A Tutorial on VIDEO COMPUTING ACCV-2000

Mubarak Shah
School Of Computer Science
University of Central Florida
Orlando, FL 32816
shah@cs.ucf.edu
http://cs.ucf.edu/~vision/

Course Contents

- Introduction
- Part I: Measurement of Image Motion
- Part II: Change Detection and Tracking
- Part III: Video Understanding
- Part IV: Video Phones and MPEG-4

Multimedia

- Text
- Graphics
- Audio
- Images
- Video

Imaging Configurations

- Stationary camera stationary objects
- Stationary camera moving objects
- Moving camera stationary objects
- Moving camera moving objects

Video

- sequence of images
- clip
- mosaic
- key frames

Steps in Video Computing

- Acquire (CCD arrays/synthesize (graphics))
- Process (image processing)
- Analyze (computer vision)
- Transmit (compression/networking)
- Store (compression/databases)
- Retrieve (computer vision/databases)
- Browse (computer vision/databases)
- Visualize (graphics)

Computer Vision

- Measurement of Motion
  - 2-D Motion
  - 3-D Motion
- Scene Change Detection
- Tracking
- Video Understanding
- Video Segmentation

Image Processing

- Filtering
- Compression
  - MPEG-1
  - MPEG-2
  - MPEG-4
  - MPEG-7 (Multimedia Content Description Interface)

Databases

- Storage
- Retrieval
- Video on demand
- Browsing
  - skim
  - abstract
  - key frames
  - mosaics

Networking

- Transmission
- ATM

Computer Graphics

- Visualization
- Image-based Rendering and Modeling
- Augmented Reality

Video Computing

- Computer Vision
- Image Processing
- Computer Graphics
- Databases
- Networks

PART I

Measurement of Motion

Contents

- Image Motion Models
- Optical Flow Methods
  - Horn & Schunck
  - Lucas and Kanade
  - Anandan et al
  - Mann & Picard
- Video Mosaics

3-D Rigid Motion

$$\begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix} = R\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + T = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + \begin{bmatrix} T_X \\ T_Y \\ T_Z \end{bmatrix}$$

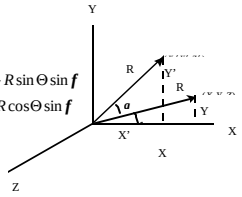
Rotation matrix (9 unknowns)

Translation (3 unknowns)

Rotation

$$X = R\cos\phi$$
$$Y = R\sin\phi$$

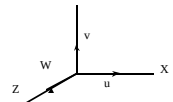
$$X' = R\cos(\Theta+\phi) = R\cos\Theta\cos\phi - R\sin\Theta\sin\phi$$
$$Y' = R\sin(\Theta+\phi) = R\sin\Theta\cos\phi + R\cos\Theta\sin\phi$$
$$X' = X\cos\Theta - Y\sin\Theta$$
$$Y' = X\sin\Theta + Y\cos\Theta$$



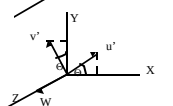
$$\begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix} = \begin{bmatrix} \cos\Theta & -\sin\Theta & 0 \\ \sin\Theta & \cos\Theta & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

Rotation (continued)

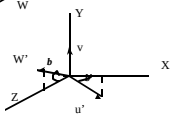
$$R = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$



$$R = \begin{bmatrix} \cos\Theta & -\sin\Theta & 0 \\ \sin\Theta & \cos\Theta & 0 \\ & & \end{bmatrix}$$



$$R = \begin{bmatrix} \cos\beta & 0 & -\sin\beta \\ 0 & 1 & 0 \\ & & \end{bmatrix}$$



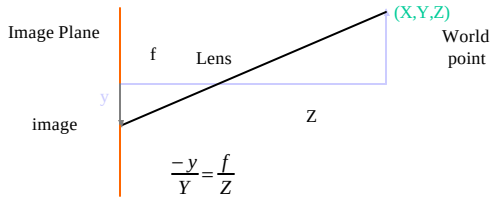
Euler Angles

$$R = R_z^{\alpha} R_y^{\beta} R_x^{\gamma} = \begin{bmatrix} \cos\alpha\cos\beta & \cos\alpha\sin\beta\sin\gamma - \sin\alpha\cos\gamma & \cos\alpha\sin\beta\cos\gamma + \sin\alpha\sin\gamma \\ \sin\alpha\cos\beta & \sin\alpha\sin\beta\sin\gamma + \cos\alpha\cos\gamma & \sin\alpha\sin\beta\cos\gamma - \cos\alpha\sin\gamma \\ -\sin\beta & \cos\beta\sin\gamma & \cos\beta\cos\gamma \end{bmatrix}$$

if angles are small( )

$$R = \begin{bmatrix} 1 & -\alpha & \beta \\ \alpha & 1 & -\gamma \\ & & \end{bmatrix}$$

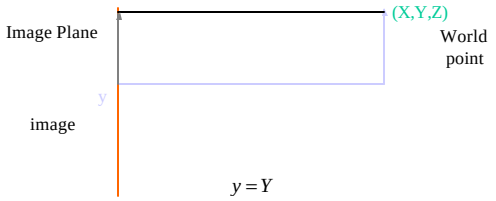
Perspective Projection



$$\frac{-y}{Y}=\frac{f}{Z}$$

$$y=-\frac{fY}{Z} \quad x=-\frac{fX}{Z}$$

Orthographic Projection



$$y=Y$$

$$x=X$$

Displacement Model

Orthographic Projection

$$x = X$$

$$y = Y$$

(x,y)=image coordinates,
(X,Y,Z)=world coordinates

$$\begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix} = R\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + T = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + \begin{bmatrix} T_X \\ T_Y \\ T_Z \end{bmatrix}$$

$$x' = r_{11}x + r_{12}y + (r_{13}Z + T_X)$$

$$y' = r_{21}x + r_{22}y + (r_{23}Z + T_Y)$$

$$x' = a_1x + a_2y + b_1$$

$$y' = a_3x + a_4y + b_2$$

$$\mathbf{x}' = \mathbf{A}\mathbf{x} + \mathbf{b}$$ Affine Transformation

Orthographic Projection (contd.)

$$\begin{bmatrix} X' \\ Y \\ Z' \end{bmatrix} = R\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + T = \begin{bmatrix} 1 & -\alpha & \beta \\ \alpha & 1 & \gamma \\ -\beta & \gamma & 1 \end{bmatrix}\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + \begin{bmatrix} T_X \\ T_Y \\ T_Z \end{bmatrix}$$

$$x' = x - \alpha y + \beta Z + T_X$$

$$y' = \alpha x + y - \gamma Z + T_Y$$

Plane+Perspective(projective)

$$aX+bY+cZ=1$$

equation of a plane

$$\begin{bmatrix} a & b & c \end{bmatrix}\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = 1$$

$$\begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix} = R\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + T\begin{bmatrix} a & b & c \end{bmatrix}\begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

$$\begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix} = R\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + T$$

$$\begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix} = A\begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

3d rigid motion

$x' = \frac{X'}{Z'}$ $y' = \frac{Y'}{Z'}$ focal length = -1

Plane+perspective (contd.)

$$x' = \frac{a_1 x + a_2 y + a_3}{a_7 x + a_8 y + 1} \qquad \mathbf{X}' = \frac{\mathbf{A}\mathbf{X} + b}{C^T \mathbf{X} + 1}$$

scale ambiguity $y' = \frac{a_4 x + a_5 y + a_6}{a_7 x + a_8 y + 1}$ find a's by least squares

Projective

$$\begin{bmatrix} x & y & 1 & 0 & 0 & 0 & -xx' & -yx' \\ 0 & 0 & 0 & x & y & 1 & -xy' & -yy' \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \\ a_7 \\ a_8 \end{bmatrix} = \begin{bmatrix} x' \\ y' \end{bmatrix}$$

Summary of Displacement Models

Translation $x' = x + b_1$, $y' = y + b_2$

Rigid $x' = x\cos\theta - y\sin\theta + b_1$, $y' = x\sin\theta + y\cos\theta + b_2$

Affine $x' = a_1 x + a_2 y + b_1$, $y' = a_3 x + a_4 y + b_2$

Projective $x' = \frac{a_1 x + a_2 y + b_1}{c_1 x + c_2 y + 1}$, $y' = \frac{a_3 x + a_4 y + b_1}{c_1 x + c_2 y + 1}$

Biquadratic $x' = a_1 + a_2 x + a_3 y + a_4 x^2 + a_5 y^2 + a_6 xy$, $y' = a_7 + a_8 x + a_9 y + a_{10} x^2 + a_{11} y^2 + a_{12} xy$

Bilinear $x' = a_1 + a_2 x + a_3 y + a_4 xy$, $y' = a_5 + a_6 x + a_7 y + a_8 xy$

Pseudo Perspective $x' = a_1 + a_2 x + a_3 y + a_4 x^2 + a_5 xy$, $y' = a_6 + a_7 x + a_8 y + a_4 xy + a_5 y^2$

Displacement Models (contd)

- Translation
  - simple
  - used in block matching
  - no zoom, no rotation, no pan and tilt
- Rigid
  - rotation and translation
  - no zoom, no pan and tilt

Displacement Models (contd)

- Affine
  - rotation about optical axis only
  - can not capture pan and tilt
  - orthographic projection
- Projective
  - exact eight parameters (3 rotations, 3 translations and 2 scalings)
  - difficult to estimate

Displacement Models (contd)

- Biquadratic
  - obtained by second order Taylor series
  - 12 parameters
- Bilinear
  - obtained from biquadratic model by removing square terms
  - most widely used
  - not related to any physical 3D motion
- Pseudo-perspective
  - obtained by removing two square terms and constraining four remaining to 2 degrees of freedom

