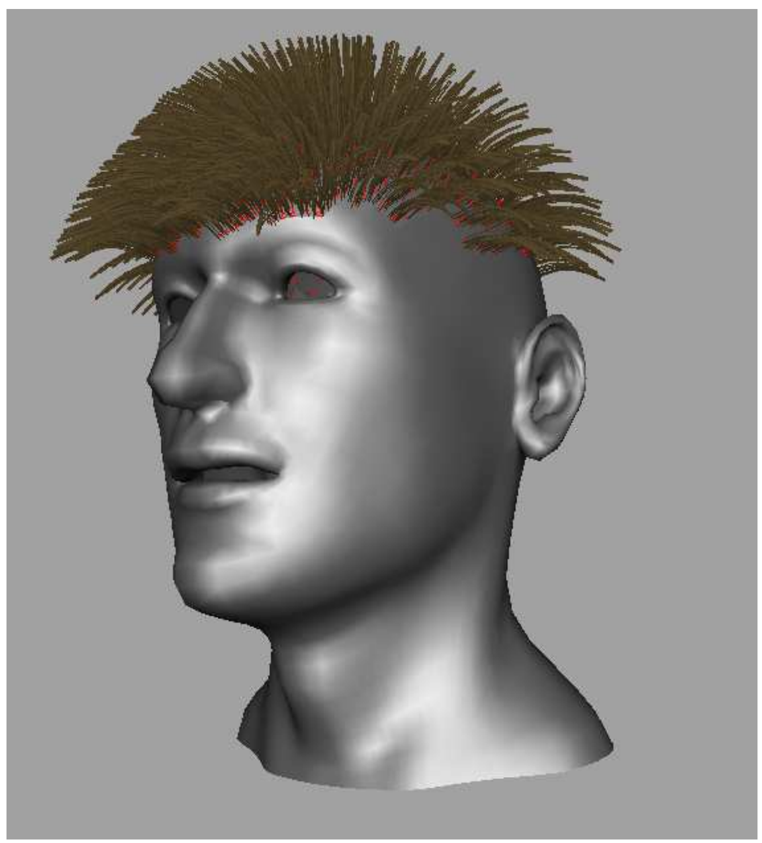

čelní
nosní
lícní
horní čelist
dolní čelist
temenní
čelní
týlní
spánková
lícní
dolní čelist
horní čelist

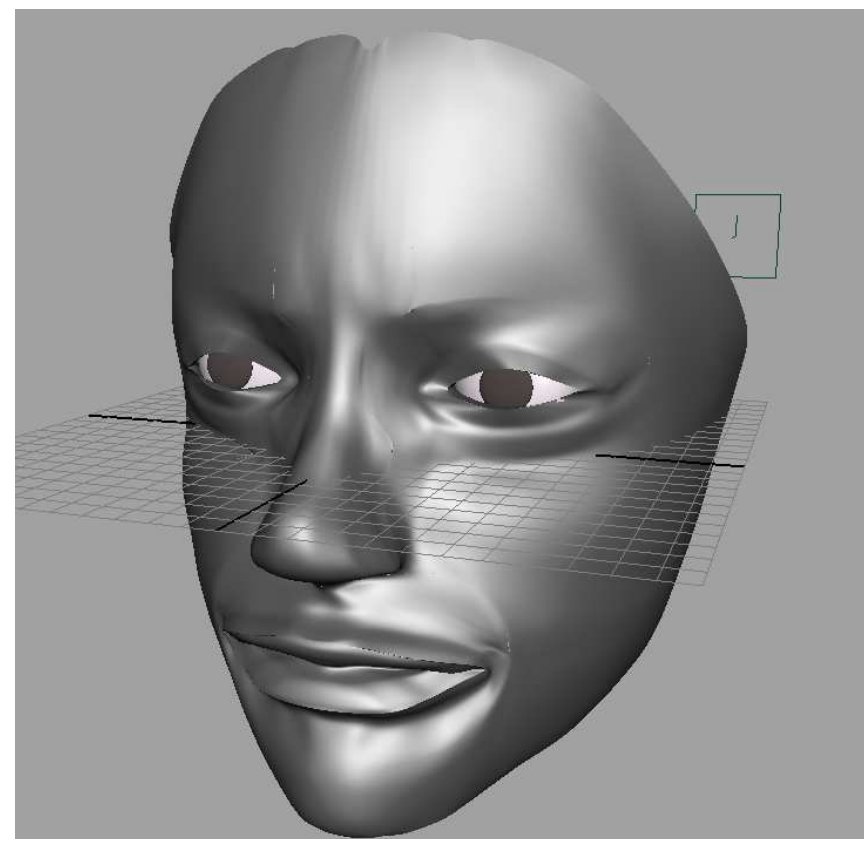
pay attention to the structure, names are not important for know



