Lecture-19

Structure from Motion

Program-3 (Due Tuesday April 24, 4PM)

- Implement Lucas-Kanade optical flow method using pyramids.
  - You will need to implement the following pieces:
    - Simple Lucas-Kanade without pyramids
    - Gaussian pyramids
    - Interpolation of optical flow in order to propagate optical flow from higher level of pyramid to lower level of pyramid
  - You will be given two images, your program should compute
    - Optical flow without pyramids and displays the flow vectors on the screen. You will have to sample (e.g. show flow vector at every fourth pixel) and scale the flow vectors to show on the screen. Otherwise there will be too many vectors, you won’t be able to see them.
    - Optical flow with pyramids and displays the flow vectors on the screen.

Problem

- Given optical flow or point correspondences, compute 3-D motion (translation and rotation) and depth.

Main Points

- What projection?
- How many points?
- How many frames?
- Single Vs multiple motions.
- Mostly non-linear problem.

Displacement Model

Point Correspondences

3-D Rigid Motion (displacement)

$$\begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix} = R\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + T = \begin{bmatrix} 1 & -a & b \\ a & 1 & g \\ -b & g & 1 \end{bmatrix}\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + \begin{bmatrix} T_X \\ T_Y \\ T_Z \end{bmatrix}$$

$$X' = X - aY + bZ + T_X$$
$$Y' = aX + Y - gZ + T_Y$$
$$Z' = -bX - gY + Z + T_Z$$

Orthographic Projection (displacement model)

$$X' = X - aY + bZ + T_X$$
$$Y' = aX + Y - gZ + T_Y$$
$$Z' = -bX - gY + Z + T_Z$$

$$x' = x - ay + bZ + T_X$$
$$y' = ax + y - gZ + T_Y$$

Perspective Projection (displacement)

$$X' = X - aY + bZ + T_X$$
$$Y' = aX + Y - gZ + T_Y$$
$$Z' = -bX - gY + Z + T_Z$$

$$x' = \frac{x - ay + b + \frac{T_X}{Z}}{-bx + gy + 1 + \frac{T_Z}{Z}}$$

$$y' = \frac{ax + y + g + \frac{T_Y}{Z}}{-bx + gy + 1 + \frac{T_Z}{Z}}$$

Instantaneous Velocity Model

Optical Flow

3-D Rigid Motion

$$\begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix} = \begin{bmatrix} 1 & -a & b \\ a & 1 & -g \\ -b & g & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + \begin{bmatrix} T_X \\ T_Y \\ T_Z \end{bmatrix}$$

$$\begin{bmatrix} X'-X \\ Y'-Y \\ Z'-Z \end{bmatrix} = \begin{bmatrix} 0 & -\boldsymbol{a} & \boldsymbol{b} \\ \boldsymbol{a} & 0 & -\boldsymbol{g} \\ -\boldsymbol{b} & \boldsymbol{g} & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + \begin{bmatrix} T_X \\ T_Y \\ T_Z \end{bmatrix}$$

$$\begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{Z} \end{bmatrix} = \begin{bmatrix} 0 & -\Omega_3 & \Omega_2 \\ \Omega_3 & 0 & -\Omega_1 \\ -\Omega_2 & \Omega_1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + \begin{bmatrix} V_1 \\ V_2 \\ V_3 \end{bmatrix}$$

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

3-D Rigid Motion

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

$$\mathbf{X} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix},$$
$$\Omega = \begin{bmatrix} \Omega_1 \\ \Omega_2 \\ \Omega_3 \end{bmatrix}$$

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

Cross Product

Orthographic Projection

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

$$y = Y$$
$$x = X$$

$$u = \dot{x} = \Omega_2 Z - \Omega_3 y + V_1$$
$$v = \dot{y} = \Omega_3 x - \Omega_1 Z + V_2$$

(u,v) is optical flow

Perspective Projection (arbitrary flow)

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

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

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

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

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

Perspective Projection (optical flow)

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

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

$$u = \frac{fV_1 - V_3 x}{Z} + f\Omega_2 - \Omega_3 y - \frac{\Omega_1}{f}xy + \frac{\Omega_2}{f}x^2$$

$$v = \frac{fV_2 - V_3 y}{Z} - f\Omega_1 + \Omega_3 x + \frac{\Omega_2}{f}xy - \frac{\Omega_1}{f}y^2$$