Instantaneous Velocity Model

3-D Rigid Motion

$$\begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix} = \begin{bmatrix} 1 & -\alpha & \beta \\ \alpha & 1 & -\gamma \\ -\beta & \gamma & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + \begin{bmatrix} T_X \\ T_Y \\ T_Z \end{bmatrix}$$

$$\begin{bmatrix} X'-X \\ Y'-Y \\ Z'-Z \end{bmatrix} = \begin{bmatrix} 0 & -\alpha & \beta \\ \alpha & 0 & -\gamma \\ -\beta & \gamma & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + \begin{bmatrix} T_X \\ T_Y \\ T_Z \end{bmatrix}$$

$$\begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{Z} \end{bmatrix} = \begin{bmatrix} 0 & -\Omega_3 & \Omega_2 \\ \Omega_3 & 0 & -\Omega_1 \\ -\Omega_2 & \Omega_1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + \begin{bmatrix} T_X \\ T_Y \\ T_Z \end{bmatrix}$$

$$\dot{\mathbf{X}} = \Omega \times \mathbf{X} + \mathbf{V}$$

$$\begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix} = \left(\begin{bmatrix} 0 & -\alpha & \beta \\ \alpha & 0 & -\gamma \\ -\beta & \gamma & 0 \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \right) \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + \begin{bmatrix} T_X \\ T_Y \\ T_Z \end{bmatrix}$$

3-D Rigid Motion

$$\dot{\mathbf{X}} = \Omega \times \mathbf{X} + \mathbf{V}$$

$$\dot{X} = \Omega_2 Z - \Omega_3 Y + V_1$$

$$\dot{Y} = \Omega_3 X - \Omega_1 Z + V_2$$

$$\dot{Z} = \Omega_1 Y - \Omega_2 X + V_3$$

Orthographic Projection

$u = \dot{x} = \Omega_2 Z - \Omega_3 y + V_1$ (u,v) is optical flow

$v = \dot{y} = \Omega_3 x - \Omega_1 Z + V_2$

$$\dot{\mathbf{X}} = \Omega \times \mathbf{X} + \mathbf{V}$$

$$\dot{X} = \Omega_2 Z - \Omega_3 Y + V_1$$

$$\dot{Y} = \Omega_3 X - \Omega_1 Z + V_2$$

$$\dot{Z} = \Omega_1 Y - \Omega_2 X + V_3$$

Perspective Projection (arbitrary flow)

$$x = \frac{fX}{Z} \qquad u = \dot{x} = \frac{fZ\dot{X} - fX\dot{Z}}{Z^2} = f\frac{\dot{X}}{Z} - x\frac{\dot{Z}}{Z}$$

$$y = \frac{fY}{Z} \qquad v = \dot{y} = \frac{fZ\dot{Y} - fY\dot{Z}}{Z^2} = f\frac{\dot{Y}}{Z} - y\frac{\dot{Z}}{Z}$$

$$u = f\left(\frac{V_1}{Z} + \Omega_2\right) - \frac{V_3}{Z}x - \Omega_3 y - \frac{\Omega_1}{f}xy + \frac{\Omega_2}{f}x^2$$

$$v = f\left(\frac{V_2}{Z} - \Omega_1\right) + \Omega_3 x - \frac{V_3}{Z}y + \frac{\Omega_2}{f}xy - \frac{\Omega_1}{f}y^2$$

Plane+orthographic(Affine)

$Z = a + bX + cY$

$u = V_1 + \Omega^2 Z - \Omega^3 y$

$u = b^1 + a^1 x + a^2 y$

$$\mathbf{u} = \mathbf{A}\,\mathbf{x} + \mathbf{b}$$

$b_1 = V_1 + a\Omega_2$

$a_1 = b\Omega_2$

$a_2 = c\Omega_2 - \Omega_3$

$b_2 = V_2 - a\Omega_1$

$a_3 = \Omega_3 - b\Omega_1$

$a_4 = -c\Omega_1$

Plane+Perspective (pseudo perspective)

$$u = f\left(\frac{V_1}{Z} + \Omega_2\right) - \frac{V_3}{Z}x - \Omega_3 y - \frac{\Omega_1}{f}xy + \frac{\Omega_2}{f}x^2 \qquad Z = a + bX + cY$$

$$v = f\left(\frac{V_2}{Z} - \Omega_1\right) + \Omega_3 x - \frac{V_3}{Z}y + \frac{\Omega_2}{f}xy - \frac{\Omega_1}{f}y^2 \qquad \frac{1}{Z} = \frac{1}{a} - \frac{b}{a}x - \frac{c}{a}y$$

$$u = a_1 + a_2 x + a_3 y + a_4 x^2 + a_5 xy$$

$$v = a_{64} + a_7 x + a_8 y + a_4 xy + a_5 y^2$$

Measurement of Image Motion

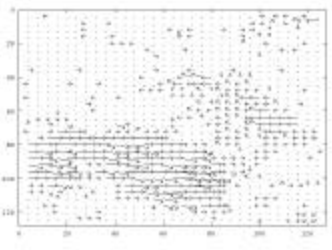
- Local Motion (Optical Flow)
- Global Motion (Frame Alignment)

Computing Optical Flow

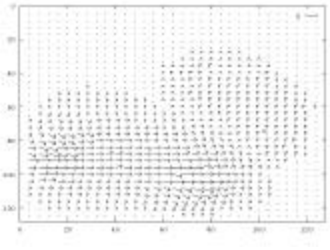
Image from Hamburg Taxi seq

Image from Hamburg Taxi seq

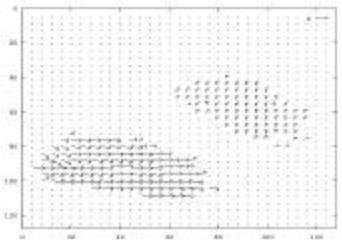
Fleet & Jepson optical flow



Horn & Schunck optical flow



Tian & Shah optical flow



Horn&Schunck Optical Flow

$$f(x,y,t) = f(x+dx, y+dy, t+dt)$$

⇩ Taylor Series

$$f(x,y,t) = f(x,y,t) + \frac{\partial f}{\partial x}dx + \frac{\partial f}{\partial y}dy + \frac{\partial f}{\partial t}dt$$

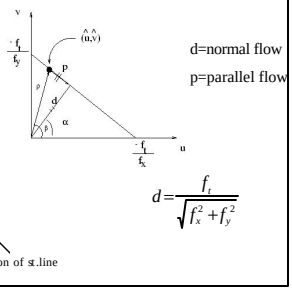
$$f_x dx + f_y dy + f_t dt = 0$$

brightness constancy eq

Interpretation of optical flow eq

$$f_x u + f_y v + f_t = 0$$

$$v = -\frac{f_x}{f_y}u - \frac{f_t}{f_y}$$



d=normal flow

p=parallel flow

$$d = \frac{f_t}{\sqrt{f_x^2 + f_y^2}}$$

Equation of st.line

Horn&Schunck (contd)

$$\iint \{(f_x u + f_y v + f_t)^2 + \lambda(u_x^2 + u_y^2 + v_x^2 + v_y^2)\}dxdy$$

⇩ min

variational calculus

$$(f_x u + f_y v + f_t)f_x + \lambda(\Delta^2 u) = 0 \qquad u = u_{av} - f_x \frac{P}{D}$$

$$(f_x u + f_y v + f_t)f_y + \lambda(\Delta^2 v) = 0 \qquad v = v_{av} - f_y \frac{P}{D}$$

⇩ discrete version

$$(f_x u + f_y v + f_t)f_x + \lambda(u - u_{av}) = 0 \qquad P = f_x u_{av} + f_y v_{av} + f_t$$

$$(f_x u + f_y v + f_t)f_y + \lambda(v - v_{av}) = 0 \qquad D = \lambda + f_x^2 + f_y^2$$

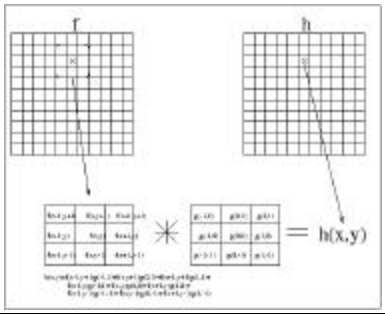
Algorithm-1

- k=0
- Initialize $u^K \quad v^K$
- Repeat until some error measure is satisfied

$$u^K = u_{av}^{k-1} - f_x \frac{P}{D} \qquad P = f_x u_{av} + f_y v_{av} + f_t$$

$$v = v_{av}^{K-1} - f_y \frac{P}{D} \qquad D = \lambda + f_x^2 + f_y^2$$

Convolution

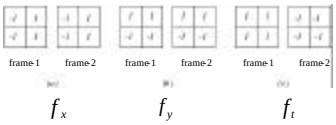


Convolution (contd)

$$h(x,y) = \sum_{i=-1}^{1} \sum_{j=-1}^{1} f(x+i, y+j) g(i,j)$$

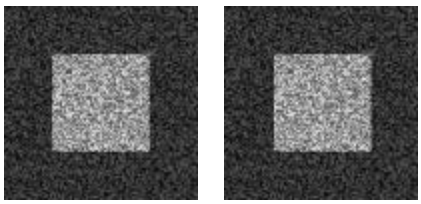
$$h(x,y) = f(x,y) * g(x,y)$$

Derivative Masks

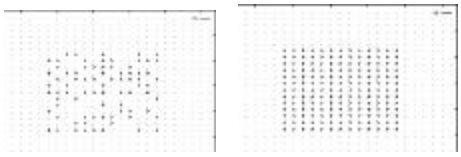


f_x f_y f_t

Synthetic Images



Results



One iteration

10 iterations

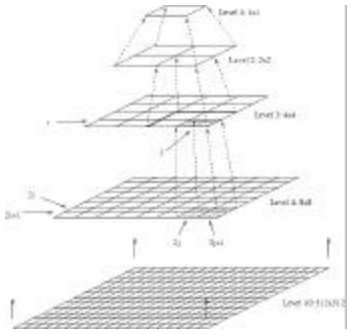
Comments

- Algorithm-1 works only for small motion.
- If object moves faster, the brightness changes rapidly, 2x2 or 3x3 masks fail to estimate spatiotemporal derivatives.
- Pyramids can be used to compute large optical flow vectors.