Topology of the Face

- facial mask
- eyes
- lips
- teeth
- tongue
- ears





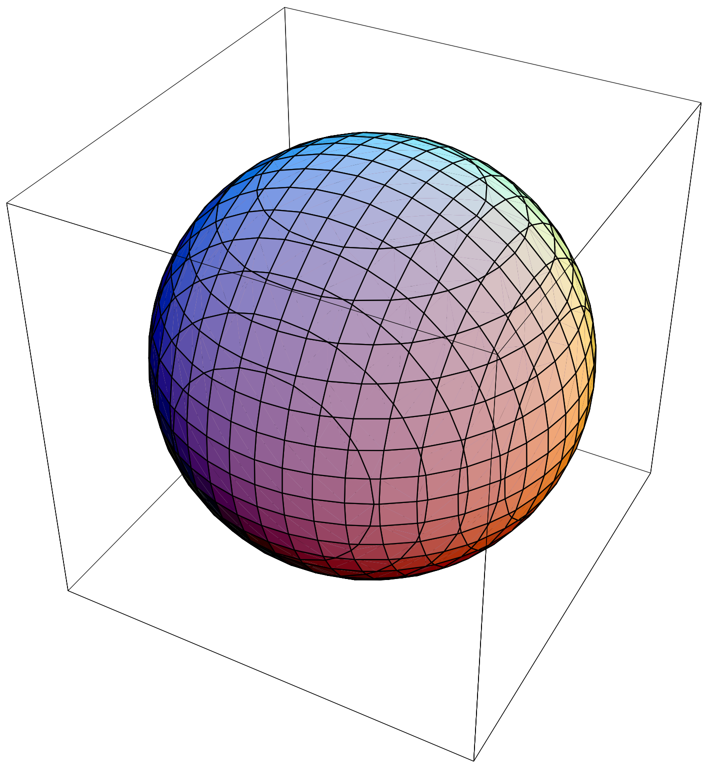
Possible Represenations

- Volume representations
  - CSG
  - Voxel-based rep.
- Surface rep.
  - implicit surfaces
  - parametric surfaces (Non Uniform Rational B-Spline Surfaces, Hierarchical B-Splines)
  - Polygonal Surfaces

Implicit surfaces

$$f(x, y, z) = 0$$

$$f(x, y, z) = x^2 + (y - 0.5)^2 + (z - 0.5)^2 - 1$$



NURBS

The NURBS surface is determined by set of control vertices P, weights w of the vertices, normalised B-spline base functions $N_{i,p}(u)$ and $N_{j,q}(u)$, and knot vectors U and V.

$$Q(u,v) = \frac{\sum_{i=0}^{n}\sum_{j=0}^{m} w_{i,j} P_{i,j} N_{i,p}(u) N_{j,q}(v)}{\sum_{i=0}^{n}\sum_{j=0}^{m} w_{i,j} N_{i,p}(u) N_{j,q}(v)}$$