Pure Translation (FOE)

$$u^{(T)} = \frac{fV_1 - V_3 x}{Z}$$

$$v^{(T)} = \frac{fV_2 - V_3 y}{Z}$$

$$x_0 = f\,\frac{V_1}{V_3},\, y_0 = f\,\frac{V_2}{V_3}$$

$$u^{(T)} = (x_0 - x)\frac{V_3}{Z}$$

$$v^{(T)} = (y_0 - y)\frac{V_3}{Z}$$

Pure Translation (FOE)

$$u^{(T)} = \frac{fV_1 - V_3 x}{Z}$$

$$v^{(T)} = \frac{fV_2 - V_3 y}{Z}$$

$$u^{(T)} = \frac{fV_1}{Z}$$

If V_3=0

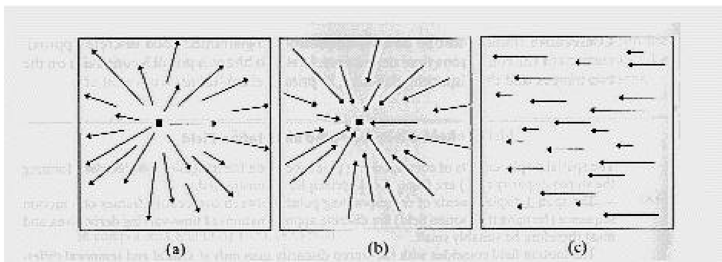
$$v^{(T)} = \frac{fV_2}{Z}$$

Pure Translation (FOE)

- If V3 is not zero, the flow field is radial, and all vectors point towards (or away from) a single point. If $V_3=0$, the flow field is parallel.
- The length of flow vectors is inversely proportional to the depth, if V_3 is not zero, then it is also proportional to the distance between p and p_0.

Pure Translation (FOE)

- p_0 is the vanishing point of the direction of translation.
- p_0 is the intersection of the ray parallel to the translation vector with the image plane.



Structure From Motion

ORTHOGRAPHIC PROJECTION

Orthographic Projection (displacement)

$$\begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix} = R\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + T = \begin{bmatrix} 1 & -a & b \\ a & 1 & g \\ -b & g & 1 \end{bmatrix}\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + \begin{bmatrix} T_X \\ T_Y \\ T_Z \end{bmatrix}$$

$$x' = x - ay + bZ + T_X$$
$$y' = ax + y - gZ + T_Y$$

Simple Method

- **Two Steps Method**

  -Assume depth is known, compute motion

$$\begin{bmatrix} x' - x \\ y' - y \end{bmatrix} = \begin{bmatrix} -y & Z & 0 & 1 & 0 \\ x & 0 & -Z & 0 & 1 \end{bmatrix}\begin{bmatrix} a \\ b \\ g \\ T_x \\ T_y \end{bmatrix}$$

Simple Method

-Assume motion is known, refine depth

$$\begin{bmatrix} b \\ -g \end{bmatrix} [Z] = \begin{bmatrix} x' - x - ay - T_x \\ y' - y - ax - T_y \end{bmatrix}$$

Structure from Motion

Heeger & Jepson sfm method

Three Step Algorithm

- Find Translation using search.
- Find Rotation using least squares fit.
- Find Depth.

Heeger & Jepson sfm method

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

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

$$\begin{bmatrix} u(x,y) \\ v(x,y) \end{bmatrix} = \frac{1}{Z(x,y)} \begin{bmatrix} -1 & 0 & x \\ 0 & -1 & y \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \\ V_3 \end{bmatrix} + \begin{bmatrix} xy & -(1+x^2) & y \\ 1+y^2 & -xy & -x \end{bmatrix} \begin{bmatrix} \Omega_1 \\ \Omega_2 \\ \Omega_3 \end{bmatrix}$$