Pyramids

- Very useful for representing images.
- Pyramid is built by using multiple copies of image.
- Each level in the pyramid is 1/4 of the size of previous level.
- The lowest level is of the highest resolution.
- The highest level is of the lowest resolution.

Pyramid



Gaussian Pyramid

Algorithm-2 (Optical Flow)

- Create Gaussian pyramid of both frames.
- Repeat
  - apply algorithm-1 at the current level of pyramid.
  - propagate flow by using bilinear interpolation to the next level, where it is used as an initial estimate.
  - Go back to step 2

Horn&Schunck Method

- Good only for translation model.
- Over-smoothing of boundaries.
- Does not work well for real sequences.

Other Optical Flow Methods

Important Issues

- What motion model?
- What function to be minimized?
- What minimization method?

Minimization Methods

- Least Squares fit
- Weighted Least Squares fit
- Newton-Raphson
- Gradient Descent
- Levenberg-Marquadet

Lucas & Kanade (Least Squares)

- Optical flow eq

$$f_x u + f_y v = -f_t$$

- Consider 3 by 3 window

$$f_{x1} u + f_{y1} v = -f_{t1}$$
$$\vdots$$
$$f_{x9} u + f_{y9} v = -f_{t9}$$

$$\begin{bmatrix} f_{x1} & f_{y1} \\ \vdots & \vdots \\ f_{x9} & f_{y9} \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} -f_{t1} \\ \vdots \\ -f_{t9} \end{bmatrix}$$

$$\mathbf{Au} = \mathbf{f_t}$$

Lucas & Kanade

$$\mathbf{Au} = \mathbf{f_t}$$

$$\mathbf{A^T A u} = \mathbf{A^T f_t}$$

$$\mathbf{u} = (\mathbf{A^T A})^{-1} \mathbf{A^T f_t}$$

$$\min \sum_{i=-2}^{2} \sum_{j=-2}^{2} (f_{xi} u + f_{yi} v + f_{ti})^2$$

Lucas & Kanade

$$\min \sum_{i=-2}^{2} \sum_{j=-2}^{2} (f_{xi} u + f_{yi} v + f_{ti})^2$$

$$\sum (f_{xi} u + f_{yi} v + f_{ti}) f_{xi} = 0$$

$$\sum (f_{xi} u + f_{yi} v + f_{ti}) f_{yi} = 0$$

Lucas & Kanade

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \sum f_{xi}^2 & \sum f_{xi} f_{yi} \\ \sum f_{xi} f_{yi} & \sum f_{yi}^2 \end{bmatrix}^{-1} \begin{bmatrix} -\sum f_{xi} f_{ti} \\ -\sum f_{yi} f_{ti} \end{bmatrix}$$

Lucas & Kanade

$$\min \sum_{i=-2}^{2} \sum_{j=-2}^{2} w_i (f_{xi} u + f_{yi} v + f_{ti})^2$$

$$\mathbf{WAu} = \mathbf{Wf_t}$$

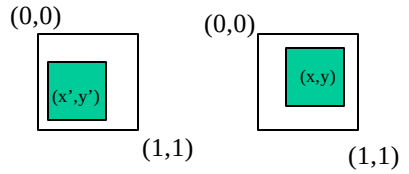
$$\mathbf{A^T W A u} = \mathbf{A^T W f_t}$$

$$\mathbf{u} = (\mathbf{A^T W A})^{-1} \mathbf{A^T W f_t}$$

Anandan

Affine

Affine



$$u(x,y) = a_1 x + a_2 y + b_1$$
$$v(x,y) = a_3 x + a_4 y + b_2$$

$$X' = X - U$$

Anandan

$$u(x,y) = a_1 x + a_2 y + b_1$$
$$v(x,y) = a_3 x + a_4 y + b_2$$

•Affine

$$\begin{bmatrix} u(x,y) \\ v(x,y) \end{bmatrix} = \begin{bmatrix} x & y & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & x & y & 1 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ b_1 \\ a_3 \\ a_4 \\ b_2 \end{bmatrix}$$

$$\mathbf{u(x) = X(x)a}$$

Anandan

$$\mathbf{u(x) = X(x)a}$$

$$E(\boldsymbol{d}) = \sum_x (f_t + f_{\mathbf{x}}^T \boldsymbol{d})^2$$

$$f_{\mathbf{x}} = \begin{bmatrix} f_x \\ f_y \end{bmatrix}$$

$$E(\boldsymbol{d}) = \sum_x (f_t + f_{\mathbf{x}}^T \mathbf{X}\boldsymbol{d})^2$$

min

Optical flow constraint eq

$$f_x u + f_y v = -f_t$$

Anandan

$$\left[\sum \mathbf{X}^{\mathbf{T}} (\mathbf{f}_{\mathbf{X}}) (\mathbf{f}_{\mathbf{x}})^T \mathbf{X}\right] \boldsymbol{d}\mathbf{a} = -\sum \mathbf{X}^T \mathbf{f}_{\mathbf{X}} f_t$$

$$Ax = b$$

Basic Components

- Pyramid construction
- Motion estimation
- Image warping
- Coarse-to-fine refinement

Mann & Picard

Projective

Projective Flow (weighted)

$$u_f f_x + v_f f_y + f_t = 0$$

$$\mathbf{u}_m^T \mathbf{f}_\mathbf{x} + f_t = 0$$

$$\mathbf{x}' = \frac{A\mathbf{x} + \mathbf{b}}{\mathbf{C}^{\mathbf{T}}\mathbf{x} + 1}$$

$$\mathbf{u}_m = \mathbf{x}' - \mathbf{x} = \frac{A\mathbf{x} + \mathbf{b}}{\mathbf{C}^{\mathbf{T}}\mathbf{x} + 1}$$

Projective Flow (weighted)

$$\boldsymbol{e}_{flow} = \sum (\mathbf{u}_\mathbf{m}^{\mathbf{T}}\mathbf{f}_\mathbf{X} + f_\mathbf{t})^2$$

$$= \sum ((\frac{A\mathbf{x}+\mathbf{b}}{\mathbf{C}\mathbf{x}^{\mathbf{T}}+1} - \mathbf{x})^T \mathbf{f}_\mathbf{x} + f_t)^2$$

$$= \sum ((\mathbf{A}\mathbf{x}+\mathbf{b} - (\mathbf{C}^{\mathbf{T}}\mathbf{x}+1)x)^T \mathbf{f}_x + (\mathbf{C}^T\mathbf{x}+1) f_t)^2$$

⇩ **minimize**

Projective Flow (weighted)

$$(\sum \boldsymbol{f}\boldsymbol{f}^T)\mathbf{a} = \sum (\mathbf{x}^{\mathbf{T}}\mathbf{f}_x - f_t)\boldsymbol{f}$$

$$a = [a_{11}, a_{12}, b_1, a_{21}, a_{22}, b_2, c_1, c_2]^T$$

$$\boldsymbol{f}^t = [f_x x, f_x y, f_x, f_y x, f_y y, f_y, x f_t - x^2 f_x - xy f_y, y f_t - xy f_x - y^2 f_y]$$

Projective Flow (unweighted)

Bilinear

$$\mathbf{x}' = \frac{A\mathbf{x} + \mathbf{b}}{\mathbf{C}^{\mathbf{T}}\mathbf{x} + 1}$$

⇩ Taylor Series

$$u_m + x = a_1 + a_2 x + a_3 y + a_4 xy$$

$$v_m + y = a_5 + a_6 x + a_7 y + a_8 xy$$

Pseudo-Perspective

$$\mathbf{x}' = \frac{A\,\mathbf{x} + \mathbf{b}}{\mathbf{C}^{\mathbf{T}}\mathbf{x} + 1}$$

⇩ **Taylor Series**

$$x + u_m = a_1 + a_2 x + a_3 y + a_4 x^2 + a_5 xy$$
$$y + v_m = a_6 + a_7 x + a_8 y + a_4 xy + a_5 y^2$$

Projective Flow (unweighted)

$$\boldsymbol{e}_{flow} = \sum (\mathbf{u}_{\mathbf{m}}^{\mathbf{T}}\mathbf{f}_{\mathbf{X}} + f_{\mathbf{t}})^2$$

⇩ **Minimize**

Bilinear and Pseudo-Perspective

$$(\sum \Phi \Phi^T)\mathbf{q} = -\sum f_t \Phi$$

$$\Phi^T = [f_x(xy, x, y, 1), \quad f_y(xy, x, y, 1)]$$

bilinear

$$\Phi^T = [f_x(x, y, 1) \quad f_y(x, y, 1) \quad c_1 \quad c_2]$$

$$c_1 = x^2 f_x + xy f_x$$
$$c_2 = xy f_x + y^2 f_y$$

Pseudo perspective

Algorithm-1

- Estimate "q" (using approximate model, e.g. bilinear model).
- Relate "q" to "p"
  - select four points S1, S2, S3, S4
  - apply approximate model using "q" to compute (x_k', y_k')
  - estimate exact "p":

True Projective

$$x' = \frac{a_1 x + a_2 y + b_1}{c_1 x + c_2 y + 1}$$

$$y' = \frac{a_3 x + a_4 y + b_1}{c_1 x + c_2 y + 1}$$

$$\begin{bmatrix} x_k' \\ y_k' \end{bmatrix} = \begin{bmatrix} x_k & y_k & 1 & 0 & 0 & 0 & -x_k x_k' & -y_k x_k' \\ 0 & 0 & 0 & x_k & y_k & 1 & -x_k y_k' & -y_k y_k' \end{bmatrix} \mathbf{a}$$

$$\mathbf{a} = [a_1 \quad a_2 \quad b_1 \quad a_3 \quad a_4 \quad b_2 \quad c_1 \quad c_1]^T$$

$$\begin{bmatrix} x_1' \\ y_1' \\ \\ \\ x_k' \\ y_k' \end{bmatrix} = \begin{bmatrix} x_1 & y_1 & 1 & 0 & 0 & 0 & -x_1 x_1' & -y_1 x_1' \\ 0 & 0 & 0 & x_1 & y_1 & 1 & -x_1 y_1' & -y_1 y_1' \\ & & & & & & & \\ & & & & & & & \\ x_k & y_k & 1 & 0 & 0 & 0 & -x_k x' & -y_k x_k' \\ 0 & 0 & 0 & x_k & y_k & 1 & -x_k y_k' & -y_k y' \end{bmatrix} \mathbf{a}$$

P = **Aa**

Perform least squares fit to compute a.

