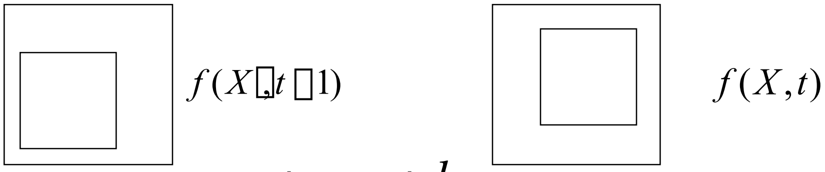
Lecture-6

Szeliski, Mann & Picard

Szeliski

Projective

Projective



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

$$y' = \frac{a_3 x + a_4 y + b_2}{c_1 x + c_2 y + 1}$$

Szeliski

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

Projective

$$y' = \frac{a_3 x + a_4 y + b_2}{c_1 x + c_2 y + 1}$$

$$E = \sum [f(x',y') - f(x,y)]^2 = \sum e^2$$

$\Big\Downarrow$ min

Szeliski

Motion Vector:

$$\mathbf{m} = \begin{bmatrix} a_1 & a_2 & a_3 & a_4 & b_1 & b_2 & c_1 & c_2 \end{bmatrix}^T$$

Szeliski (Levenberg-Marquadet)

$$\alpha_{kl} = \sum_n \frac{\partial e_n}{\partial m_k} \frac{\partial e_n}{\partial m_l} \qquad b_k = -\sum e \frac{\partial e_n}{\partial m_k}$$

gradient

$$\Delta m = (A + \lambda I)^{-1} b$$

Approximation of Hessian (*J^T^J, Jacobian)*

$$\frac{\partial e}{\partial a_1} = \frac{\partial e}{\partial x'}\frac{\partial x'}{\partial a_1} + \frac{\partial e}{\partial y'}\frac{\partial y'}{\partial a_1}$$

$$E = \sum [f(x',y') - f(x,y)]^2 = \sum e^2$$

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

$$y' = \frac{a_3 x + a_4 y + b_2}{c_1 x + c_2 y + 1}$$

Approximation of Hessian

$$J^T = \begin{bmatrix} \frac{\partial e_1}{\partial m_1} & \frac{\partial e_2}{\partial m_1} & \cdots & \frac{\partial e_n}{\partial m_1} \\ \frac{\partial e_1}{\partial m_2} & \frac{\partial e_2}{\partial m_2} & \cdots & \frac{\partial e_n}{\partial m_2} \\ \frac{\partial e_1}{\partial m_3} & \frac{\partial e_2}{\partial m_3} & \cdots & \frac{\partial e_n}{\partial m_3} \\ \frac{\partial e_1}{\partial m_4} & \frac{\partial e_2}{\partial m_4} & \cdots & \frac{\partial e_n}{\partial m_4} \\ \frac{\partial e_1}{\partial m_5} & \frac{\partial e_2}{\partial m_5} & \cdots & \frac{\partial e_n}{\partial m_5} \\ \frac{\partial e_1}{\partial m_6} & \frac{\partial e_2}{\partial m_6} & \cdots & \frac{\partial e_n}{\partial m_6} \\ \frac{\partial e_1}{\partial m_7} & \frac{\partial e_2}{\partial m_7} & \cdots & \frac{\partial e_n}{\partial m_7} \\ \frac{\partial e_1}{\partial m_8} & \frac{\partial e_2}{\partial m_8} & \cdots & \frac{\partial e_n}{\partial m_8} \end{bmatrix} \quad J = \begin{bmatrix} \frac{\partial e_1}{\partial m_1} & \frac{\partial e_1}{\partial m_2} & \frac{\partial e_1}{\partial m_3} & \frac{\partial e_1}{\partial m_4} & \frac{\partial e_1}{\partial m_5} & \frac{\partial e_1}{\partial m_6} & \frac{\partial e_1}{\partial m_7} & \frac{\partial e_1}{\partial m_8} \\ \frac{\partial e_2}{\partial m_1} & \frac{\partial e_2}{\partial m_2} & \frac{\partial e_2}{\partial m_3} & \frac{\partial e_2}{\partial m_4} & \frac{\partial e_2}{\partial m_5} & \frac{\partial e_2}{\partial m_6} & \frac{\partial e_2}{\partial m_7} & \frac{\partial e_2}{\partial m_8} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \frac{\partial e_n}{\partial m_1} & \frac{\partial e_n}{\partial m_2} & \frac{\partial e_n}{\partial m_3} & \frac{\partial e_n}{\partial m_4} & \frac{\partial e_n}{\partial m_5} & \frac{\partial e_n}{\partial m_6} & \frac{\partial e_n}{\partial m_7} & \frac{\partial e_n}{\partial m_8} \end{bmatrix}$$