Heeger & Jepson sfm method

$$\begin{bmatrix} u(x,y) \\ v(x,y) \end{bmatrix} = \frac{1}{z(x,y)} \begin{bmatrix} -1 & 0 & x \\ 0 & -1 & y \end{bmatrix} \mathbf{V} + \begin{bmatrix} xy & -(1+x^2) & y \\ 1+y^2 & -xy & -x \end{bmatrix} \Omega$$

$$\downarrow$$

$$\Theta(\mathbf{x},\mathbf{y}) = p(x,y)\mathbf{A}(x,y)\mathbf{V} + \mathbf{B}(x,y)\Omega$$

One point (x,y)

Heeger & Jepson sfm method

$$\Theta(\mathbf{x},\mathbf{y}) = p(x,y)\mathbf{A}(x,y)\mathbf{V} + \mathbf{B}(x,y)\Omega$$

$$\begin{bmatrix} \Theta(x_1,y_1) \\ \vdots \\ \vdots \\ \vdots \\ \Theta(x_n,y_n) \end{bmatrix} = \begin{bmatrix} \mathbf{A}(x_1,y_1)\mathbf{V} & \cdots & \cdots & \cdots & 0 \\ \vdots & & & & \vdots \\ \vdots & & & & \vdots \\ \vdots & & & & \vdots \\ \mathbf{0} & \cdots & \cdots & \cdots & \mathbf{A}(x_n,y_n)\mathbf{V} \end{bmatrix} \begin{bmatrix} p(x_1,y_1) \\ \vdots \\ \vdots \\ p(x_n,y_n) \\ \end{bmatrix} + \begin{bmatrix} \mathbf{B}(x_1,y_1) \\ \vdots \\ \vdots \\ \\ \mathbf{B}(x_n,y_n) \end{bmatrix} \Omega$$

n points

Heeger & Jepson sfm method

$$\begin{bmatrix} \Theta(x_1, y_1) \\ \vdots \\ \vdots \\ \vdots \\ \Theta(x_n, y_n) \end{bmatrix} = \begin{bmatrix} \mathbf{A}(x_1, y_1)\mathbf{V} & \dots & \dots & \dots & 0 & \mathbf{B}(x_1, y_1) \\ \vdots & & & & & \vdots \\ \vdots & & & & & \vdots \\ \vdots & & & & & \vdots \\ \mathbf{0} & \dots & \dots & \dots & \mathbf{A}(x_n, y_n)\mathbf{V} & \mathbf{B}(x_n, y_n) \end{bmatrix} \begin{bmatrix} p(x_1, y_1) \\ \vdots \\ \vdots \\ p(x_n, y_n) \\ \Omega \end{bmatrix}$$

$$\Theta = \mathbf{C}(\mathbf{V})\mathbf{q}$$

Heeger & Jepson sfm method

$$E(\mathbf{V}, \mathbf{q}) = \left\| \Theta - \mathbf{C}(\mathbf{V})\mathbf{q} \right\|^2$$

$$\mathbf{E}(\mathbf{V}) = \left\| \Theta^T \mathbf{C}^{\perp}(V) \right\|^2$$

$\mathbf{C}^{\perp}(V)$ Orthogonal complement to $\mathbf{C}(\mathbf{V})$

Find translation by search.

$$\mathbf{C}(\mathbf{V}) = \overline{\mathbf{C}}(\mathbf{V})\mathbf{U}(\mathbf{V})$$

QR decomposition

Orthonormal &
Upper triangular

$$E(\mathbf{V},\mathbf{q}) = \left\|\Theta - \overline{\mathbf{C}}(\mathbf{V})\mathbf{U}(\mathbf{V})\mathbf{q}\right\|^2$$

$$E(\mathbf{V},\mathbf{q}) = \left\|\Theta - \overline{\mathbf{C}}(\mathbf{V})\overline{\mathbf{q}}\right\|^2$$

$$\Downarrow \quad \text{minimize}$$

$$\hat{\mathbf{q}} = \overline{\mathbf{C}}^{\mathbf{T}}(\mathbf{V})\Theta$$

$$E(\mathbf{V}) = \left\|\Theta - \overline{\mathbf{C}}(\mathbf{V})\overline{\mathbf{C}}^T(\mathbf{V})\Theta\right\|^2$$

$$E(\mathbf{V}) = \left\|(I - \overline{\mathbf{C}}(\mathbf{V})\overline{\mathbf{C}}^T(\mathbf{V}))\Theta\right\|^2$$

