Lecture-13

Model-base Video Compression

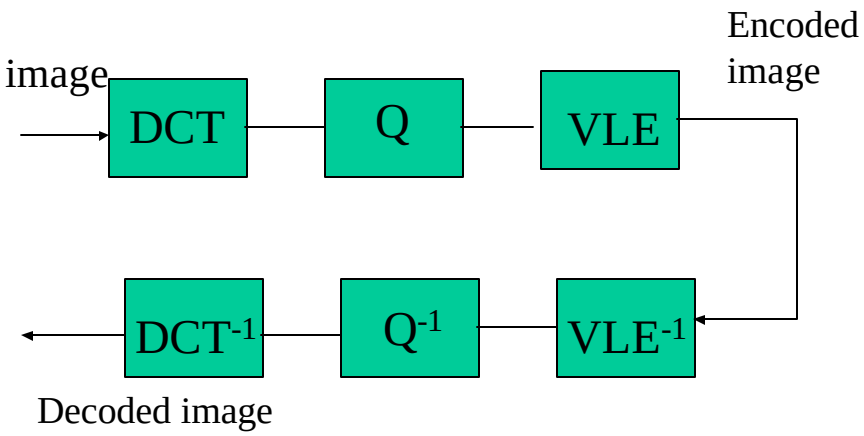
JPEG Baseline Coding

- Divide image into blocks of size 8X8.
- Level shift all 64 pixels values in each block by subtracting $2^{n-1,}$ (where 2^n is the maximum number of gray levels).
- Compute 2D DCT of a block.
- Quantize DCT coefficients using quantization table.

JPEG Baseline Coding

- Zig-zag scan the quantized DCT coefficients to form 1-D sequence.
- Code 1-D sequence (AC and DC) using JPEG Huffman variable length codes.