$$A = J^T J$$

$$\alpha_{kl} = \sum_n \frac{\partial e_n}{\partial m_k}\frac{\partial e_n}{\partial m_l}$$

Gradient Vector

$$b = \begin{bmatrix} -\sum_n e_n \frac{\partial e_n}{\partial a_1} \\ -\sum_n e_n \frac{\partial e_n}{\partial a_2} \\ -\sum_n e_n \frac{\partial e_n}{\partial a_3} \\ -\sum_n e_n \frac{\partial e_n}{\partial a_4} \\ -\sum_n e \frac{\partial e_n}{\partial b_1} \\ -\sum_n e_n \frac{\partial e_n}{\partial b_2} \\ -\sum_n e_n \frac{\partial e_n}{\partial c_1} \\ -\sum_n e_n \frac{\partial e_n}{\partial c_2} \end{bmatrix}$$

Partial Derivatives wrt motion parameters

$$\frac{\partial e}{\partial a_1} = \frac{\partial e}{\partial x'}\frac{\partial x'}{\partial a_1} + \frac{\partial e}{\partial y'}\frac{\partial y'}{\partial a_1}$$

$$x' = \frac{a_1 x + a_2 y + b_1}{c_1 x + c_2 y + 1}, \quad y' = \frac{a_3 x + a_4 y + b_2}{c_1 x + c_2 y + 1}$$

$$\frac{\partial x'}{\partial a_1} = \frac{x}{c_1 x + c_2 y + 1} \qquad \frac{\partial y'}{\partial a_1} = 0$$

$$\frac{\partial x'}{\partial a_2} = \frac{y}{c_1 x + c_2 y + 1} \qquad \frac{\partial y'}{\partial a_2} = 0$$

$$\frac{\partial x'}{\partial a_3} = 0 \qquad \frac{\partial y'}{\partial a_3} = \frac{x}{c_1 x + c_2 y + 1}$$

$$\frac{\partial x'}{\partial a_4} = 0 \qquad \frac{\partial y'}{\partial a_4} = \frac{y}{c_1 x + c_2 y + 1}$$

$$\frac{\partial x'}{\partial b_1} = \frac{1}{c_1 x + c_2 y + 1} \qquad \frac{\partial y'}{\partial b_1} = 0$$

$$\frac{\partial x'}{\partial b_2} = 0 \qquad \frac{\partial y'}{\partial b_2} = \frac{1}{c_1 x + c_2 y + 1}$$

$$\frac{\partial x'}{\partial c_1} = \frac{-x(a_1 x + a_2 y + b_1)}{(c_1 x + c_2 y + 1)^2} \qquad \frac{\partial y'}{\partial c_1} = \frac{-x(a_3 x + a_4 y + b_2)}{(c_1 x + c_2 y + 1)^2}$$

$$\frac{\partial x'}{\partial c_2} = \frac{-y(a_1 x + a_2 y + b_1)}{(c_1 x + c_2 y + 1)^2} \qquad \frac{\partial y'}{\partial c_2} = \frac{-y(a_3 x + a_4 y + b_2)}{(c_1 x + c_2 y + 1)^2}$$