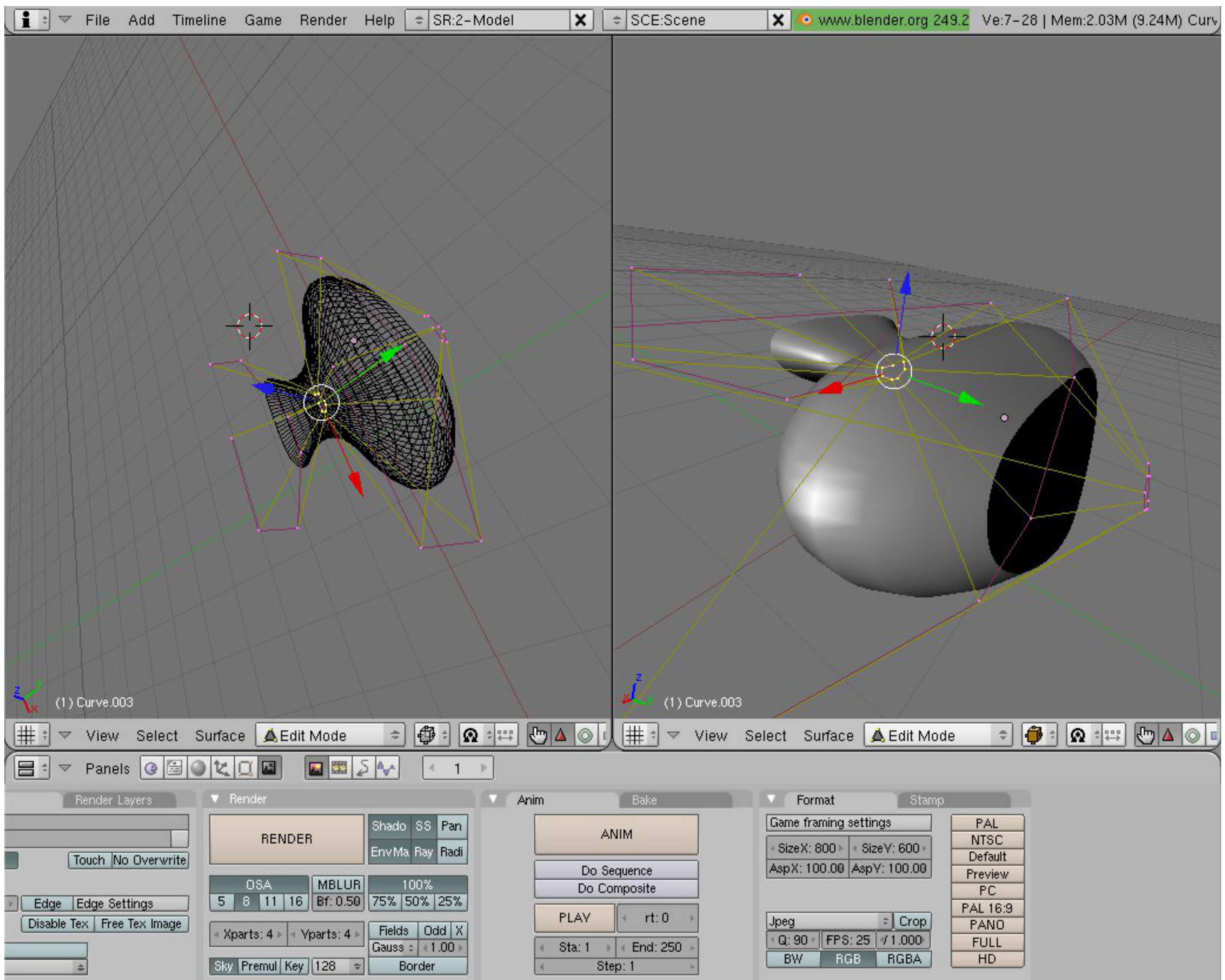
The surface can be expressed as

$$Q(u,v) = \sum_{i=0}^{n}\sum_{j=0}^{m} P_{i,j} R_{i,p}(u) R_{j,q}(v)$$

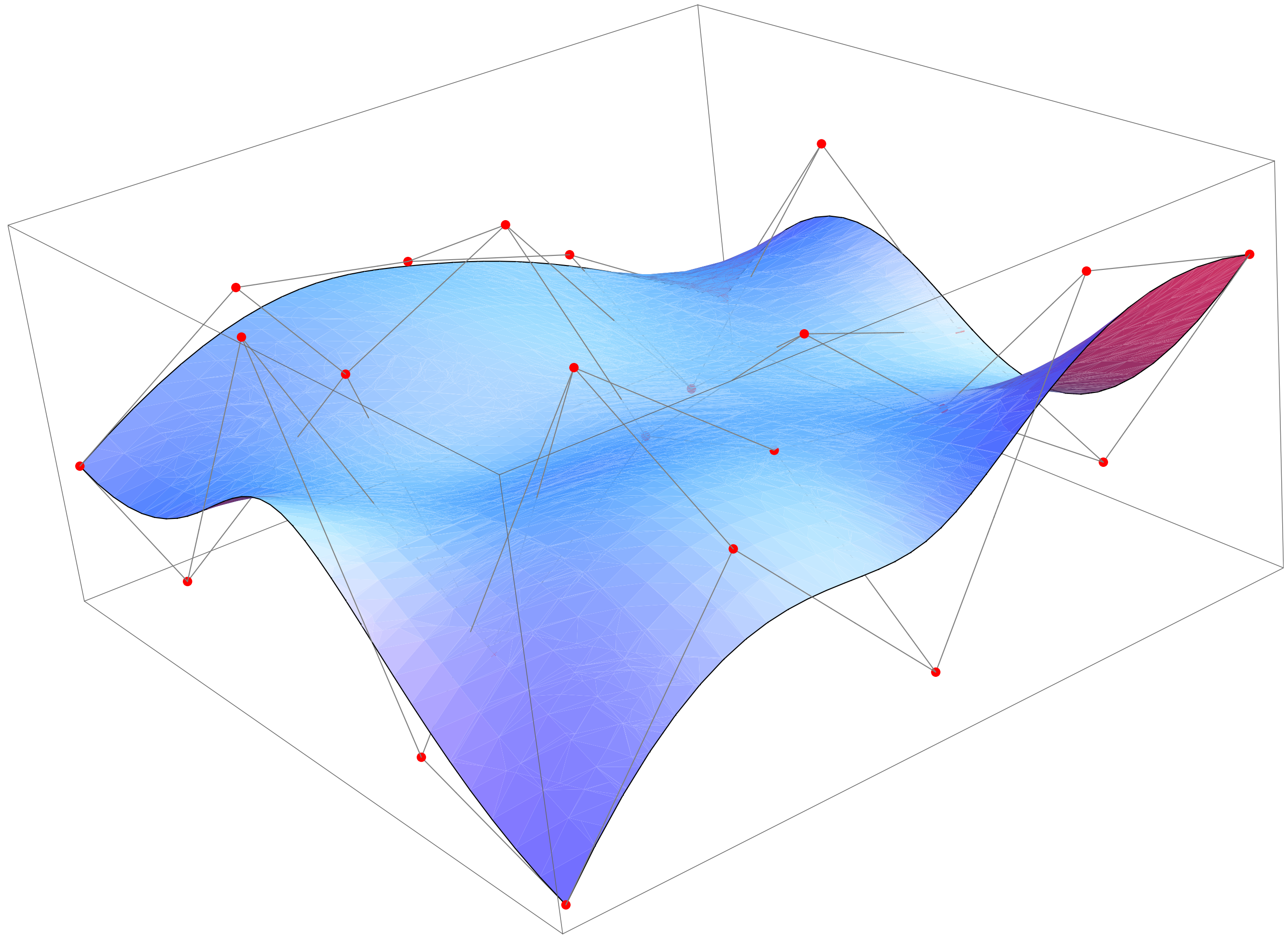
where R is called rational B-Spline base:

$R_{i,p}(t) = \frac{w_i N_{i,p}(t)}{\sum_{j=0}^{n} w_j N_{j,p}(t)}$

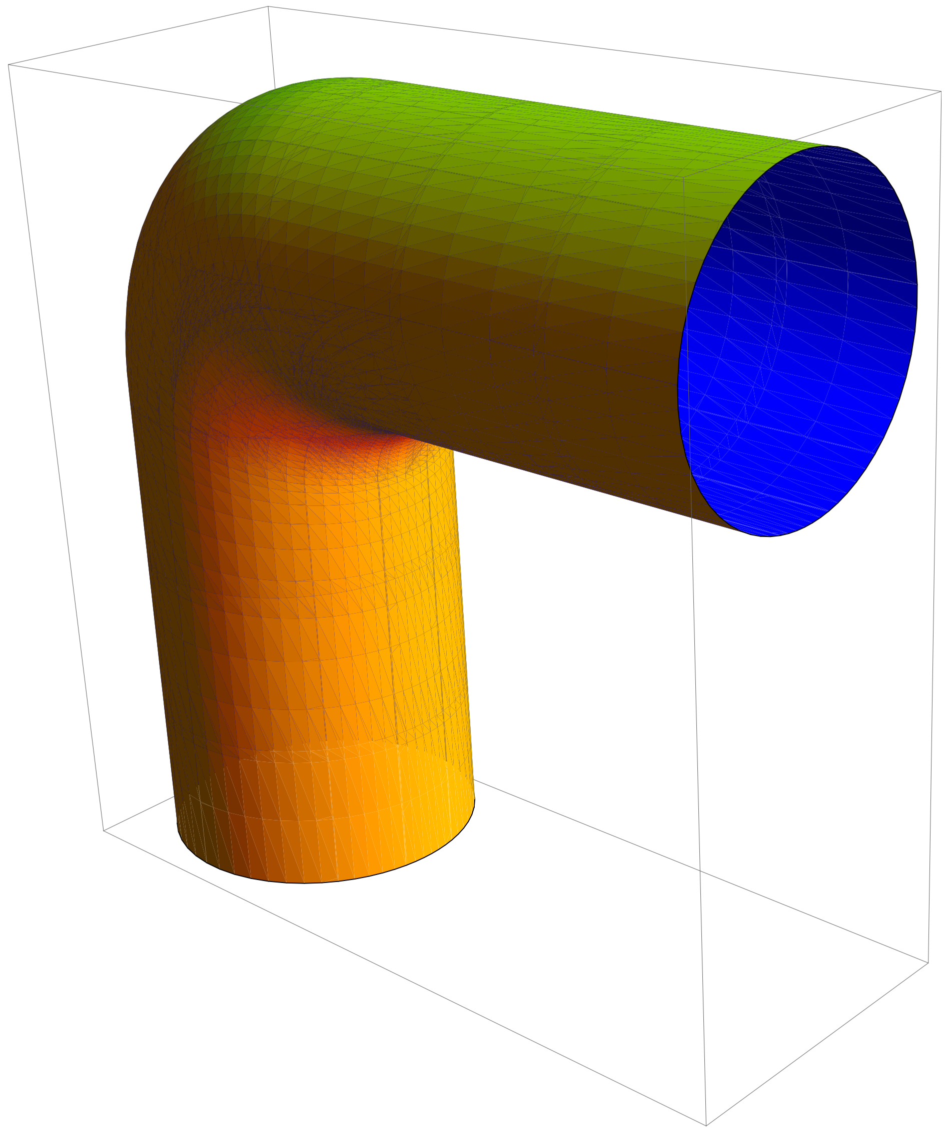
NURBS


File
Add
Timeline
Game
Render
Help
SR:2-Model
SCE:Scene
www.blender.org 249.2
Ve:7-28 | Mem:2.03M (9.24M) Curv
(1) Curve.003
(1) Curve.003
View
Select
Surface
Edit Mode
View
Select
Surface
Edit Mode
Panels
Render Layers
Render
Anim
Bake
Format
Stamp
RENDER
ANIM
Do Sequence
Do Composite
PLAY
Game framing settings
SizeX: 800
SizeY: 600
AspX: 100.00
AspY: 100.00
PAL
NTSC
Default
Preview
PC
PAL 16:9
PANO
FULL
HD