JPEG BLOCK DIAGRAM



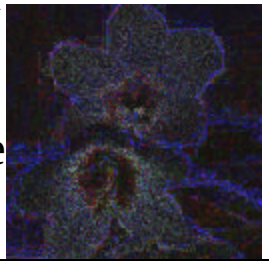
JPEG

Original 64K

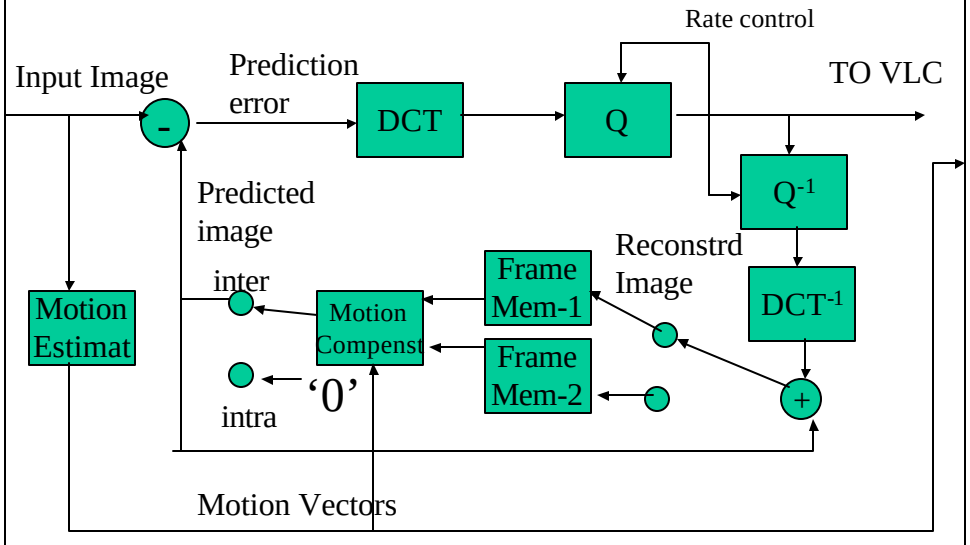
13K

5K

Difference



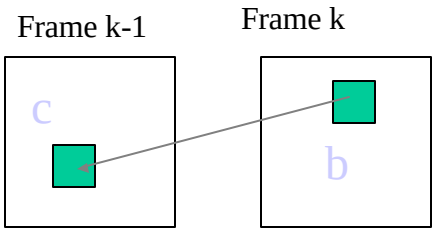
MPEG-1 Encoder



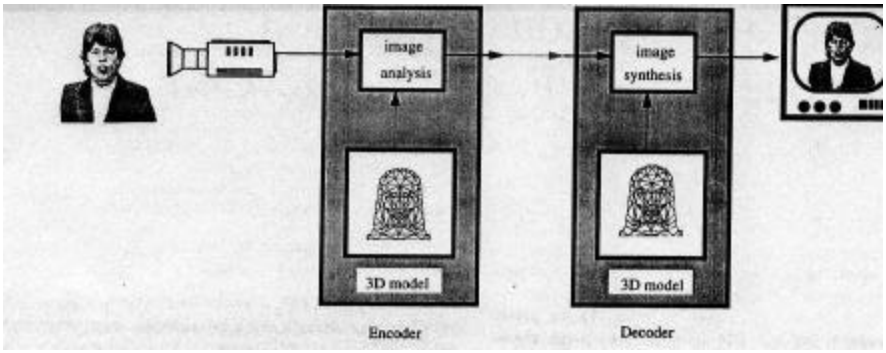
Motion Prediction

$$b' = c'$$

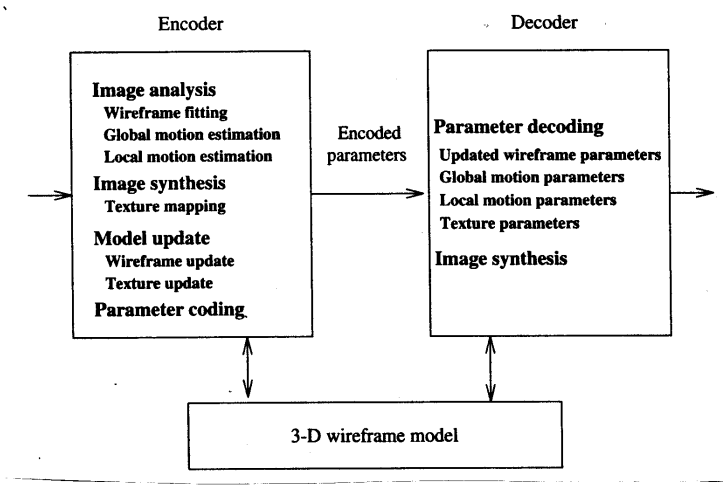
$$Error = b - b'$$



Model-Based Image Coding



Model-Based Image Coding



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.

Candide Model

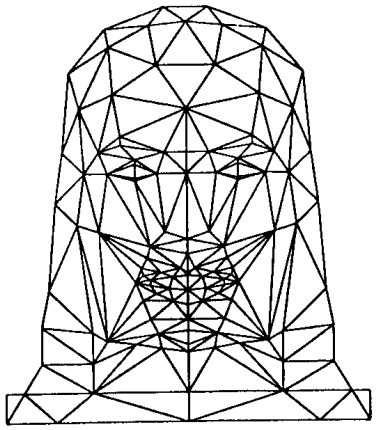


Fig. 2. Wire-frame model of the face.

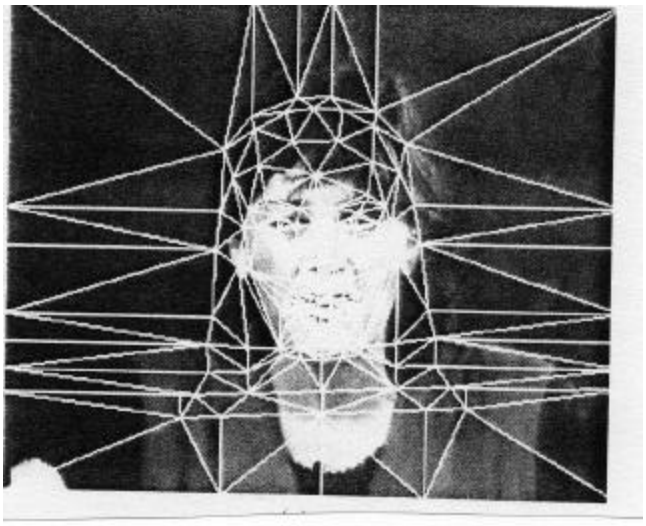
Face Model

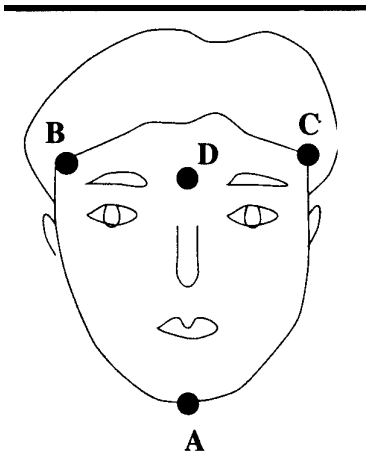
- Candide model has 108 nodes, 184 polygons.
- Candide is a generic head and shoulder model. It needs to be conformed to a particular person's face.
- Cyberware scan gives head model consisting of 460,000 polygons.