Partial derivatives wrt image coordinates

$$E = \sum [f(x',y') - f(x,y)]^2 = \sum e^2$$

$$\frac{\partial e}{\partial x} = f_{x'}$$

$$\frac{\partial e}{\partial y} = f_{y'}$$

Partial derivatives

$$\frac{\partial e}{\partial a_1} = \frac{\partial e}{\partial x'}\frac{\partial x'}{\partial a_1} + \frac{\partial e}{\partial y'}\frac{\partial y'}{\partial a_1} = f_{x'}\frac{x}{c_1 x + c_2 y + 1}$$

$$\frac{\partial e}{\partial a_2} = \frac{\partial e}{\partial x'}\frac{\partial x'}{\partial a_2} + \frac{\partial e}{\partial y'}\frac{\partial y'}{\partial a_2} = f_{x'}\frac{y}{c_1 x + c_2 y + 1}$$

$$\frac{\partial e}{\partial a_3} = \frac{\partial e}{\partial x'}\frac{\partial x'}{\partial a_3} + \frac{\partial e}{\partial y'}\frac{\partial y'}{\partial a_3} = f_{y'}\frac{x}{c_1 x + c_2 y + 1}$$

$$\frac{\partial e}{\partial a_4} = \frac{\partial e}{\partial x'}\frac{\partial x'}{\partial a_4} + \frac{\partial e}{\partial y'}\frac{\partial y'}{\partial a_4} = f_{y'}\frac{y}{c_1 x + c_2 y + 1}$$

$$\frac{\partial e}{\partial b_1} = \frac{\partial e}{\partial x'}\frac{\partial x'}{\partial b_1} + \frac{\partial e}{\partial y'}\frac{\partial y'}{\partial b_1} = f_{x'}\frac{1}{c_1 x + c_2 y + 1}$$

$$\frac{\partial e}{\partial b_2} = \frac{\partial e}{\partial x'}\frac{\partial x'}{\partial b_2} + \frac{\partial e}{\partial y'}\frac{\partial y'}{\partial b_2} = f_{y'}\frac{1}{c_1 x + c_2 y + 1}$$

$$\frac{\partial e}{\partial c_1} = \frac{\partial e}{\partial x'}\frac{\partial x'}{\partial c_1} + \frac{\partial e}{\partial y'}\frac{\partial y'}{\partial c_1} = f_{x'}\frac{-x(a_1 x + a_2 y + b_1)}{(c_1 x + c_2 y + 1)^2} + f_{y'}\frac{-x(a_3 x + a_4 y + b_2)}{(c_1 x + c_2 y + 1)^2}$$

$$\frac{\partial e}{\partial c_2} = \frac{\partial e}{\partial x'}\frac{\partial x'}{\partial c_2} + \frac{\partial e}{\partial y'}\frac{\partial y'}{\partial c_2} = f_{x'}\frac{-y(a_1 x + a_2 y + b_1)}{(c_1 x + c_2 y + 1)^2} + f_{y'}\frac{-y(a_3 x + a_4 y + b_2)}{(c_1 x + c_2 y + 1)^2}$$

Gradient Vector

$$\mathbf{b} = \begin{bmatrix} -\sum e f_{x'} \frac{x}{c_1 x + c_2 y + 1} \\ -\sum e f_{x'} \frac{y}{c_1 x + c_2 y + 1} \\ -\sum e f_{y'} \frac{x}{c_1 x + c_2 y + 1} \\ -\sum e f_{y'} \frac{y}{c_1 x + c_2 y + 1} \\ -\sum e f_{x'} \frac{1}{c_1 x + c_2 y + 1} \\ -\sum e f_{y'} \frac{1}{c_1 x + c_2 y + 1} \\ \sum ex \left[\frac{f_{x'}(a_1 x + a_2 y + b_1) + f_{y'}(a_3 x + a_4 y + b_2)}{(c_1 x + c_2 y + 1)^2} \right] \\ \sum ey \left[\frac{f_{x'}(a_1 x + a_2 y + b_1) + f_{y'}(a_3 x + a_4 y + b_2)}{(c_1 x + c_2 y + 1)^2} \right] \end{bmatrix}$$