Final Algorithm

- A Gaussian pyramid of three or four levels is constructed for each frame in the sequence.
- The parameters "p" are estimated at the top level of the pyramid, between the two lowest resolution images, "g" and "h", using algorithm-1.

Final Algorithm

- The estimated "p" is applied to the next higher resolution image in the pyramid, to make images at that level nearly congruent.
- The process continues down the pyramid until the highest resolution image in the pyramid is reached.

Video Mosaics

- Mosaic aligns different pieces of a scene into a larger piece, and seamlessly blend them.
  - High resolution image from low resolution images
  - Increased filed of view

Steps in Generating A Mosaic

- Take pictures
- Pick reference image
- Determine transformation between frames
- Warp all images to the same reference view

Applications of Mosaics

- Virtual Environments
- Computer Games
- Movie Special Effects
- Video Compression

Webpages

- http://n1nlf1.eecg.toronto.edu/tip.ps.gz
  Video Orbits of the projective group, S. Mann and R. Picard. (paper)
- http://wearcam.org/pencigraphy (C code for generating mosaics)

Webpages

- http://ww-bcs.mit.edu/people/adelson/papers.html
  - The Laplacian Pyramid as a compact code, Burt and Adelson, IEEE Trans on Communication, 1983.
- J. Bergen, P. Anandan, K. Hanna, and R. Hingorani, "Hierarchical Model-Based Motion Estimation", ECCV-92, pp 237-22.

Webpages

- http://www.cs.cmu.edu/afs/cs/project/cil/ftp/html/v-source.html (c code for several optical flow algorithms)
- ftp://csd.uwo.ca/pub/vision
  Performance of optical flow techniques (paper)

Barron, Fleet and Beauchermin

Webpages

- http://www.wisdom.weizmann.ac.il/~irani/abstracts/mosaics.html ("Efficient representations of video sequences and their applications", Michal Irani, P. Anandan, Jim Bergen, Rakesh Kumar, and Steve Hsu)
- R. Szeliski. "Video mosaics for virtual environments", IEEE Computer Graphics and Applications, pages,22-30, March 1996.

Part II

Change Detection and Tracking

Contents

- Change Detection
- Pfinder
- W4
- Skin Detection
- Tracking People Using Color

Change Detection

Main Points

- Detect pixels which are changing due to motion of objects.
- Not necessarily measure motion (optical flow), only detect motion.
- A set of connected pixels which are changing may correspond to moving object.

Picture Difference

$$D_i(x,y) = \left\{ \begin{array}{ll} 1 & if \quad DP(x,y) > T \\ 0 & \ldots\ldots otherwise \end{array} \right\}$$

$$DP(x,y) = | f_i(x,y) - f_{i-1}(x,y) |$$

$$DP(x,y) = \sum_{i=-m}^{m} \sum_{j=-m}^{m} | f_i(x+i, y+j) - f_{i-1}(x+i, y+j) |$$

$$DP(x,y) = \sum_{i=-m}^{m} \sum_{i=-m}^{m} \sum_{k=-m}^{m} | f_i(x+i, y+j) - f_{i+k}(x+i, y+j) |$$

Background Image

• The first image of a sequence without any moving objects, is background image.

• Median filter

$$B(x,y) = median\ (f_1(x,y), \ldots, f_n(x,y))$$

PFINDER

Pentland

Pfinder

- Segment a human from an arbitrary complex background.
- It only works for single person situations.
- All approaches based on background modeling work only for fixed cameras.

Algorithm

- Learn background model by watching 30 second video
- Detect moving object by measuring deviations from background model
- Segment moving blob into smaller blobs by minimizing covariance of a blob
- Predict position of a blob in the next frame using Kalman filter
- Assign each pixel in the new frame to a class with max likelihood.
- Update background and blob statistics

Learning Background Image

- Each pixel in the background has associated mean color value and a covariance matrix.
- The color distribution for each pixel is described by Gaussian.
- YUV color space is used.

Detecting Moving Objects

- After background model has been learned, Pfinder watches for large deviations from the model.
- Deviations are measured in terms of Mahalanobis distance in color.
- If the distance is sufficient then the process of building a blob model is started.

Detecting Moving Objects

- For each of k blob in the image, log-likelihood is computed

$$d_k = -.5(y-\mu_k)^T K_k^{-1}(y-\mu_k) - .5\ln|K_k| - .5m\ln(2\pi)$$

- Log likelihood values are used to classify pixels

$$s(x,y) = \arg\max_k (d_k(x,y))$$

Updating

- The statistical model for the background is updated.

$$K^t = E[(y-\mu^t)(y-\mu^t)^T]$$
$$\mu^t = (1-\alpha)\mu^{t-1} + \alpha y$$

- The statistics of each blob (mean and covariance) are re-computed.

W4 (Who, When, Where, What)

Davis

W4

- Compute "minimum"(M(x)), "maximum" (N(x)), and "largest absolute difference" (L(x)).

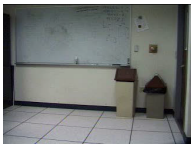
$$D_i(x,y) = \begin{Bmatrix} 1 & if & |M(x,y) - f_i(x,y)| > L(x,y) or \\ & & |N(x,y) - f_i(x,y)| > L(x,y) \\ 0 & \ldots & otherwise \end{Bmatrix}$$

- Theoretically, the performance of this tracker should be worse than others.
- Even if one value is far away from the mean, then that value will result in an abnormally high value of L.
- Having short training time is better for this tracker.

Limitations

- Multiple people
- Occlusion
- Shadows
- Slow moving people
- Still background objects and deposited objects
- Multiple processes (swaying of trees..)

Sohaib Khan & Mubarak Shah, "Tracking in Presence of Occlusion", ACCV-2000




Webpage

- http://www.cs.cmu.edu/~vsam
  - DARPA Visual Surveillance and Monitoring program

Skin Detection

Kjeldsen and Kender

Training

- Crop skin regions in the training images.
- Build histogram of training images.
- Ideally this histogram should be bi-modal, one peak corresponding to the skin pixels, other to the non-skin pixels.
- Practically there may be several peaks corresponding to skin, and non-skin pixels.

Training

- Apply threshold to skin peaks to remove small peaks.
- Label all values (colors) under skin peaks as "skin", and the remaining values as "non-skin".
- Generate a look-up table for all possible colors in the image, and assign "skin" or "non-skin" label.

Detection

- For each pixel in the image, determine its label from the "look-up table" generated during training.

Building Histogram

- Instead of incrementing the pixel counts in a particular histogram bin:
  - for skin pixel increment the bins centered around the given value by a Gaussian function.
  - For non-skin pixels decrement the bins centered around the given value by a smaller Gaussian function.

Tracking People Using Color

Fieguth and Terzopoulos

• Computer mean color vector for each sub region.

$$(r_i, g_i, b_i) = \frac{1}{|R_i|} \sum_{(x,y) \in R_i} (r(x,y), g(x,y), b(x,y))$$

Fieguth and Terzopoulos

• Compute goodness of fit.

$$\Psi_i = \frac{\max\left\{\frac{r_i}{\bar{r}_i}, \frac{g_i}{\bar{g}_i}, \frac{b_i}{\bar{b}_i}\right\}}{\min\left\{\frac{r_i}{\bar{r}_i}, \frac{g_i}{\bar{g}_i}, \frac{b_i}{\bar{b}_i}\right\}}$$

$(\bar{r}_i, \bar{g}_i, \bar{b}_i)$ Target

(r_i, g_i, b_i) Measurement

Fieguth and Terzopoulos

- Tracking

$$\Psi(x_H, y_H) = \sum_{i=1}^{N} \frac{\Psi_i(x_H + x_i, y_H + y_i)}{N}$$

$$(\hat{x}, \hat{y}) = \arg_{(x_H, y_H)} \min\{ \Psi(x_H, y_H)\}$$

Fieguth and Terzopoulos

- Non-linear velocity estimator

$$\text{if} \quad (\boldsymbol{r}(f).\boldsymbol{r}(f-1) > 0) \quad v(f) \; += \; \boldsymbol{d}\,\frac{\text{sgn}(\boldsymbol{r}(f))}{\Delta t}$$

$$\text{if} \quad (\boldsymbol{r}(f).v(f-1) < 0) \quad v(f) \; += \; \boldsymbol{d}\,\frac{\text{sgn}(\boldsymbol{r}(f))}{\Delta t}$$

$$\text{if} \quad (\boldsymbol{r}(f) = 0) \quad v(f) \; += \; \boldsymbol{d}\,\frac{\text{sgn}(v(f))}{2\Delta t}$$