Wireframe Model Fitting

- Fit orthographic projection of wireframe to the frontal view of speaker using Affine transformation.
  - Locate three to four features in the image and the projection of a model.
  - Find parameters of Affine transformation using least squares fit.
  - Apply Affine to all vertices, and scale $\sqrt{(a_1^2 + a_4^2)^2 / 2}$ depth.

$$\begin{bmatrix} x_2 \\ y_2 \end{bmatrix} = \begin{bmatrix} a_1 & a_2 \\ a_3 & a_4 \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}$$

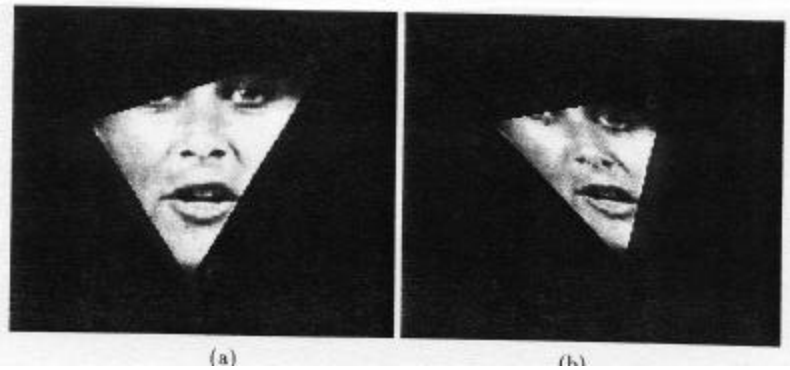




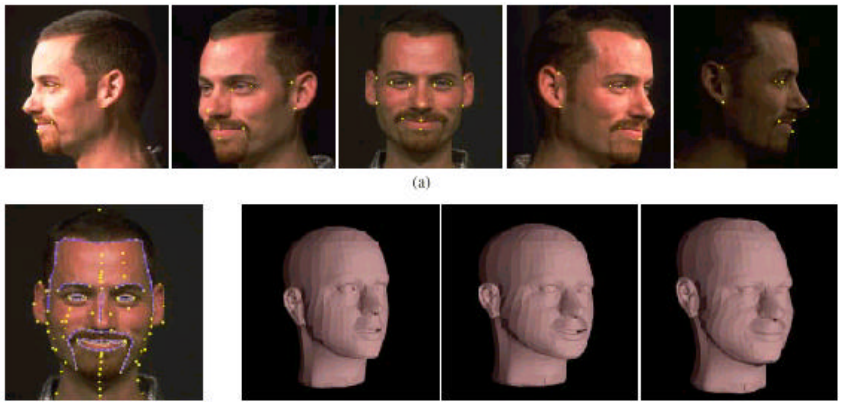
Synthesis

- 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.

Texture Mapping



Synthesizing Realistic Facial Expressions from Photographs: Pighin et al SIGGRAPH'98



3D Rigid Transformation

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

Camera coordinates

Wireframe coordinates

$$x_i'^k = f^k \frac{X_i'^k}{Z_i'^k}, y_i'^k = f^k \frac{Y_i'^k}{Z_i'^k}$$ perspective

K=camera no.

3D Rigid Transformation

$$x_i'^k = f^k \frac{X_i'^k}{Z_i'^k}, y_i'^k = f^k \frac{Y_i'^k}{Z_i'^k}$$

$$x_i'^k = f_k \frac{r_{11}^k X_i + r_{12}^k Y_i + r_{13}^k Z_i + T_X^k}{r_{31}^k X_i + r_{32}^k Y_i + r_{33}^k Z_i + T_Z^k}$$

$$y_i'^k = f_k \frac{r_{21}^k X_i + r_{22}^k Y_i + r_{23}^k Z_i + T_Y^k}{r_{31}^k X_i + r_{32}^k Y_i + r_{33}^k Z_i + T_Z^k}$$

Model Fitting

$$x_i'^k = f_k \frac{\mathbf{r}_x^{\mathbf{k}} \mathbf{p}_{\mathbf{i}} + T_X^k}{\mathbf{r}_z^{\mathbf{k}} \mathbf{p}_{\mathbf{i}} + T_Z^k}$$

$$y_i'^k = f_k \frac{\mathbf{r}_y^{\mathbf{k}} \mathbf{p}_{\mathbf{i}} + T_Y^k}{\mathbf{r}_z^{\mathbf{k}} \mathbf{p}_{\mathbf{i}} + T_Z^k}$$

Model Fitting

$$x_i'^k = f_k \frac{\mathbf{r}_x^{\mathbf{k}} \mathbf{p}_{\mathbf{i}} + T_X^k}{\mathbf{r}_z^{\mathbf{k}} \mathbf{p}_{\mathbf{i}} + T_Z^k}$$

$$y_i'^k = f_k \frac{\mathbf{r}_y^{\mathbf{k}} \mathbf{p}_{\mathbf{i}} + T_Y^k}{\mathbf{r}_z^{\mathbf{k}} \mathbf{p}_{\mathbf{i}} + T_Z^k} \qquad h^k = \frac{1}{T_Z^k}, s^k = f^k h^k$$

$$x_i'^k = s_k \frac{\mathbf{r}_x^{\mathbf{k}} \mathbf{p}_{\mathbf{i}} + T_X^k}{1 + h^k \mathbf{r}_z^{\mathbf{k}} \mathbf{p}_{\mathbf{i}}}$$

$$y_i'^k = s_k \frac{\mathbf{r}_y^{\mathbf{k}} \mathbf{p}_{\mathbf{i}} + T_Y^k}{1 + h^k \mathbf{r}_z^{\mathbf{k}} \mathbf{p}_{\mathbf{i}}}$$

