Estimation Using Flexible Wireframe Model

Main Points

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

Generalized Optical Flow Constraint

Lambertian Model

$$f(x, y, t) = r N(t).L$$

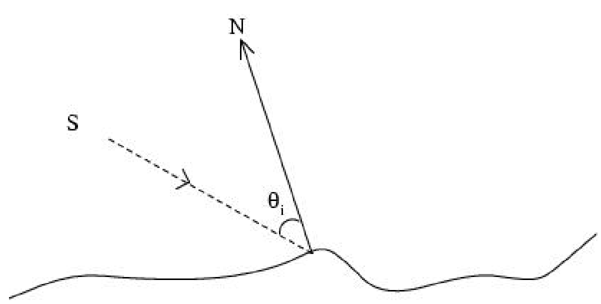
Albedo
Surface Normal
(-p,-q,1)

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

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

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

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

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

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

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

Orthographic Projection

$$u = \dot{x} = \Omega_2 Z - \Omega_3 y + V_1$$

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

(u,v) is optical flow

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

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

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

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

Optical flow equation

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

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

Homework 3.1
Show this.

Equation 24.8, page 473.

Error Function

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

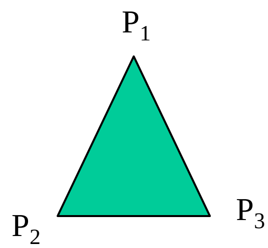
$$p_i x_1^{(ij)}+q_i x_2^{(ij)}+c_i = p_j x_1^{(ij)}+q_j x_2^{(ij)}+c_j$$

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

Homework 3.3
Show this.

Equation of a Planar Patch

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



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

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

Equation of a Planar Patch

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

Homework 3.2
Show this.

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

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

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

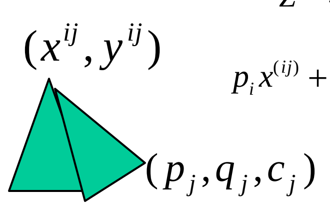
Structure of Wireframe Model

- 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.
- 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

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

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

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.

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

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

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

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.
- (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)
  - 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

- 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

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

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

  - » 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)

Synthesizing Realistic Facial Expressions from Photographs

Pighin et al

SIGGRAPH'98

The Artist's Complete Guide to Facial Expression: Gary Faigin

- There is no landscape that we know as well as the human face. The twenty-five-odd square inches containing the features is the most intimately scrutinized piece of territory in existence, examined constantly, and carefully, with far more than an intellectual interest. Every detail of the nose, eyes, and mouth, every regularity in proportion, every variation from one individual to the next, are matters about which we are all authorities.

Main Points

- One view is not enough.
- Fitting of wire frame model to the image is a complex problem (pose estimation)
- Texture mapping is important problem

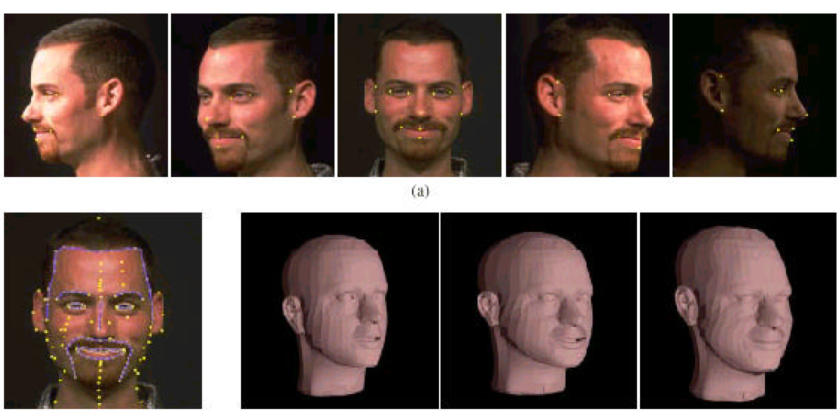
Synthesizing Realistic Facial Expressions

- 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 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.

Synthesizing Realistic Facial Expressions



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

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

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

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

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

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

Model Fitting

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

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

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

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

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.

Model Fitting

- Solve for unknowns in five steps:

- Use linear least squares fit.

- When solving for an unknown, assume other parameters are known.

Least Squares Fit

$$a_j.x - b_j = 0$$

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

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

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

Update for p

$$a_{2k+0} = w_i^k (x_i^k \boldsymbol{h}^k r_z^k - s^k r_x^k) \qquad \mathrm{b}_{2k+0} = w_i^k (s^k T_x^k - x_i^k)$$
$$a_{2k+1} = w_i^k (y_i^k \boldsymbol{h}^k r_z^k - s^k r_y^k) \qquad \mathrm{b}_{2k+1} = w_i^k (s^k T_y^k - y_i^k)$$

$$a_j.x - b_j = 0$$

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

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

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

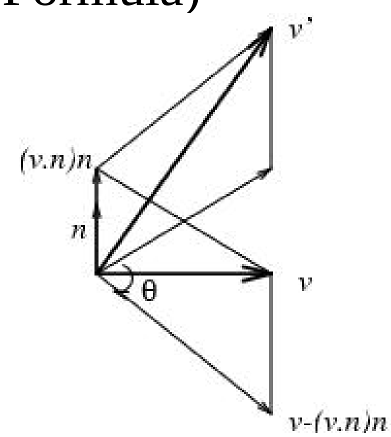
$$\sum_j (a_j a_j^T)x = \sum_j b_j a_j$$

Update for s^k

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

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

Rotation Around an Arbitrary Axis (Rodriguez's Formula)



$$V = (V.n)n + (V - (V.n)n)$$

$$V'_\perp = \cos\boldsymbol{q}(V - (V.n)n) + \sin\boldsymbol{q}(n \times (V - (V.n)n)$$

$$V'_{\mathrm{II}} = (V - (V.n)n)$$

$$V' = V'_\perp + V'_{\mathrm{II}}$$

$$V' = \cos q V + \sin q n \times v + (1 - \cos q)(V.n)n$$

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

$$n \times (n \times V) = (V.n)n - V$$

Rotation Around an Arbitrary Axis (Rodriguez's Formula)

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

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

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

Rotation Around an Arbitrary Axis (Rodriguez's Formula)

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

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

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