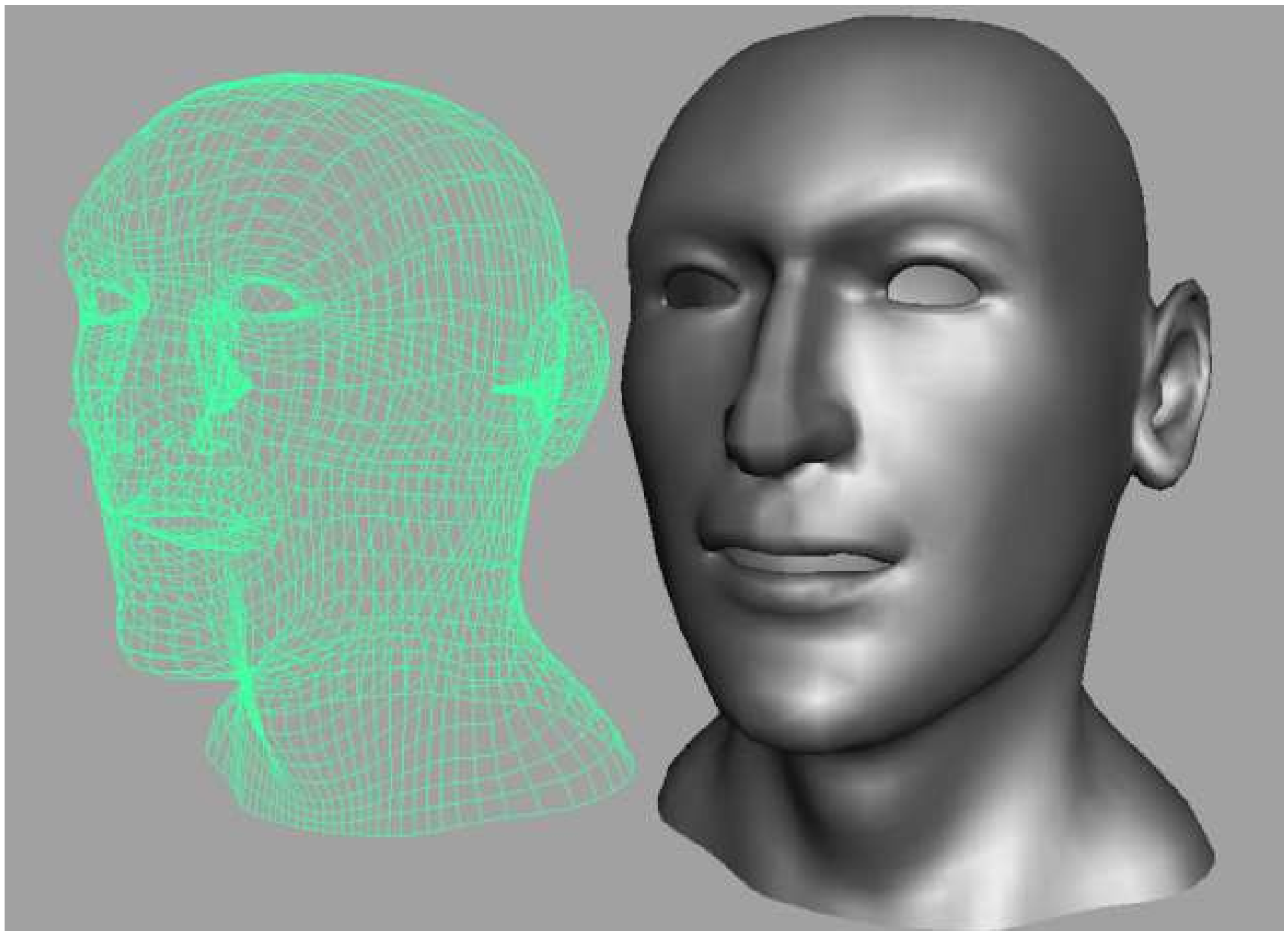
NURBS



NURBS

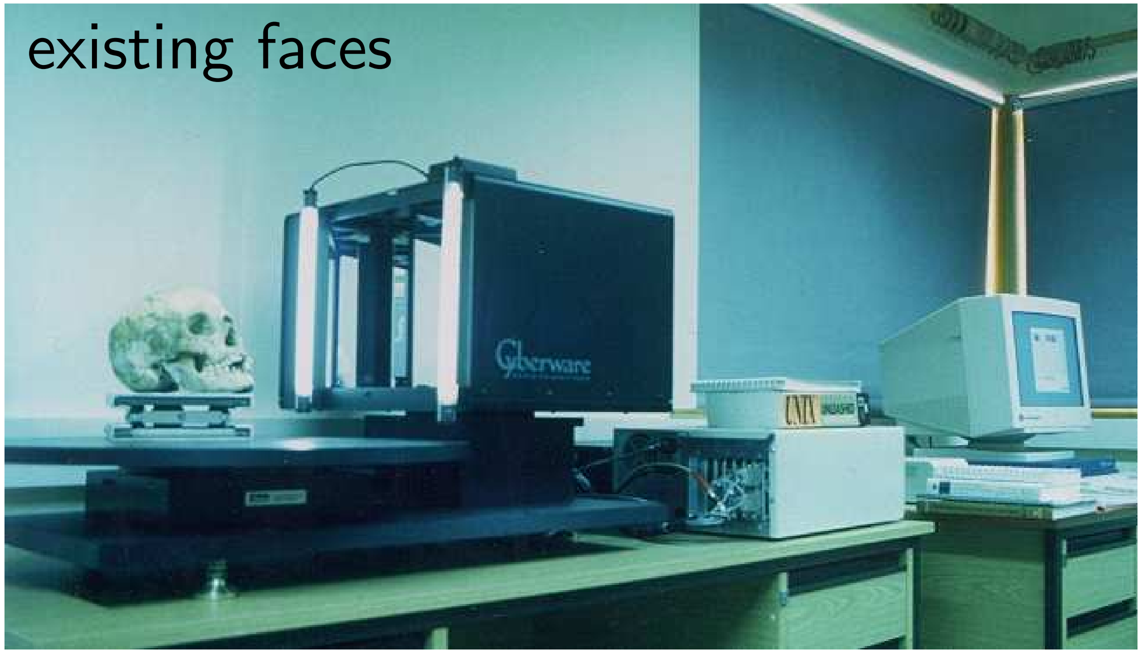


Polygonal Representation



Getting the surface Data

- 3D measuring techniques
- Photogrametric method
- Sculpting interactive method
- Assembling from simple shapes
- Creating new faces from existing faces



Photogrametric method

- in basic form the two photographs (from ortogonal directions) of the face are taken
- the two coordinates for the same point from both photographs are measured
- distortions, very low precision

Better Photogrametric method

We need information about view-transformations for each photograph

$$|x, y, z, 1| . \begin{vmatrix} T1_{11} & T1_{12} & T1_{13} \\ T1_{21} & T1_{22} & T1_{23} \\ T1_{31} & T1_{32} & T1_{33} \\ T1_{41} & T1_{42} & T1_{43} \end{vmatrix} = |x'_1, y'_1, h_1|$$

$$|x, y, z, 1| . \begin{vmatrix} T2_{11} & T2_{12} & T2_{13} \\ T2_{21} & T2_{22} & T2_{23} \\ T2_{31} & T2_{32} & T2_{33} \\ T2_{41} & T2_{42} & T2_{43} \end{vmatrix} = |x'_2, y'_2, h_2|$$