Model Fitting

$$x_i'^k = s_k \frac{\mathbf{r}_x^{\mathbf{k}} \mathbf{p}_{\mathbf{i}} + T_X^k}{1 + h^k \mathbf{r}_z^{\mathbf{k}} \mathbf{p}_{\mathbf{i}}}$$

$$y_i'^k = s_k \frac{\mathbf{r}_y^{\mathbf{k}} \mathbf{p}_{\mathbf{i}} + T_Y^k}{1 + h^k \mathbf{r}_z^{\mathbf{k}} \mathbf{p}_{\mathbf{i}}} \quad w_i^k = (1 + h^k (\mathbf{r}_z^k . \mathbf{p}_i))^{-1}$$

$$w_i^k (x_i'^k + x_i'^k h^k (\mathbf{r}_{\mathbf{z}}^{\mathbf{k}} . \mathbf{p}_{\mathbf{i}}) - s^k (\mathbf{r}_X^{\mathbf{k}} . \mathbf{p}_{\mathbf{i}} + T_X^k)) = 0$$

$$w_i^k (y_i'^k + y_i'^k h^k (\mathbf{r}_{\mathbf{z}}^{\mathbf{k}} . \mathbf{p}_{\mathbf{i}}) - s^k (\mathbf{r}_Y^{\mathbf{k}} . \mathbf{p}_{\mathbf{i}} + T_Y^k)) = 0$$

Model Fitting

- Solve for unknowns in five steps:

$$s^k; \mathbf{p}_i; \mathbf{R}^k; T_X^k, T_Y^k; h^k$$

  - Use linear least squares fit.
  - When solving for an unknown, assume other parameters are known.

Least Squares Fit

$$a_j.x - b_j = 0$$

$$\sum_j (a_j.x - b_j)^2$$

$$\sum_j (a_j a_j^T)x = \sum_j b_j a_j$$

Update for p

$$a_j.x - b_j = 0$$

$$\sum_j (a_j.x - b_j)^2$$

$$\sum_j (a_j a_j^T)x = \sum_j b_j a_j$$

$$w_i^k(x_i'^k + x_i'^k h^k(\mathbf{r_z^k}.\mathbf{p_i}) - s^k(\mathbf{r}_X^k.\mathbf{p_i} + T_X^k)) = 0$$

$$w_i^k(y_i'^k + y_i'^k h^k(\mathbf{r_z^k}.\mathbf{p_i}) - s^k(\mathbf{r}_Y^k.\mathbf{p_i} + T_Y^k)) = 0$$

Update for s^k

$$a_{2k+0} = w_i^k(r_x^k.p_i + t_x^k) \qquad b_{2k+0} = w_i^k(x_i^k + x_i^k \boldsymbol{h}^k(r_z^k.p_i))$$

$$a_{2k+1} = w_i^k(r_y^k.p_i + t_y^k) \qquad b_{2k+1} = w_i^k(y_i^k + y_i^k \boldsymbol{h}^k(r_z^k.p_i))$$

$$x_i' + x_i' h(\mathbf{r}_z.\mathbf{p}_i) - s(\mathbf{r}_x.\mathbf{p}_i + T_x) = 0$$
$$y_i' + y_i' h(\mathbf{r}_z.\mathbf{p}_i) - s(\mathbf{r}_y.\mathbf{p}_i + T_y) = 0$$

Solving for T_x and T_y

$$sT_x = x_i' + x_i' h(\mathbf{r}_z.\mathbf{p}_i) - s(\mathbf{r}_x.\mathbf{p}_i)$$
$$a_0 = s, b_0 = x_i' + x_i' h(\mathbf{r}_z.\mathbf{p}_i) - s(\mathbf{r}_x.\mathbf{p}_i)$$

$$sT_y = y_i' + y_i' h(\mathbf{r}_z.\mathbf{p}_i) - s(\mathbf{r}_y.\mathbf{p}_i)$$
$$a_1 = s, b_1 = y_i' + y_i' h(\mathbf{r}_z.\mathbf{p}_i) - s(\mathbf{r}_y.\mathbf{p}_i)$$

$$x_i' + x_i' h(\mathbf{r}_z.\mathbf{p}_i) - s(\mathbf{r}_x.\mathbf{p}_i + T_x) = 0$$
$$y_i' + y_i' h(\mathbf{r}_z.\mathbf{p}_i) - s(\mathbf{r}_y.\mathbf{p}_i + T_y) = 0$$

Solving for h

$$a_0 = x_i'(\mathbf{r}_z.\mathbf{p}_i), b_0 = s(\mathbf{r}_x.\mathbf{p}_i + T_x) - x_i'$$
$$a_1 = y_i'(\mathbf{r}_z.\mathbf{p}_i), b_1 = s(\mathbf{r}_y.\mathbf{p}_i + T_y) - y'$$