Null space

$$\mathbf{E}(\mathbf{V}) = \left\|\Theta^T\mathbf{C}^{\perp}(V)\right\|^2$$

Translation

Unit vector translation can be represented by spherical coordinates:

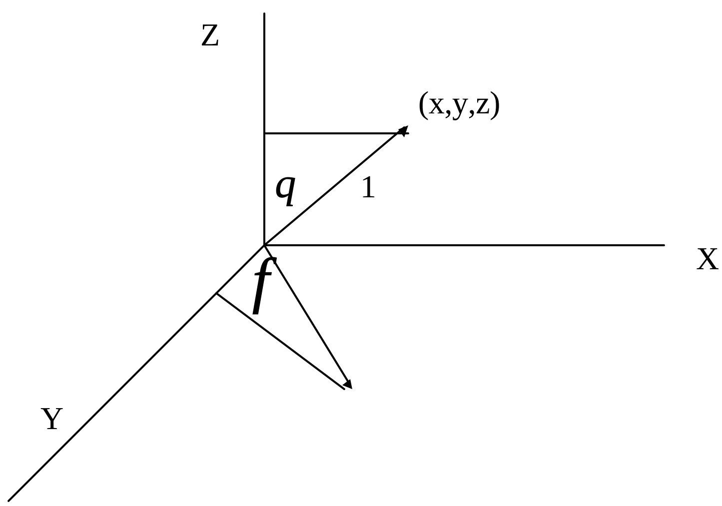
$$\mathbf{V} = (\sin q \cos f, \sin q \sin f, \cos q)$$

$0 \le q \le 90$ Slant

$0 \le f \le 360$ Tilt

Only half of sphere can be considered

Spherical Coordinates



Rotation

$$\Theta(\mathbf{x},\mathbf{y}) = p(x,y)\mathbf{A}(x,y)\mathbf{V} + \mathbf{B}(x,y)\boldsymbol{\Omega}$$

$$d^T(x,y,V)\Theta(\mathbf{x},\mathbf{y}) = d^T(x,y,V)\mathbf{B}(x,y)\boldsymbol{\Omega}$$

$d^T(x,y,V)$ is perpendicular to $\mathbf{A}(x,y)\mathbf{V}$

Rotation

Find rotation by least squares fit

$$d^T(x_1,y_1,V)\Theta(\mathbf{x}_1,\mathbf{y}_1) = d^T(x_1,y_1,V)\mathbf{B}(x_1,y_1)\boldsymbol{\Omega}$$
$$\vdots$$
$$d^T(x_n,y_n,V)\Theta(\mathbf{x}_n,\mathbf{y}_n) = d^T(x_n,y_n,V)\mathbf{B}(x_n,y_n)\boldsymbol{\Omega}$$

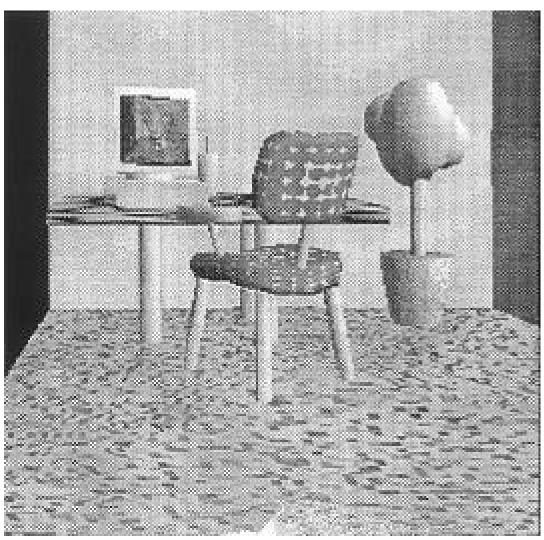
Depth

Find depth for each pixel (x,y) from following eqs

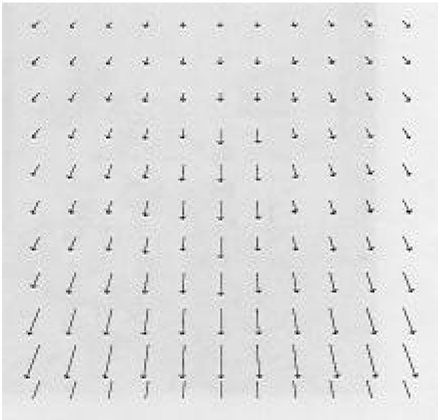
$$u = -(\frac{V_1}{Z} + \Omega_2) + \frac{V_3}{Z} x + \Omega_3 y + \Omega_1 xy - \Omega_2 x^2$$

