JPEG Quantization Table (Luma)

$$Q_{u,v} = \begin{bmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 51 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{bmatrix}$$

JPEG Quantization Table (Chroma)

$$\begin{bmatrix} 17 & 18 & 24 & 47 & 99 & 99 & 99 & 99 \\ 18 & 21 & 26 & 66 & 99 & 99 & 99 & 99 \\ 24 & 26 & 56 & 99 & 99 & 99 & 99 & 99 \\ 47 & 66 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \end{bmatrix}$$

Lecture-8

STRUCTURE FROM MOTION

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.

Orthographic Projection (displacement)

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

$$x' = x - ay + bZ + T_X$$

$$y' = ax + y - gZ + T_Y$$

Perspective Projection (displacement)

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

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

Orthographic Projection (optical flow)

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

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

Perspective Projection (optical flow)

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

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

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 V3=0

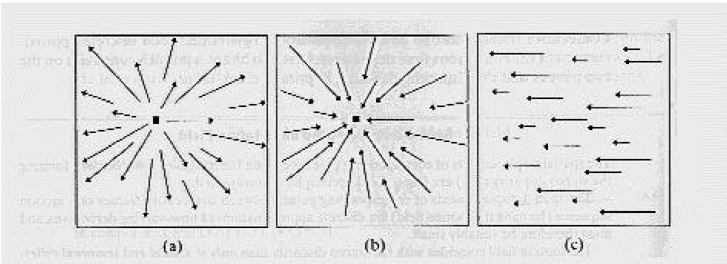
$$v^{(T)} = \frac{fV_2}{Z}$$

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 V3=0, the flow field is parallel.
- The length of flow vectors is inversely proportional to the depth, if V3 is not zero, then it is also proportional to the distance between p and p0.

Pure Translation (FOE)

- P0 is the vanishing point of the direction of translation.
- P0 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

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

⇓

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

One point (x,y)

Heeger & Jepson sfm method

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

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

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

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

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

⇓ 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:

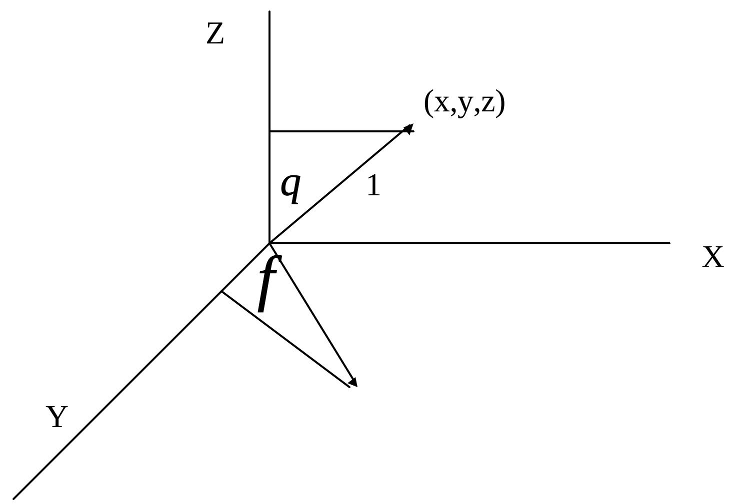
$$\mathbf{V} = (\sin q \cos f, \sin q \sin f, \cos q)$$

$0 \le q \le 90$ Slant

$0 \le f \le 360$ Tilt

Only half of sphere can be considered

Spherical Coordinates



Rotation

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

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

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

Rotation

Find rotation by least squares fit

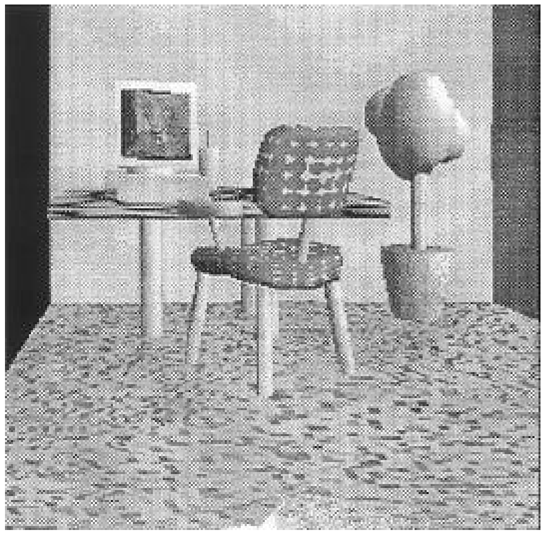
$$d^T(x_1, y_1, V)\Theta(\mathbf{x}_1, \mathbf{y}_1) = d^T(x_1, y_1, V)\mathbf{B}(x_1, y_1)\mathbf{\Omega}$$
$$\vdots$$
$$d^T(x_n, y_n, V)\Theta(\mathbf{x}_n, \mathbf{y}_n) = d^T(x_n, y_n, V)\mathbf{B}(x_n, y_n)\Omega$$