Rotation Around an Arbitrary Axis (Rodriguez's Formula)

$$V' = V + \sin q \; n \times V + (1 - \cos q)\, n \times (n \times V)$$

$$V' = R(n,q)V$$

$$R(n,\boldsymbol{q}) = I + \sin \boldsymbol{q}\; X(n) + (1 - \cos \boldsymbol{q}) X^2(n)$$

$$X(n) = \begin{bmatrix} 0 & -n_z & n_x \\ n_z & 0 & -n_x \\ -n_y & n_x & 0 \end{bmatrix}$$

Rotation Around an Arbitrary Axis (Rodriguez's Formula)

$$r = \| r \| \frac{r}{\| r |} = qn$$

$$R(r,\boldsymbol{q}) = I + \sin \boldsymbol{q} \frac{X(r)}{\| r \|} + (1 - \cos \boldsymbol{q}) \frac{X^2(r)}{\| r \|^2}$$

$$X(r) = \begin{bmatrix} 0 & -r_z & r_x \\ r_z & 0 & -r_x \\ -r_y & r_x & 0 \end{bmatrix}$$

$$\tilde{R} \approx I + X(m) \qquad m = qn = (m_x, m_y, m_z)$$

Solving for rotation:

$$w_i^k (x_i'^k + x_i'^k h^k (\mathbf{r_z^k}.\mathbf{p_i}) - s^k (\mathbf{r_X^k}.\mathbf{p_i} + T_X^k)) = 0$$
$$w_i^k (y_i'^k + y_i'^k h^k (\mathbf{r_z^k}.\mathbf{p_i}) - s^k (\mathbf{r_Y^k}.\mathbf{p_i} + T_Y^k)) = 0$$

$$w_i^k (x_i'^k + x_i'^k h(\tilde{r}_z^k.q_i) - s^k (\tilde{r}_x^k.q_i + T_x^k)) = 0$$
$$w_i^k (y_i'^k + y_i'^k h(\tilde{r}_z^k.q_i) - s^k (\tilde{r}_y^k.q_i + T_y^k)) = 0$$

$$x_i' + x_i' h(-m_y q_x + m_x q_y + q_z) - s(q_x - m_z q_y + m_y q_z + T_x) = 0$$

$$y_i' + y_i' h(-m_y q_x + m_x q_y + q_z) - s(m_z q_x + q_y - m_x q_z + T_y) = 0$$

$$\tilde{r}_z^k = (-m_y, m_x, 1)$$
$$\tilde{r}_x^k = (1, -m_z, m_y)$$
$$\tilde{r}_y^k = (m_z, 1, -m_x)$$

$$q_i = R^{it} p_i$$

$$R^{it+1} \leftarrow \tilde{R} R^{it}$$

$$x_i' + x_i' h(-m_y q_x + m_x q_y + q_z) - s(q_x - m_z q_y + m_y q_z + T_x) = 0$$
$$y_i' + y_i' h(-m_y q_x + m_x q_y + q_z) - s(m_z q_x + q_y - m_x q_z + T_y) = 0$$

$$\begin{bmatrix} x'hq_y & -x'hq_x - sq_z & sq_y \end{bmatrix} \begin{bmatrix} m_x \\ m_y \\ m_z \end{bmatrix} = \begin{bmatrix} sT_x - x' - x'hq_z + sq_x \end{bmatrix}$$

$$\begin{bmatrix} y'hq_y + sq_z & -y'hq_x & -sq_x \end{bmatrix} \begin{bmatrix} m_x \\ m_y \\ m_z \end{bmatrix} = \begin{bmatrix} sT_y - y' - y'hq_z + sq_y \end{bmatrix}$$

$$a_0 = \begin{bmatrix} x'\boldsymbol{h}q_y & -x'\boldsymbol{h}q_x + sq_z & sq_y \end{bmatrix}, b_0 = \begin{bmatrix} sT_x - x' - x'\boldsymbol{h}q_z + sq_x \end{bmatrix}$$

$$a_1 = \begin{bmatrix} y'\boldsymbol{h}q_y - sq_z & -y'\boldsymbol{h}q_x & sq_x \end{bmatrix}, b_1 = \begin{bmatrix} sT_y - y' - y'\boldsymbol{h}q_z + sq_y \end{bmatrix}$$