The coordinates measured from photographs:

$$x_{m_1} = \frac{x'_1}{h_1} \qquad x_{m_2} = \frac{x'_2}{h_2}$$

$$y_{m_1} = \frac{y'_1}{h_1} \qquad y_{m_2} = \frac{y'_2}{h_2}$$

Better Photogrametric method

After substitution and rewriting in matrix form we will get:

$$
\begin{vmatrix}
T1_{11} - x_{m_1}T1_{13} & T1_{21} - x_{m_1}T1_{23} & T1_{31}x_{m_1}T1_{33} \\
T1_{12} - y_{m_1}T1_{13} & T1_{22} - y_{m_1}T1_{23} & T1_{32}y_{m_1}T1_{33} \\
T2_{11} - x_{m_2}T2_{13} & T2_{21} - x_{m_2}T2_{23} & T2_{31}x_{m_2}T2_{33} \\
T2_{12} - y_{m_2}T2_{13} & T2_{22} - y_{m_2}T2_{23} & T2_{32}y_{m_2}T2_{33}
\end{vmatrix}
\begin{vmatrix}
x \\
y \\
z
\end{vmatrix}
=
$$

$$
\begin{vmatrix}
x_{m_1}T1_{43} - T1_{41} \\
y_{m_1}T1_{43} - T1_{42} \\
x_{m_2}T2_{43} - T2_{41} \\
y_{m_2}T2_{43} - T2_{42}
\end{vmatrix}
$$

Which is $[A].[X] = [B]$ – the four equations for three unknown x, y, z. In general for n views we will get n T matrices $\Rightarrow 2n$ equations for three unknowns.

Better Photogrametric method

Let's multiply both sides by $[A]^T$:

$$[A]^T[A][X] = [A]^T[B]$$

We will get three equations with three unknowns. Thus we reduce number of equations.

The method assumes we know transformations T. We can compute them first taking six reference points and solving 12 equations with 12 unknowns - each one for one item of T.

all these techniques work over polygonal mesh

- *interpolation* - assumes of two key states of the mesh
- *performance driven* - motion capture
- *direct parametrization* - reduction of controlled parts to lower number of parameters
- *pseudo-muscle based* - simulation using simplified models of muscles
- *muscle based* - physical simulation based on simulation at level of real muscles