Bibliography

- .J. K. Aggarwal and Q. Cai, "Human Motion Analysis: A Review", *Computer Vision and Image Understanding*, Vol. 73, No. 3, March, pp. 428-440, 1999
- .Azarbayejani, C. Wren and A. Pentland, "Real-Time 3D Tracking of the Human Body", MIT Media Laboratory, Perceptual Computing Section, TR No. 374, May 1996
- .W.E.L. Grimson *et. al.*, "Using Adaptive Tracking to Classify and Monitor Activities in a Site", *Proceedings of Computer Vision and Pattern Recognition*, Santa Barbara, June 23-25, 1998, pp. 22-29

Bibliography

- .Takeo Kanade *et. al.* "Advances in Cooperative Multi-Sensor Video Surveillance", *Proceedings of Image Understanding workshop*, Monterey California, Nov 20-23, 1998, pp. 3-24
- .Haritaoglu I., Harwood D, Davis L, "W^4 - Who, Where, When, What: A Real Time System for Detecting and Tracking People", *International Face and Gesture Recognition Conference*, 1998
- .Paul Fieguth, Demetri Terzopoulos, "Color-Based Tracking of Heads and Other Mobile Objects at Video Frame Rates", *CVPR 1997*, pp. 21-27

Part III

VIDEO UNDERSTANDING

Contents

Monitoring Human Behavior In an Office Environment

Doug Ayers and Mubarak Shah, "Recognizing Human Activities In an Office Environment", Workshop on Applications of Computer Vision, October, 1998

Goals of the System

- Recognize human actions in a room for which **prior knowledge** is available.
- Handle multiple people
- Provide a textual description of each action
- Extract "key frames" for each action

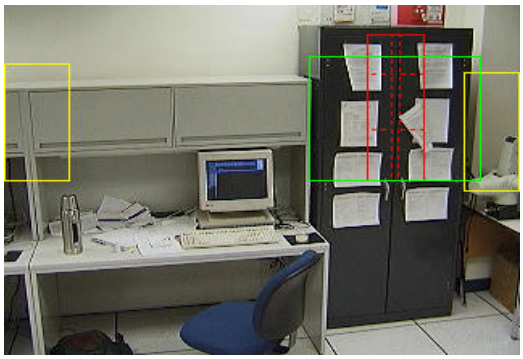
Possible Actions

- **Enter**
- **Leave**
- **Sitting** or **Standing**
- **Picking Up Object**
- **Put Down Object**
- **…..**

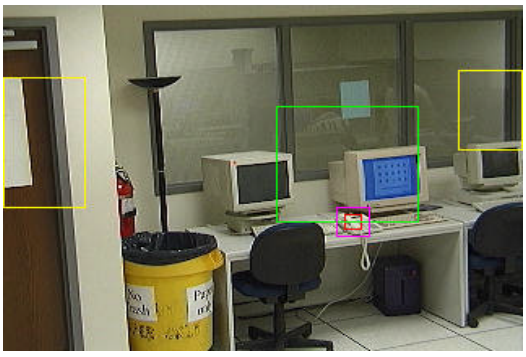
Prior Knowledge

- Spatial layout of the scene:
  - Location of **entrances** and **exits**
  - Location of **objects** and some information about how they are use
- Context can then be used to improve recognition and save computation

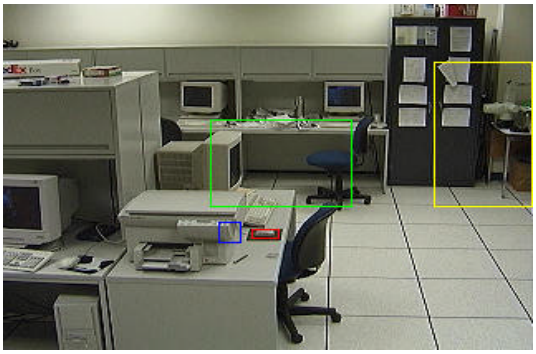
Layout of Scene 1



Layout of Scene 2



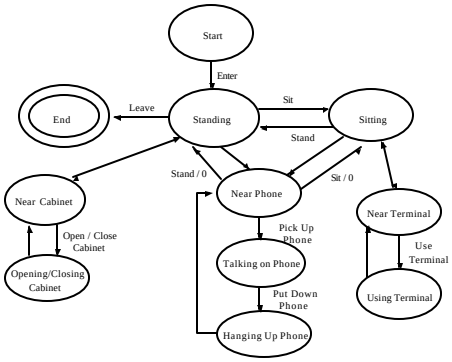
Layout of Scene 4



Major Components

- Skin Detection
- Tracking
- Scene Change Detection
- Action Recognition

State Model For Action Recognition

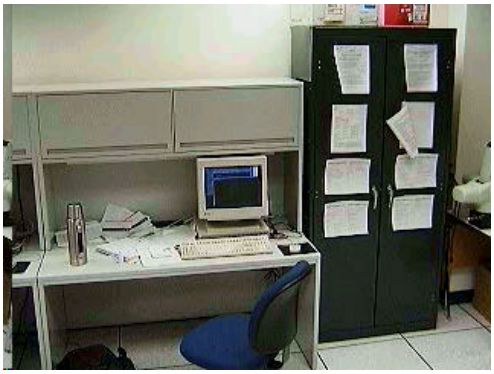


Key Frames

- Why get key frames?
  - Key frames take less space to store
  - Key frames take less time to transmit
  - Key frames can be viewed more quickly
- We use heuristics to determine when key frames are taken
  - Some are taken before the action occurs
  - Some are taken after the action occurs

Results

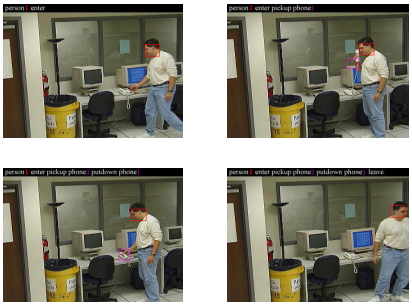
http://www.cs.ucf.edu/~ayers/research.html



Key Frames Sequence 1 (350 frames), Part 1

Key Frames Sequence 1 (350 frames), Part 2

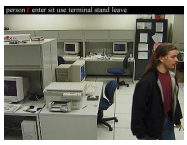
Key Frames Sequence 2 (200 frames)



Key Frames Sequence 3 (200 frames)

Key Frames Sequence 4 (399 frames), Part 1

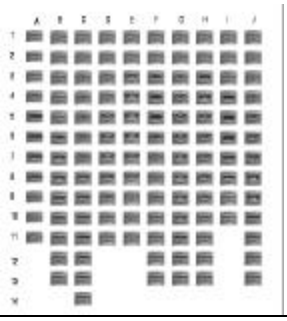
Key Frames Sequence 4 (399 frames), Part 2



Visual Lipreading

Li Nan, Shawn Dettmer, and Mubarak Shah, “Visual Lipreading”, Workshop on Face and Gesture Recognition, Zurich, 1995.

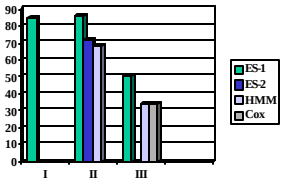
Image Sequences of “A” to “J”



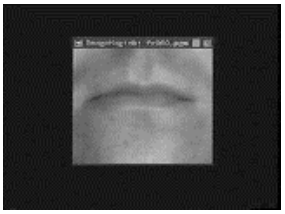
Particulars

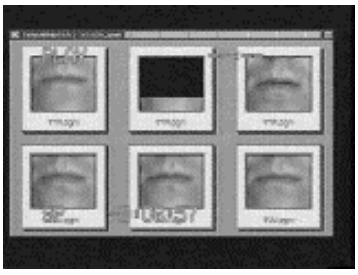
- Problem: Pattern differ spatially
- Solution: Spatial registration using SSD
- Problem : Articulations vary in length, and thus, in number of frames.
- Solution: Dynamic programming for temporal warping of sequences.
- Problem: Features should have compact representation.
- Solution: Principle Component Analysis.

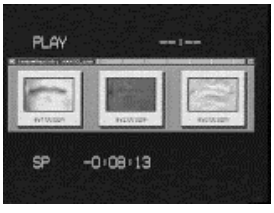
Results



I: “A” to “J” one speaker, 10 training seqs
II. “A” to “M”, one speaker, 10 training seqs
III. “A” to “Z”, ten speakers, two training seqs/letter/person







Hand Gesture Recognition

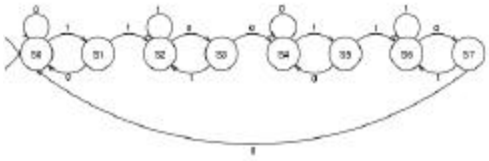
Jim Davis and Mubarak Shah,
"Visual Gesture Recognition", IEE
Proc. Vis Image Signal Processing,
October 1993.

Seven Gestures

Gesture Phases

- Hand fixed in the start position.
- Fingers or hand move smoothly to gesture position.
- Hand fixed in gesture position.
- Fingers or hand return smoothly to start position.