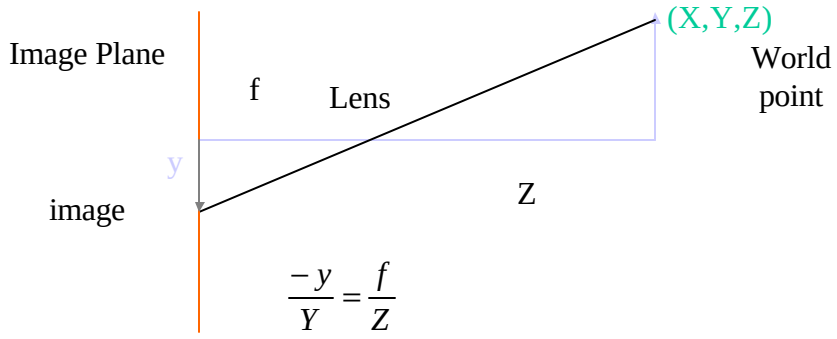
$$\sum_j (a_j a_j^T) x = \sum_j b_j a_j$$

Show the results

Video Phones

Motion Estimation

Perspective Projection



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

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

Perspective Projection (arbitrary flow)

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

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

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

$$\dot{X} = \Omega_2 Z - \Omega_3 Y + V_1$$
$$\dot{Y} = \Omega_3 X - \Omega_1 Z + V_2$$
$$\dot{Z} = \Omega_1 Y - \Omega_2 X + V_3$$

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

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

Perspective Projection (optical flow)

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

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

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

Optical Flow Constraint Eq

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

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

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

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

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

$$\mathbf{Ax} = \mathbf{b}$$

$$\begin{bmatrix} \vdots & & & & & \\ (f_x \frac{f}{Z}) & (f_y \frac{f}{Z}) & (\frac{f}{Z}(f_x x - f_y y) & (-f_x \frac{xy}{f} + f_y \frac{y^2}{f} - f_y f) & (f_x f + f_x \frac{x^2}{f} + f_y \frac{xy}{f}) & (f_x y + f_y x) \\ \vdots & & & & & \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \\ V_3 \\ \Omega_1 \\ \Omega_2 \\ \Omega_3 \end{bmatrix} = \begin{bmatrix} \vdots \\ f_t \\ \vdots \end{bmatrix}$$

Li et al, IEEE PAMI, June 1993

Estimation Using Flexible Wireframe Model

Digital Video Processing,
M. Tekalp, Prentice Hall, 1995.

Main Points

- Model photometric effects
- Simultaneously compute 3D motion and adapt the wireframe model.

Generalized Optical Flow Constraint

Lambertian Model

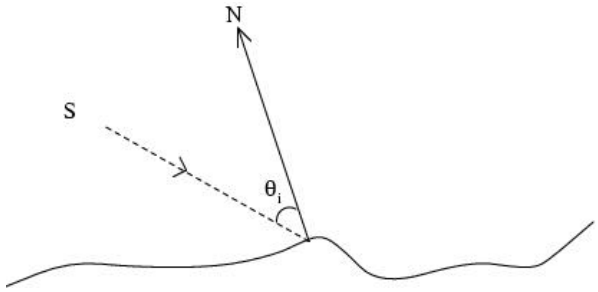
$$f(x,y,t) = rN(t).L$$

$$\frac{df(x,y,t)}{dt} = rL.\frac{dN}{dt}$$

Albedo
Surface Normal
(-p,-q,1)

$$f_x u + f_y v + f_t = rL.\frac{dN}{dt}$$

Lambertian Model



S=L, light source

$$f(x,y) = n.L = (n_x, n_y, n_z).(l_x, l_y, l_z)$$

$$f(x,y) = n.L = (\frac{1}{\sqrt{p^2 + q^2 + 1}}(-p,-q,1)).(l_x, l_y, l_z)$$

Sphere

$$z = \sqrt{\left(R^2 - x^2 - y^2\right)}$$

$$p = \frac{\partial z}{\partial x} = -\frac{x}{z}$$

$$q = \frac{\partial z}{\partial y} = -\frac{y}{z}$$

$$(n_x, n_y, n_z) = \frac{1}{R}(x, y, z)$$

Sphere

Vase

(1, 0, 1) (-1, 1, 1) (-1,-1, 1)

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

Orthographic Projection

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

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

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

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

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

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

Optical flow equation

$$f_x(\Omega_2 Z - \Omega_3 y + V_1) + f_y(\Omega_3 x - \Omega_1 Z + V_2) + f_t = \rho L . \frac{dN}{dt}$$

$$f_x(\Omega_2 Z - \Omega_3 y + V_1) + f_y(\Omega_3 x - \Omega_1 Z + V_2) + f_t = \rho L . \left[\frac{(-p', -q', 1)^T}{\sqrt{p'^2 + q'^2 + 1}} - \frac{(-p, -q, 1)^T}{\sqrt{p^2 + q^2 + 1}} \right]$$

Equation 24.8, page 473.