Depth

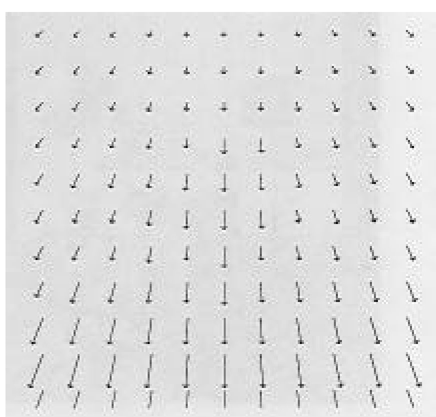
Find depth for each pixel (x,y) from following eqs

$$u = -(\frac{V_1}{Z} + \Omega_2) + \frac{V_3}{Z}x + \Omega_3 y + \Omega_1 xy - \Omega_2 x^2$$
$$v = -(\frac{V_2}{Z} - \Omega_1) - \Omega_3 x + \frac{V_3}{Z}y - \Omega_2 xy + \Omega_1 y^2$$

Synthetic Image

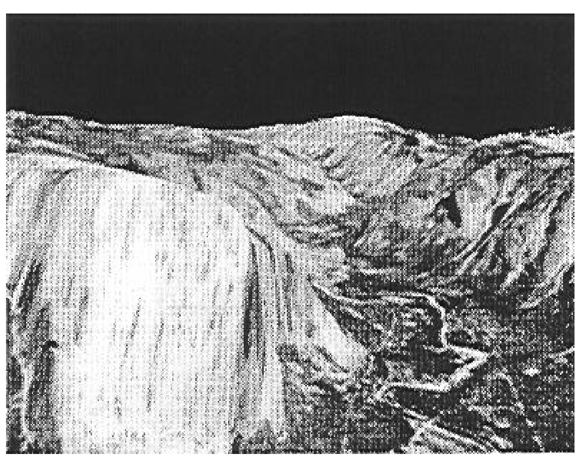


Optical Flow

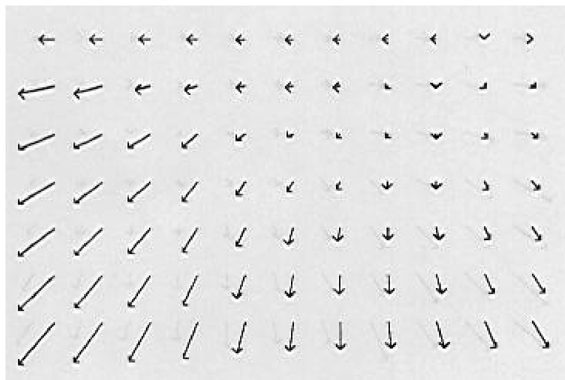


Computed Depth Map

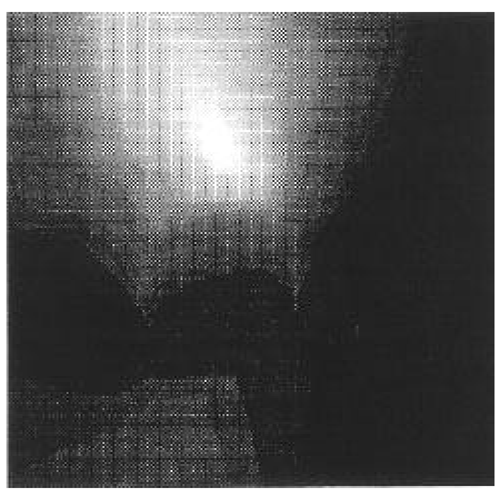
Synthetic Image



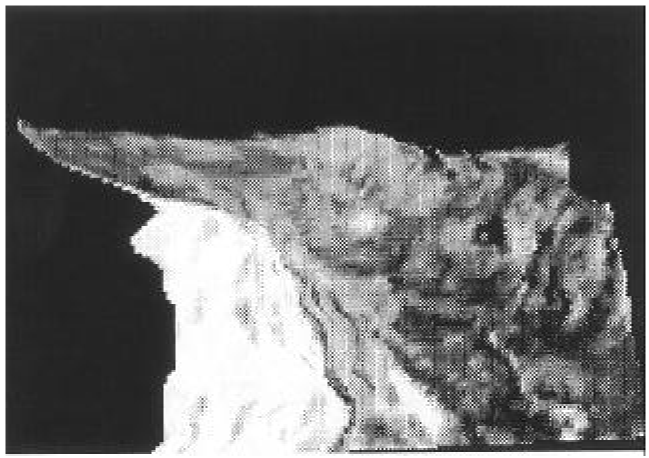
Optical Flow



Translation Search Space



Novel View Generated from Reconstructed Depth



Another Novel View Generated from Reconstructed Depth