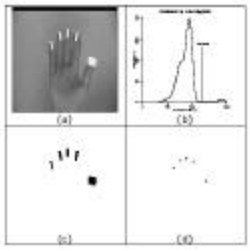
Finite State Machine



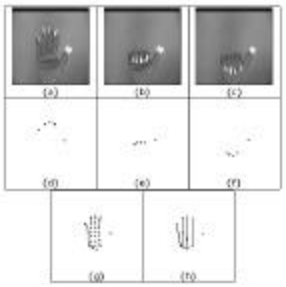
Main Steps

- Detect fingertips.
- Create fingertip trajectories using motion correspondence of fingertip points.
- Fit vectors and assign motion code to unknown gesture.
- Match

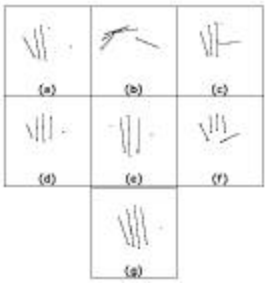
Detecting Fingertips



Vector Extraction



Vector Representation of Gestures



Results

Results

Run	Frames	L	R	U	D	T	G	S
1	200	√	√	√	√	√	√	√
2	250	√	√	√	√	√	√	√
3	250	√	√	√	X	√	√	√
4	250	√	√	√	√	√	√	√
5	300	√	√	√	√	√	√	√
6	300	√	√	√	√	√	√	√
7	300	√	√	√	√	√	√	√
8	300	√	√	√	√	√	√	√
9	300	√	√	√	√	*	*	*
10	300	√	√	√	√	√	√	√

L = Left, R = Right, U = Up, D = Down, T = Rotate, G = Grab, S = Stop, √ - Recognized, X - Not Recognized, * - Error in Sequence.

Action Recognition Using Temporal Templates

A. Bobick and J. Davis, "Action Recognition Using Temporal Templates", *Motion-Based Recognition*, ed: Mubarak Shah & Ramesh Jain, Kluwer Academic Publishers, 1997

Main Points

- Use seven Hu moments of MHI and MEI to recognize different exercises.
- Use seven views (-90 degrees to +90 degrees in increments of 30 degrees).
- For each exercise several samples are recorded using all seven views, and the mean and covariance matrices for the seven moments are computed as a model.
- During recognition, for an unknown exercise all seven moments are computed, and compared with all 18 exercises using Mahalanobis distance.
- The exercise with minimum distance is computed as the match.
- They present recognition results with one and two view sequences, as compared to seven view sequences used for model generation.

MEI and MHI

Motion-Energy Images (MEI)

$$E_t(x,y,t) = \bigcup_{i=0}^{t-1} D(x,y,t-i)$$ **Difference Pictures**

Motion History Images (MHI)

$$H_t(x,y,t) = \begin{cases} t & if D(x,y,t)=1 \\ \max(0, H_t(x,y,t-1)-1) & otherwise \end{cases}$$

Virtual 3-D Blackboard:

Finger Tracking with a Single Camera

Andrew Wu, Mubarak Shah & Niels Lobo, FG-2000

REU 1999

awu@uiuc.edu http://www.cs.ucf.edu/~vision

(go to REU99)

Summary of Algorithm

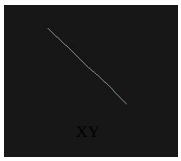
- Find head and arms using skin detection.
- Estimate locations of shoulder and elbow .
- Determine finger tip from arm outline, and track it.
- Compute 3 -D trajectory of finger tip using spherical kinematics.

Movie

3-D finger tracking
of a semi-circle

..\..\..\public_html\montage-semi.html

3-D Trajectory



Graphs of semi-circle movement, from varying viewpoints

Saddle Point movie

..\..\..\public_html\move-sad.html

Spiral movie

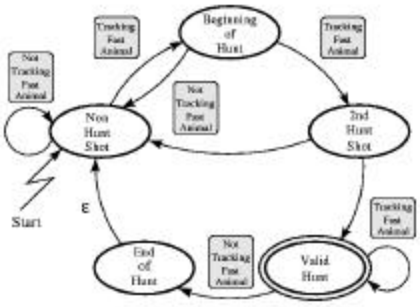
..\..\..\public_html\move-spire.html

Open GL Animation

A Framework for the Design of Visual Event Detectors

Niels Haering and Niels Da Vitoria Lobo, ACCV-2000

Hunt events



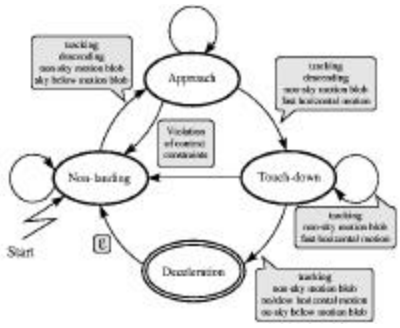
Hunts

Non-hunt

Hunt

Non-hunt

Landing Events



Landing Events

Non-landing

Approach

Touch-down

Deceleration

Non-landing

Papers

http://www.cs.ucf.edu/~vision

- Claudette Cedras and Mubarak Shah, "Motion-Based Recognition: A survey", Image and Vision Computing, March 1995.
- Jim Davis and Mubarak Shah, "Visual Gesture Recognition", IEE Proc. Vis Image Signal Processing, October 1993.
- Li Nan, Shawn Dettmer, and Mubarak Shah, "Visual Lipreading", Workshop on Face and Gesture Recognition, Zurich, 1995.
- Doug Ayers and Mubarak Shah, "Recognizing Human Activities In an Office Environment", Workshop on Applications of Computer Vision, October, 1998.

Book

- Mubarak Shah and Ramesh Jain, "**Motion-Based Recognition**", Kluwer Academic Publishers, 1997 ISBN 0-7923-4618-1.

Book

Mubarak Shah and Ramesh Jain, "**Motion-Based Recognition**", Kluwer Academic Publishers, 1997 ISBN 0-7923-4618-1.

Contents

- Mubarak Shah and Ramesh Jain, "Visual Recognition of Activities, Gestures, Facial Expressions and Speech: An Introduction and a Perspective"
- Human Activity Recognition
  - Y. Yacoob and L. Davis, "Estimating Image Motion Using Temporal Multi-Scale Models of Flow and Acceleration
  - A. Baumberg and D. Hogg, "Learning Deformable Models for Tracking the Human Body
  - S. Seitz and C. Dyer, "Cyclic Motion Analysis Using the Period Trace"

Contents (contd.)

- R. Pollana and R. Nelson, "Temporal Texture and Activity Recognition"
- A. Bobick and J. Davis, "Action Recognition Using Temporal Templates"
- N. Goddard, "Human Activity Recognition"
- K. Rohr, "Human Movement Analysis Based on Explicit Motion Models"

Contents (contd.)

- Gesture Recognition and Facial Expression Recognition
  - A. Bobick and A. Wilson, "State-Based Recognition of Gestures"
  - T. Starner and A. Pentland, "Real-Time American Sign Language Recognition from Video Using Hidden Markov Models"
  - M. Black , Y. Yacoob and S. Ju, "Recognizing Human Motion Using Parameterized Models of Optical Flow"

Contents (contd.)

- I. Essa and A. Pentland, "Facial Expression Recognition Using Image Motion"
- Lipreading
  - C. Bregler and S. Omohumdro, "Learning Visual Models for Lipreading"
  - A. Goldschen, O. Garcia and E. Petajan, "Continuous Automatic Speech Recognition by Lipreading"
  - N. Li, S. Dettmer and M. Shah, "Visually Recognizing Speech Using Eigensequences"

Part IV

Video Phones and MPEG-4

MPEG-1 & MPEG -2 Artifacts

- Blockiness
  - poor motion estimation
  - seen during dissolves and fades
- Mosquito Noises
  - edges of objects (high frequency DCT terms)
- Dirty Window
  - streaks or noise remain stationary while objects move
- Wavy Noise
  - seen during pans across crowds
  - coarsely quantized high frequency terms cause errors

Where MPEG-2 will fail?

- Motions which are not translation
  - zooms
  - rotations
  - non-rigid (smoke)
  - dissolves
- Others
  - shadows
  - scene cuts
  - changes in brightness

Video Compression At Low Bitrate

- The quality of block-based coding video (MPEG-1 & MPEG-2) at low bitrate, e.g., 10 kbps is very poor.
  - Decompressed images suffer from blockiness artifacts
  - Block matching does not account for rotation, scaling and shear

Model-Based Video Coding

Model-Based Compression

- Object-based
- Knowledge-based
- Semantic-based

Model-Based Compression

- Analysis
- Synthesis
- Coding

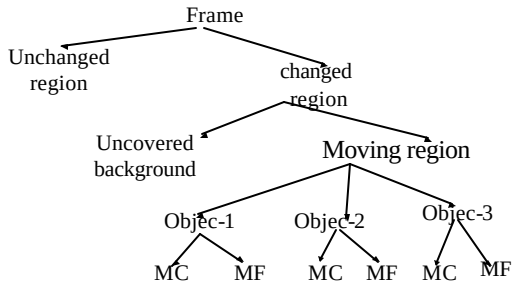
Video Compression

- MC/DCT
  - Source Model: translation motion only
  - Encoded Information: Motion vectors and color of blocks
- Object-Based
  - Source Model: moving unknown objects
    - translation only
    - affine
    - affine with triangular mesh
  - Encoded Information: Shape, motion, color of each moving object

Video Compression

- Knowledge-Based
  - Source Model: Moving known objects
  - Encoded Information: Shape, motion and color of known objects
- Semantic
  - Source Model: Facial Expressions
  - Encoded Information: Action units

Object-Based Coding



Contents

- Estimation using rigid+non-rigid motion model
- Making Faces (SIGGRAPH-98)
- Synthesizing Realistic Facial Expressions from Photographs (SIGGRAPH-98)
- MPEG-4

Model-Based Image Coding

- The transmitter and receiver both posses the same 3D face model and texture images.
- During the session, at the transmitter the facial motion parameters: global and local, are extracted.
- At the receiver the image is synthesized using estimated motion parameters.
- The difference between synthesized and actual image can be transmitted as residuals.

Face Model

- Candide model has 108 nodes, 184 polygons.
- Candide is a generic head and shoulder model. It needs to be conformed to a particular person's face.
- Cyberware scan gives head model consisting of 460,000 polygons.

Wireframe Model Fitting

- Fit orthographic projection of wireframe to the frontal view of speaker using Affine transformation.
- Locate four features in the image and the projection of model.
- Find parameters of Affine using least squares fit.
- Apply Affine to all vertices, and scale depth.