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

Synthetic Image

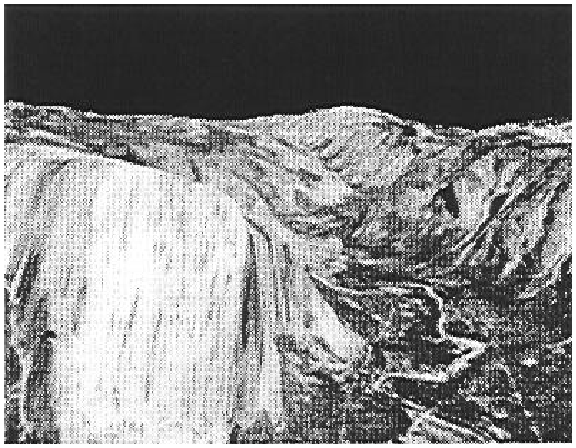


Optical Flow

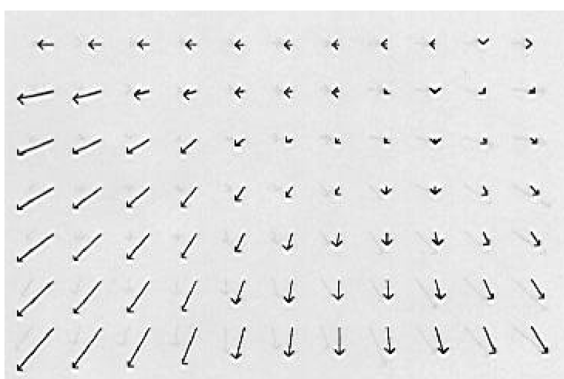


Computed Depth Map

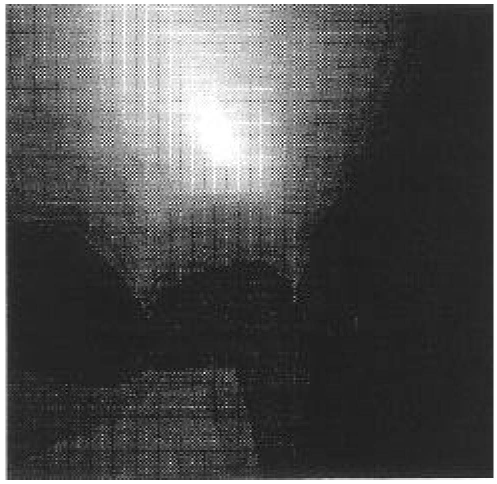
Synthetic Image



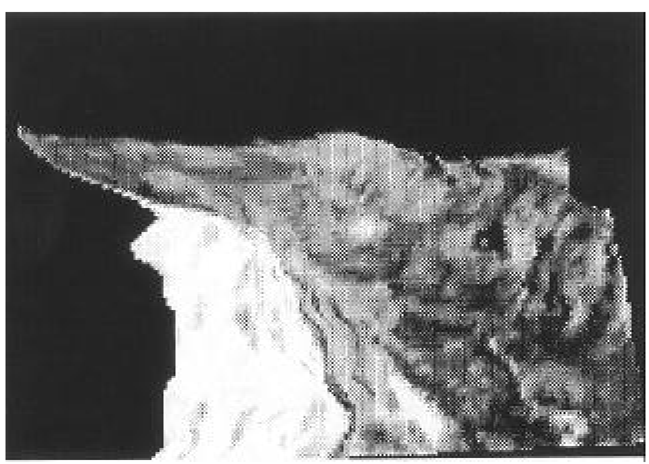
Optical Flow



Translation Search Space



Novel View Generated from Reconstructed Depth



Another Novel View Generated from Reconstructed Depth