Error Function

$$E = \sum_i \sum_{(x,y)eithpatch} e_i^2$$

$$p_i x_1^{(ij)} + q_i x_2^{(ij)} + c_i = p_j x_1^{(ij)} + q_j x_2^{(ij)} + c_j$$

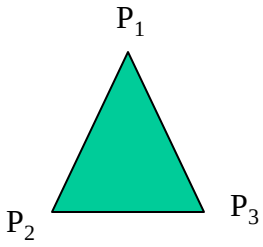
constraint

$$e_i(x, y) = f_x(\Omega_3 y - \Omega_2(p_i x + q_1 y + c_i) + V_1)$$
$$+ f_y(-\Omega_3 x + \Omega_1(p_i x + q_1 y + c_i) + V_2) + f_t$$
$$- r(L_1, L_2, L_3).\left(\frac{\left(-\frac{-\Omega_2 + p_i}{1+\Omega_2 p_i}, \frac{-\Omega_1 + q_i}{1-\Omega_1 q_i}\right)}{\left(\left(\frac{-\Omega_2 + p_i}{1+\Omega_2 p_i}\right)^2 + \left(\frac{-\Omega_1 + q_i}{1-\Omega_1 q_i}\right)^2 + 1\right)^{1/2}}\right) -$$
$$\frac{(-p_i, -q_i, 1)}{(p_i^2 + q_i^2 + 1)^{1/2}}$$

Equation of a Planar Patch

$$P_1^{(i)} = (X_1^{(i)}, Y_1^{(i)}, Z_1^{(i)})$$
$$P_2^{(i)} = (X_2^{(i)}, Y_2^{(i)}, Z_2^{(i)})$$
$$P_3^{(i)} = (X_3^{(i)}, Y_3^{(i)}, Z_3^{(i)})$$

$$P^{(i)} = (X^{(i)}, Y^{(i)}, Z^{(i)})$$



$$\overline{P^{(i)}P_1^{(i)}} \cdot \left(\overline{P_2^{(i)}P_1^{(i)}} \times \overline{P_3^{(i)}P_1^{(i)}}\right) = 0$$

Equation of a Planar Patch

$$Z^{(i)} = p_i X^{(i)} + q_i Y^{(i)} + c_i$$

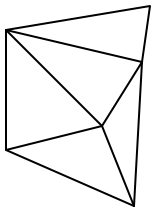
$$p_i = -\frac{(Y_2^{(i)} - Y_1^{(i)})(Z_3^{(i)} - Z_1^{(i)}) - (Z_2^{(i)} - Z_1^{(i)})(Y_3^{(i)} - Y_1^{(i)})}{(X_2^{(i)} - X_1^{(i)})(Y_3^{(i)} - Y_1^{(i)}) - (Y_2^{(i)} - Y_1^{(i)})(X_3^{(i)} - X_1^{(i)})}$$

$$q_i = -\frac{(Z_2^{(i)} - Z_1^{(i)})(X_3^{(i)} - X_1^{(i)}) - (X_2^{(i)} - X_1^{(i)})(Z_3^{(i)} - Z_1^{(i)})}{(X_2^{(i)} - X_1^{(i)})(Y_3^{(i)} - Y_1^{(i)}) - (Y_2^{(i)} - Y_1^{(i)})(X_3^{(i)} - X_1^{(i)})}$$

$$c_i = Z_1^{(i)} + X_1^{(i)} \frac{(Y_2^{(i)} - Y_1^{(i)})(Z_3^{(i)} - Z_1^{(i)}) - (Z_2^{(i)} - Z_1^{(i)})(Y_3^{(i)} - Y_1^{(i)})}{(X_2^{(i)} - X_1^{(i)})(Y_3^{(i)} - Y_1^{(i)}) - (Y_2^{(i)} - Y_1^{(i)})(X_3^{(i)} - X_1^{(i)})} +$$
$$Y_1^{(i)} \frac{(Z_2^{(i)} - Z_1^{(i)})(X_3^{(i)} - X_1^{(i)}) - (X_2^{(i)} - X_1^{(i)})(Z_3^{(i)} - Z_1^{(i)})}{(X_2^{(i)} - X_1^{(i)})(Y_3^{(i)} - Y_1^{(i)}) - (Y_2^{(i)} - Y_1^{(i)})(X_3^{(i)} - X_1^{(i)})}$$

Structure of Wireframe Model

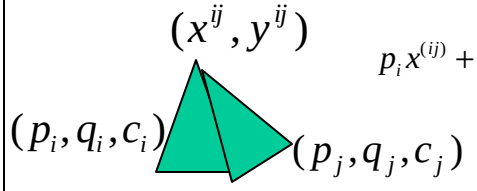
- Each triangular patch is either surrounded by two (if it is on the boundary of the wireframe) or three other triangles.