Synthesis

- Collapse initial wire frame onto the image to obtain a collection of triangles.
- Map observed texture in the first frame into respective triangles.
- Rotate and translate the initial wire frame according to global and local motion, and collapse onto the next frame.
- Map texture within each triangle from first frame to the next frame by interpolation.

Video Phones

Motion Estimation

Perspective Projection (optical flow)

$$u = f\left(\frac{V_1}{Z} + \Omega_2\right) - \frac{V_3}{Z}x - \Omega_3 y - \frac{\Omega_1}{f}xy + \frac{\Omega_2}{f}x^2$$

$$v = f\left(\frac{V_2}{Z} - \Omega_1\right) + \Omega_3 x - \frac{V_3}{Z}y + \frac{\Omega_2}{f}xy - \frac{\Omega_1}{f}y^2$$

Optical Flow Constraint Eq

$$f_x u + f_y v + f_t = 0$$

$$f_x\left(f\left(\frac{V_1}{Z}+\Omega_2\right)-\frac{V_3}{Z}x-\Omega_3 y-\frac{\Omega_1}{f}xy+\frac{\Omega_2}{f}x^2\right)+f_y$$
$$\left(f\left(\frac{V_2}{Z}-\Omega_1\right)+\Omega_3 x-\frac{V_3}{Z}y+\frac{\Omega_2}{f}xy-\frac{\Omega_1}{f}y^2\right)+f_t=0$$
$$\left(f_x\frac{f}{Z}\right)V_1+\left(f_y\frac{f}{Z}\right)V_2+\left(\frac{f}{Z}(f_x x-f_y y)\right)V_3+$$
$$\left(-f_x\frac{xy}{f}+f_y\frac{y^2}{f}-f_y f\right)\Omega_1+\left(f_x f+f_x\frac{x^2}{f}+f_y\frac{xy}{f}\right)\Omega_2+$$
$$(f_x y+f_y x)\Omega_3=-f_t$$

$$\left(f_x\frac{f}{Z}\right)V_1+\left(f_y\frac{f}{Z}\right)V_2+\left(\frac{f}{Z}(f_x x-f_y y)\right)V_3+$$
$$\left(-f_x\frac{xy}{f}+f_y\frac{y^2}{f}-f_y f\right)\Omega_1+\left(f_x f+f_x\frac{x^2}{f}+f_y\frac{xy}{f}\right)\Omega_2+$$
$$(f_x y+f_y x)\Omega_3=-f_t$$

$$\mathbf{Ax}=\mathbf{b}$$ Solve by Least Squares

$$\mathbf{x}=(V_1,V_2,V_3,\Omega_1,\Omega_2,\Omega_3)$$

$$A=\begin{bmatrix} \vdots & & & & & \\ \left(f_x\frac{f}{Z}\right) & \left(f_y\frac{f}{Z}\right) & \left(\frac{f}{Z}(f_x x-f_y y)\right) & \left(-f_x\frac{xy}{f}+f_y\frac{y^2}{f}-f_y f\right) & \left(f_x f+f_x\frac{x^2}{f}+f_y\frac{xy}{f}\right) & (f_x y+f_y x) \\ \vdots & & & & & \end{bmatrix}$$

Making Faces

Guenter et al
SIGGARPH'98

Making Faces

- System for capturing 3D geometry and color and shading (texture map).
- Six cameras capture 182 color dots on a face.
- 3D coordinates for each color dot are computed using pairs of images.
- Cyberware scanner is used to get dense wire frame model.

Making Faces

- Two models are related by a rigid transformation.
- Movement of each node in successive frames is computed by determining correspondence of nodes.

Synthesizing Realistic Facial Expressions from Photographs

Pighin et al
SIGGRAPH'98

Synthesizing Realistic Facial Expressions

- Select 13 feature points manually in face image corresponding to points in face model created with Alias.
- Estimate camera poses and deformed 3d model points.
- Use these deformed values to deform the remaining points on the mesh using interpolation.

Synthesizing Realistic Facial Expressions

- Introduce more points feature points (99) manually, and compute deformations as before by keeping the camera poses fixed.
- Use these deformed values to deform the remaining points on the mesh using interpolation as before.
- Extract texture.
- Create new expressions using morphing.

Show Video Clip.

MPEG-4

MPEG-4

- MPEG-4 is the international standard for true multimedia coding.
- MPEG-4 provides very low bitrate & error resilience for Internet and wireless.
- MPEG-4 can be carried in MPEG-2 systems layer.

MPEG-4

- 3-D facial animation
- Wavelet texture coding
- Mesh coding with texture mapping
- Media integration of text and graphics
- Text to speech synthesis

Applications of MPEG-4

- Multimedia broadcasting and presentations
- Virtual talking humans
- Advanced interpersonal communication systems
- Games
- Storytelling
- Language teaching
- Speech rehabilitation
- Teleshopping
- Telelearning

MPEG-4

- Real audio and video objects
- Synthetic audio and video
- Integration of Synthetic & Natural contents (Synthetic & Natural Hybrid Coding)

MPEG-4

- Traditional video coding is block-based.
- MPEG-4 provides object-based representation for better compression and functionalities.
- Objects are rendered after decoding object descriptions.
- Display of content layers can be selected at MPEG-4 terminal.

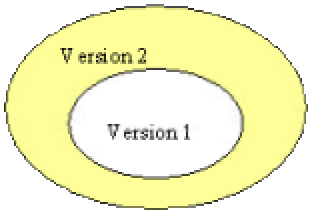
Scope & Features of MPEG-4

- Authors
  - reusability
  - flexibility
  - content owner rights
- Network providers
- End users

Media Objects

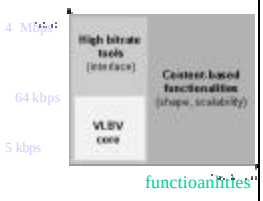
- Primitive Media Objects
- Compound Media Objects
- Examples
  - Still Images (e.g. fixed background)
  - Video objects (e.g., a talking person-without background)
  - Audio objects (e.g., the voice associated with that person)
  - etc

MPEG-4 Versions



MPEG-4

VLB Core
1. Low resolution CIF (360X288)
2. Low frame rate 15fps
3. High coding efficiency
4. Low complexity, low error
5. Random access
6. Fast forward/reverse



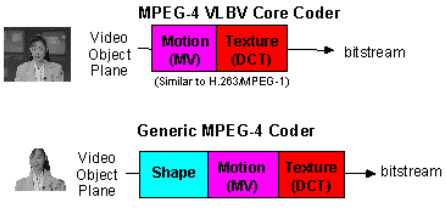
High Bitrate
1. Higher resolution
2. Higher frame rate
3. Interlaced video

Content-based functionalities
1. Interactivity
2. Flexible representation and Manipulation in the compressed Domain
3. Hybrid coding

User Interactions

- Client Side
  - content manipulation done at client terminal
    - changing position of an object
    - making it visible or invisible
    - changing the font size of text
- Server Side
  - requires back channel

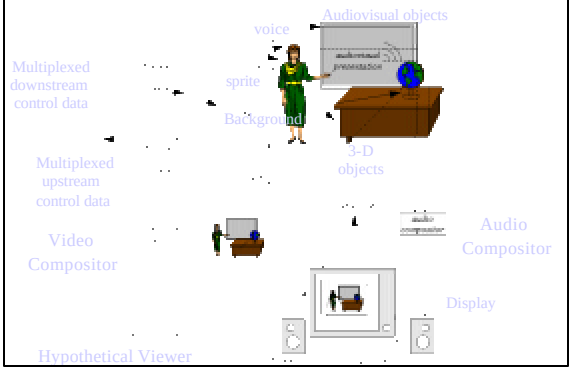
- Efficient representation of visual objects of arbitrary shape to support content-based functionalities
- Supports most functionalities of MPEG-1 and MPEG-2
  - rectangular sized images
  - several input formats
  - frame rates
  - bit rates
  - spatial, temporal and quality scalability



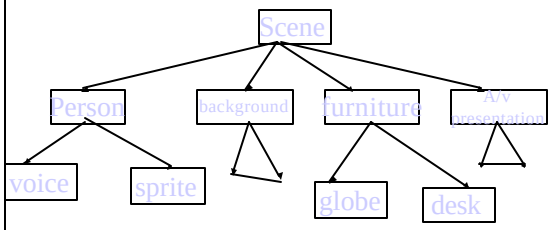
Object Composition

- Objects are organized in a scene graph.
- VRML based binary format BIF is used to specify scene graph.
- 2-D and 3-D objects, transforms and properties are specified.
- MPEG-4 allows objects to be transmitted once, and displayed repeatedly in the scene after transformations.

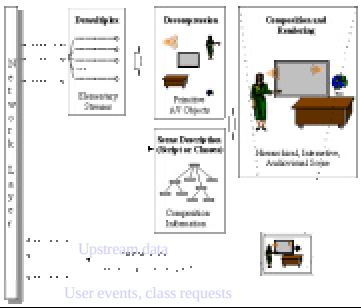
MPEG-4 Scene



Scene Graph



MPEG-4 Terminal



Textures, Images and Video

- Efficient compression of
  - images and video
  - textures for texture mapping on 2D and 3D meshes
  - implicit 2D meshes
  - time-varying geometry streams that animate meshes

2-D Animated Meshes

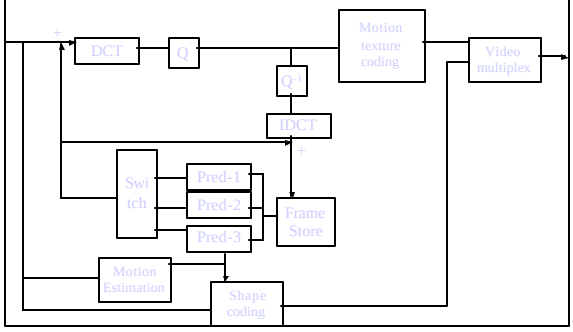
- A 2-D mesh is tessellation of a 2 -D planar region into triangles.
- Dynamic meshes contain mesh geometry and motion.
- 2-D meshes can be used for texture mapping. Three nodes of triangle defines affine motion.