Optical Motion Tracking

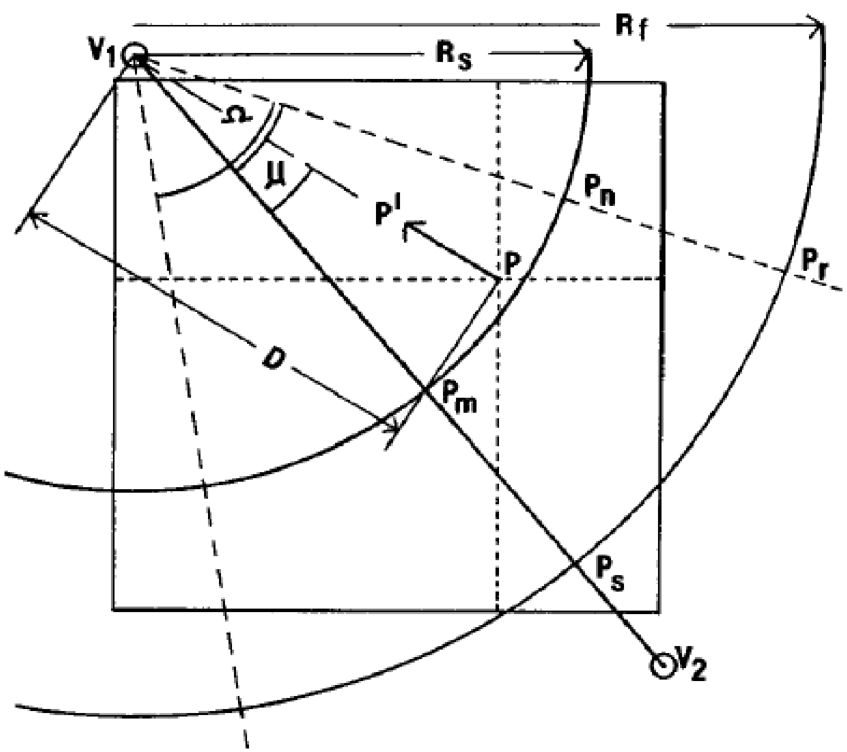
Optical Motion Tracking

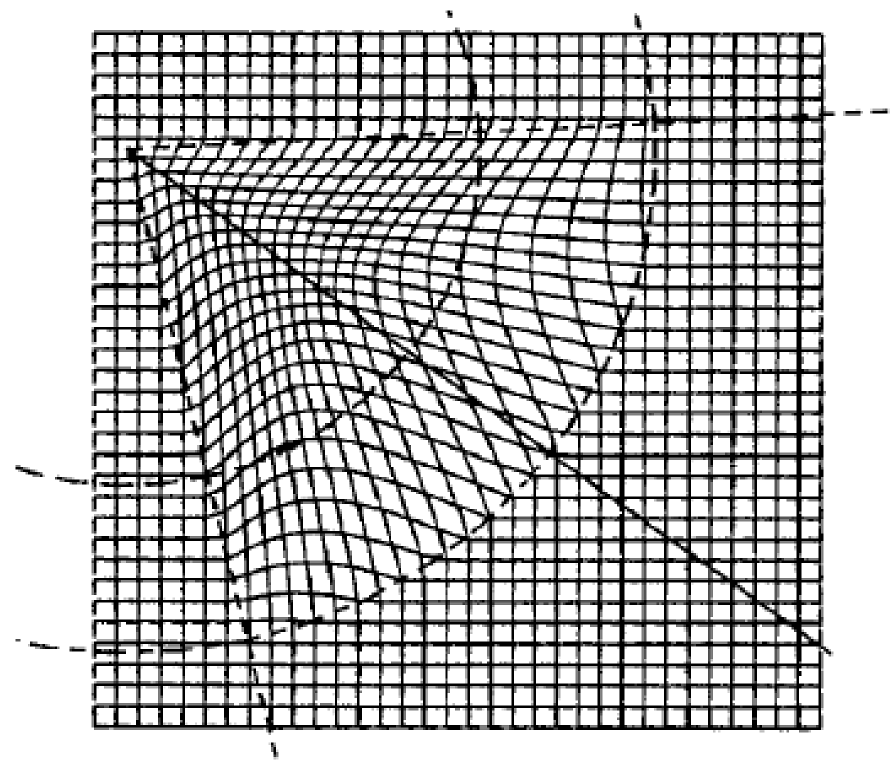
avatar

king-kong

MOCAP used in Polar Express

The muscle-skin interaction





Pictures taken from [Keith Waters87]

Pseudo Muscle-based Animation

- a face muscle is usually fixed to bone and to skin
- the displacement of a point p to p' can be expressed as $\vec{p'} - \vec{p} = \frac{\vec{F}}{k}$
- more accurate model describe new position p' as

$$
\begin{aligned}
p'_x &\propto f(K.A.R.x) \\
p'_y &\propto f(K.A.R.y) \\
p'_z &\propto f(K.A.R.z)
\end{aligned}
$$

where K is muscle spring constant, A angular displacement factor, and R is radial displacement factor

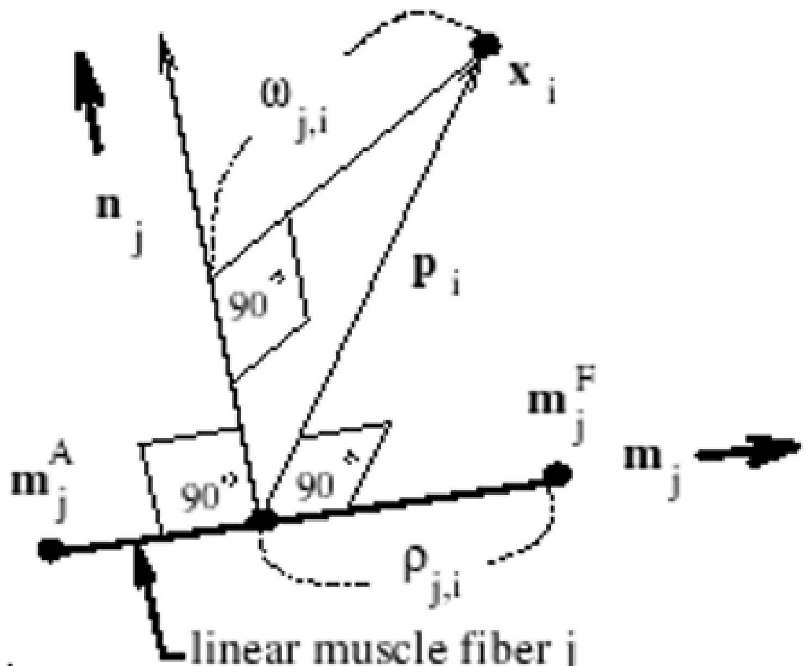
$A = cos(\frac{\mu}{2})$

$$
R = \begin{cases} cos((1 - \frac{D}{R_s})\frac{\pi}{2}) & \text{for points near the bone} \\ cos(\frac{D-R_s}{R_f-R_s}\frac{\pi}{2}) & \text{for points near the skin} \end{cases}
$$

Another approach

here the muscle m_j acts to the node x_i by the force

$$\mathbf{f}_i^j = \Theta_1(\varepsilon_{j,i})\Theta_2(\omega_{j,i})\mathbf{m}_j$$



$$\varepsilon_{j,i} = \frac{(\mathbf{m}_j^F - \mathbf{x}_i)\mathbf{m}_j}{\|\mathbf{m}_j^A - \mathbf{m}_j^F\|} \quad \text{length scaling factor}$$

$$\omega_{j,i} = \|\mathbf{p}_i - (\mathbf{p}_i\mathbf{n_j})\mathbf{n}_j\| \quad \text{width scaling factor}$$

Θ_1 and Θ_2 are force scaling functions.

The image taken from [Lee,95]

Application

The previous model can be applied in simplified form as:

$$\mathbf{x}_i' = \mathbf{x}_i + f(x_i, k_j).l_i.g(x_i, j).\mathbf{m}_j$$

by another words: x_i' is the new position of a point x_i on the skin evoked by contraction of the muscle j with muscle spring coefficient k_j and maximum length l_j.