Neighboring patches must intersect at a straight line.

$$Z^{(i)} = p_i X^{(i)} + q_i Y^{(i)} + c_i$$

$$p_i x^{(ij)} + q_i y^{(ij)} + c_i = p_j x^{(ij)} + q_j y^{(ij)} + c_j$$



Main Points of Algorithm

- Stochastic relaxation.
- In each iteration visit all patches in a sequential order.
  - If, at present iteration none of neighboring patches of *i* have been visited yet, then p_i, q_i, c_i are all independently perturbed.
  - If, only one of the neighbor, *j*, has been visited, then two parameters, say p_i, q_i, are independent and perturbed. The dependent variable c_i is calculated from the equation:

$$c_i = p_j x^{(ij)} + q_j y^{(ij)} + c_j - p_i x^{(ij)} - q_i y^{(ij)}$$

Main Points of Algorithm

– If two of the neighboring patches, say j and k, have already been visited, i.e., the variables p_k, q_k, c_{ik} and p_j, q_j, c_j have been updated, then only one variable p_i is independent, and is perturbed. q_i, c_i can be evaluated as

$$c_i = p_j x^{(ij)} + q_j y^{(ij)} + c_j - p_i x^{(ij)} - q_i y^{(ij)}$$

$$q_i = \frac{p_k x^{(ik)} + q_k y^{(ik)} + c_k - p_i x^{(ik)} - c_i}{y^{(ik)}}$$

Main Points of Algorithm

- The perturbation of structure parameters (surface normal) for each patch, results in the change of coordinates (X,Y,Z) of the nodes of wireframe.

Updating of (X,Y,Z):

Patches *i, j, k* intersect at node *n*.

$$p_i X^{(n)} + q_i Y^{(n)} + c_i = p_j X^{(n)} + q_j Y^{(n)} + c_j$$

$$p_i X^{(n)} + q_i Y^{(n)} + c_i = p_k X^{(n)} + q_k Y^{(n)} + c_k$$

$$\begin{bmatrix} X^{(n)} \\ Y^{(n)} \end{bmatrix} = \begin{bmatrix} p_i - p_j & q_i - q_j \\ p_i - p_k & q_i - q_k \end{bmatrix}^{-1} \begin{bmatrix} c_j - c_i \\ -c_i + c_k \end{bmatrix}$$

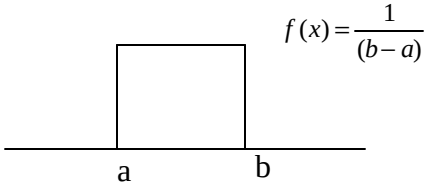
$$Z^{(i)} = p_i X^{(i)} + q_i Y^{(i)} + c_i$$

Algorithm

- **Estimate light source direction**
- Initialize coordinates of all nodes using approximately scaled wireframe model. Determine initial values of surface normals for each triangle.
- Determine initial motion parameters based on selected feature correspondences and their depth values from wireframe model. (Assume motion parameters.)
- (A) Compute the value of error function E.

- If error E is less than some threshold, then stop
- Else
  - Perturb motion parameters (3 rotations and 2 translations) by random amount (zero mean Gaussian, s.d. equal to error E) (you can use uniform distribution)
  - Perturb structure parameters (p,q,c):
    - Perturb p, q, and c of first patch by adding random amount (zero mean Gaussian, s.d. equal to error E)
    - Increment count for all neighbors of patch-1 by 1

Uniform Distribution



$$f(x) = \frac{1}{(b-a)}$$

$$\overline{X} = mean = \frac{(a+b)}{2}$$

$$\sigma^2 = \text{variance} = \frac{(a-b)^2}{12}$$

Use rand() in C
to generate random
number between a range.

- For patch 2 to *n*
  - If the count==1
    - » Perturb *p* and *q*
    - » Compute *c* using equation for c_i
    - » Increment the count

$$c_i = p_j x^{(ij)} + q_j y^{(ij)} + c_j - p_i x^{(ij)} - q_i y^{(ij)}$$

  - If count==2
    - » Perturb p_i
    - » Compute c_i and q_i using equations

$$c_i = p_j x^{(ij)} + q_j y^{(ij)} + c_j - p_i x^{(ij)} - q_i y^{(ij)}$$

$$q_i = \frac{p_k x^{(ik)} + q_k y^{(ik)} + c_k - p_i x^{(ik)} - c_i}{y^{(ik)}}$$

    - » Increment the count
  - If *p*, *q* and *c* for at least three patches intersecting at node are updated, then update the coordinates of the node using equation.

$$\begin{bmatrix} X^{(n)} \\ Y^{(n)} \end{bmatrix} = \begin{bmatrix} p_i - p_j & p_j - p_k \\ q_i - q_j & q_j - q_k \end{bmatrix}^{-1} \begin{bmatrix} c_j - c_i \\ -c_j + c_k \end{bmatrix}$$

$$Z^{(i)} = p_i X^{(i)} + q_i Y^{(i)} + c_i$$

- Go to step (A)

Program

- Candide Wireframe model is given
- Thirteen features on the face image, and corresponding nodes in the wireframe model are given
- Estimate pose: rotation, scaling, and translation of wireframe model so that it fits with the image
- Estimate frame to frame 3-D motion, distortion of wireframe model nodes.
- Use the estimated 3-D motion, and distortion of nodes to synthesize video sequence