2-D Mesh Modeling

MPEG-4 Video and Image Coding Scheme

- Shape coding and motion compensation
- DCT-based texture coding
  - standard 8x8 and shape adapted DCT
- Motion compensation
  - local block based (8x8 or 16x16)
  - global (affine) for sprites

MPEG-4 Video Coder

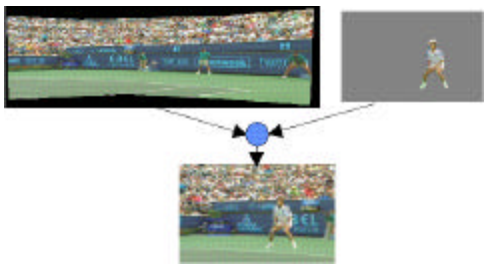


Sprite Panorama

- First compute static “sprite” or “mosaic”
- Then transmit 8 or 6 global motion (camera) parameters for each frame to reconstruct the fame from the “sprite”
- Moving foreground is transmitted separately as an arbitrary-shape video object.

Steps in Sprite Construction

- Incremental mosaic construction
- Incremental residual estimation
- Computation of significance measures on the residuals
- Spatial coding and decoding
- Visit http://www.wisdom.weizmann.ac.il/~irani/abstracts/mosaics.html



Other Objects

- Text and graphics
- Talking synthetic head and associated text
- Synthetic sound

Face and Body Animation

- Face animation is in MPEG-4 version 1.
- Body animation is in MPEG-4 version 2.
- Face animation parameters displace feature points from neutral position.
- Body animation parameters are joint angles.
- Face and body animation parameter sequences are compressed to low bit rate.
- Facial expressions: joy, sadness, anger, fear, disgust and surprise.
- Visemes

Face Model

- Face model (3D) specified in VRML, can be downloaded to the terminal with MPEG-4

Neutral Face

- Face is gazing in the Z direction
- Face axes parallel to the world axes
- Pupil is 1/3 of iris in diameter
- Eyelids are tangent to the iris
- Upper and lower teeth are touching and mouth is closed
- Tongue is flat, and the tip of tongue is touching the boundary between upper and lower teeth

Face Node

- FAP (Facial Animation Parameters)
  - FAPs allow to animate 3-D facial node at the receiver. Animation of key feature points and reproduction of visemes & expressions
- Face Definition Parameters (FDP)
  - FDP allow to configure facial model to be used at the receiver, either by sending a new model, or by adapting a previously available model. Sent only once.
- Face Interpolation Table (FIT)
  - FIT allow to define interpolation rules for FAPs that have to be interpolated at the receiver. The 3-D model is animated using FAPs sent and FAPs interpolated.
- Face Animation Table (FAT)
  - It specifies for each selected FAP the set of vertices to be affected in a new downloaded model, as well as the way they are affected. E.g. FAP 'open jaw', then table defines what that means in terms of moving the feature points.

Facial Animation Parameters (FAPS)

- 2 eyeball and 3 head rotations are represented using Euler angles
- Each FAP is expressed as a fraction of neutral face mouth width, mouth-nose distance, eye separation, or iris diameter.

FAP Groups

Group	FAPS
Visemes & expressions	2
jaw, chin, inner lower-lip, corner lip, mid-lip	16
eyeballs, pupils, eyelids	12
eyebrow	8
cheeks	4
tongue	5
head rotation	3
outer lip position	10
nose	4
ears	4

FAPS

- 31: raise_l_I_eyebrow (vertical displacement of left inner eyebrow)
- 32: raise_r_I_eyebrow(vertical displacement of right inner eyebrow)
- 33: raise_l_m_eyebrow(vertical displacement of left middle eyebrow)
- 34: raise_r_m_eyebrow(vertical displacement of right middle eyebrow)
- 35:

FAP Data

- Synthetically generated
- Extracted by analysis
  - Real-time (video phones)
  - Off-line (story telling)
  - Fully automatic (video phones)
  - Human-guided (teleshopping & gaming)

FAPs Masking Scheme Options

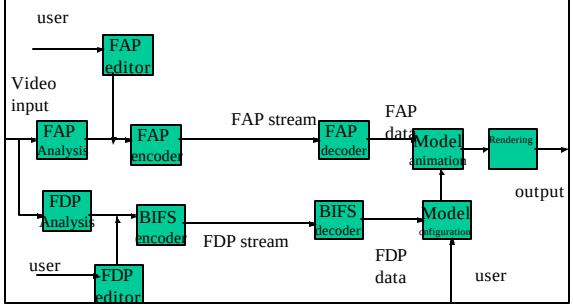
- No FAPs are coded for the corresponding group
- A mask is given indicating which FAPs in the corresponding group are coded. FAPs not coded, retain their previous values
- A mask is given indicating which FAPs in the corresponding group are coded. The decoder should interpolate FAPs not selected by the group mask.
- All FAPs in the group are coded.

Four Cases of FDP

- No FDP data is sent, residing 3-D model at the receiver is used for animation
- Feature points (calibrate the model) are sent
- Feature points and texture are sent
- Facial Animation Tables (FATs) and 3-D model are sent
  - FAT specify the FAP behavior (which and how the new model vertices should be moved for each FAP)

- It is difficult for the sender to know precisely the appearance of the synthesized result at the receiver since a large number of models may be used.

3-D Facial Animation System



FAPs

- Speech recognition can use FAPs to increase recognition rate.
- FAPs can be used to animate face models by text to speech systems
- In HCI FAPs can be used to communicate speech, emotions, etc, in particular in noisy environment.

Visemes and Expressions

- For each frame a weighted combination of two visemes and two facial expressions
- After FAPs are applied the decoder can interpret effect of visemes and expressions
- Definitions of visemes and expressions using FAPs can be downloaded

Phonemes and Visemes

- 56 phonemes
  - 37 consonants
  - 19 vowels/diphthongs
- 56 phonemes can be mapped to 35 visemes
- A triseme is made up of three visemes to capture co-articulations

56 Phonemes

Phone	Example	Phone	Example	Phone	Example	Phone	Example
aa	cot	ow	boat	g	gag	q	glottal stop
ac	bat	oy	boy	gcl	g-closure	r	red
ah	butt	oy	boy	hh	hay	s	sis
ao	about	uh	book	hv	Leheigh	sh	shoe
aw	bough	uw	boot	jh	judge	t	tot
ax	the	ux	beauty	k	kick	tcl	t-closure
axr	diner	b	bob	kcl	k-closur	th	thief
ay	bite	bcl	b-closure	l	led	v	very
eh	bet	ch	church	m	mom	w	wet
er	bird	d	dad	n	non	y	yet
ey	bait	dcl	d-closure	ng	sing	z	zoo
ih	bit	dh	they	nx	flapped -n	zh	measure
ix	roses	dx	butter	p	pop	epi	epithetic closure
iy	beat	en	button	pcl	p-closur	h#	silence
		f	fief				

Phone to Viseme Mapping

Vowel/Diphthongs

aa	ae, eh
ah	ao
aw	ax,ih,iy
axr	ay
fr	ey
ix	ow
oy	uh
uw	ux

Consonants

b,p	bcl,m,pcl	ch
dh,epi	dx,nx,q	f,v
en	hh	hv
jh	ng	r
s,sh,z	th	w
y	zh	h#
d,dcl,g,gc,k,kcl,l,n,t,tcl		

MPEG-4 Visems

Viseme_select	phonemes	example
0	none	na
1	p, b, m	put, bed, mill
2	f, v	far, voice
3	T, D	think, that
4	t, d	tip, doll
5	k, g	call, gas
6	tS, dZ, S	chair, join, she
7	s, z	sir, zeal
8	n, l	lot, not
9	r	red
10	A:	car
11	e	bed
12	I	tip
13	O	top
14	U	book

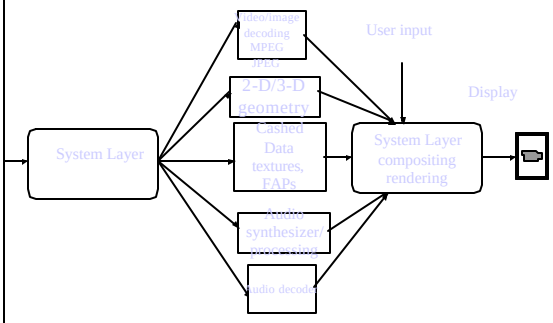
Facial Expressions

- Joy
  - The eyebrows are relaxed. The mouth is open, and mouth corners pulled back toward ears.
- Sadness
  - The inner eyebrows are bent upward. The eyes are slightly closed. The mouth is relaxed.
- Anger
  - The inner eyebrows are pulled downward and together. The eyes are wide open. The lips are pressed against each other or opened to expose teeth.

Facial Expressions

- Fear
  - The eyebrows are raised and pulled together. The inner eyebrows are bent upward. The eyes are tense and alert.
- Disgust
  - The eyebrows and eyelids are relaxed. The upper lip is raised and curled, often asymmetrically.
- Surprise
  - The eyebrows are raised. The upper eyelids are wide open, the lower relaxed. The jaw is open.

MPEG-4 Decoder



MPEG-4

- Go to http://www.cselt.it/mpeg

Conclusion

- Video Computing
  - Video Understanding
  - Video Tracking
  - Video Mosaics
  - Video Phones
  - Video Synthesis
  - Video Compression