The function $f(x_i, k_j)$ is partially based on the previous slide.

$$f(x_i, k_j) = \left(\frac{\|\mathbf{m}_j^F - \mathbf{x}_i\|}{\|\mathbf{m}_j^A - \mathbf{m}_j^F\|}\right)^{k_j}$$

The function $g(x_i, j)$ determines a measure of relation between the point x_i and the muscle m_j:

$$g(x_i, j) = \begin{cases} 1 & \text{the point belongs to the muscle} \\ 0 & \text{otherwise} \end{cases}$$

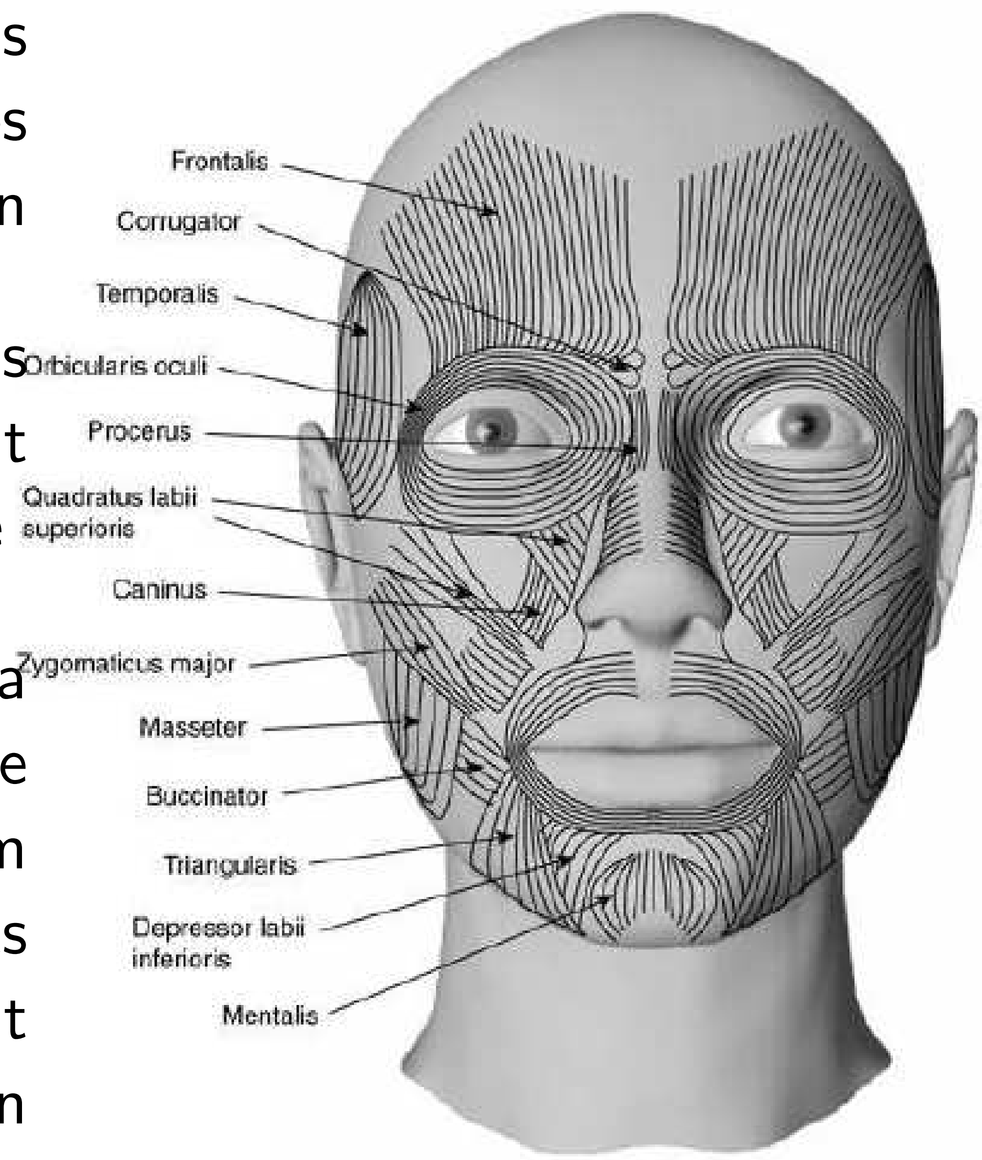
Speach Driven Simulation

- the animation of the face can be driven by a text
- our language consists of sounds with different significance
- sounds which change the meaning of words - *phonem*
- sounds which does not - *alophonem*
- configuration of face muscles typical for a phonem - *visem*
- there exists lot of phonema but corresponding visemes often have the same representation
- we can classify significant phonema and reduce the necessary set of geometry representations
- such phonema are then recognized in the text and mapped to the visems

DCGI

Speach Driven Simulation

- Each recognized visem has defined a relation to muscles which influence its representation
- the muscle is represented as a vector defined by significant antropometric points in the face
- using knowledge about phonema derived from text, muscle representations derived from geometry using significant points and with information about atributes of each muscle we can animate the face



References

DCGI

- Keith Waters: *A Muscle Model for Animating Three-DimensionalFacial Expression*, ACM Siggraph 1987.
- Yuencheng Lee, Demetri Terzopoulos, and Keith Waters: *Realistic Modeling for Facial Animation*, ACM Siggraph 1995.
- Ondřej Macháček: Diploma thesis, CTU Prague 2006.