Szeliski (Levenberg-Marquadet)

- Start with some initial value of m, and λ=.001
  - For each pixel I at (x_i, y_i)
    - Compute (x', y') using projective transform.
    - Compute $e = f(x', y') - f(x, y)$
    - Compute $\frac{\partial e}{\partial m_k} = \frac{\partial e}{\partial x'} \frac{\partial x'}{\partial m_k} + \frac{\partial e}{\partial y'} \frac{\partial y'}{\partial m_k}$

Szeliski (Levenberg-Marquadet)

-Compute A **and** b

-Solve system

$$(A \square \square I)\square m = b$$

-Update

$$m^{t+1} = m^{t} + \square m$$

Szeliski (Levenberg-Marquadet)

- check if error has decreased, if not increase by a factor of 10 and compute a new $\square m$
- If error has decreased, decrease by a factor of 10 and compute a new $\square m$
- Continue iteration until error is below threshold.

Mann & Picard

Projective

Projective Flow (weighted)

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

Optical Flow const. equation

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

$$\mathbf{x}\square = \frac{A\mathbf{x}+\mathbf{b}}{\mathbf{C}^\mathbf{T}\mathbf{x}+1}$$

Projective transform

$$\mathbf{u} = \mathbf{x}\square\square\,\mathbf{x} = \frac{A\mathbf{x}+\mathbf{b}}{\mathbf{C}^\mathbf{T}\mathbf{x}+1}$$

Projective Flow (weighted)

$$\varepsilon_{flow} = \sum \left(\mathbf{u}^{\mathbf{T}} \mathbf{f}_{\mathbf{X}} + f_{\mathbf{t}} \right)^2$$

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

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

$\Downarrow$ **minimize**

Projective Flow (weighted)

- (b) Homework 2 Derive this equation Due Sept 25

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

$$a = [a_1, a_2, b_1, a_3, a_4, b_2, c_1, c_2]^T$$

$$\phi^T = [f_x x, f_x y, f_x, f_y x, f_y y, f_y, x f_t - x^2 f_x - xy f_y, y f_t - xy f_x - y^2 f_y]$$

Projective Flow (unweighted)

Bilinear

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

$$\Big\Downarrow$$

Taylor Series & remove
Square terms

$$u \quad + x = a_1 + a_2 x + a_3 y + a_4 xy$$

$$v \quad + y = a_5 + a_6 x + a_7 y + a_8 xy$$

Pseudo-Perspective

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

$$\Downarrow$$ **Taylor Series**

$$\begin{aligned} x + u &= a_1 + a_2 x + a_3 y + a_4 x^2 + a_5 xy \\ y + v &= a_6 + a_7 x + a_8 y + a_4 xy + a_5 y^2 \end{aligned}$$

Projective Flow (unweighted)

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

Minimize

Bilinear and Pseudo-Perspective

$$(\square\ \square\square^{T})\mathbf{q} = \square\square\ f_t\square$$

(c) homewwork Derive these eqs Sept 25

$$\square^{T} = \left[f_x(xy, x, y, 1), \quad f_y\left(xy, x, y, 1\right)\right]$$ **bilinear**

$$\square^{T} = \left[f_x(x, y, 1) \quad f_y(x, y, 1) \quad c_1 \quad c_2\right]$$

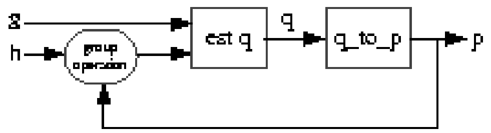
$$c_1 = x^2 f_x + xyf_x$$

$$c_2 = xyf_x + y^2 f_y$$

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 $(x_k\square, y_k\square)$
  - – estimate exact “p”:



True Projective

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

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

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

$$\mathbf{a} = \begin{bmatrix} a_1 & a_2 & b_1 & a_3 & a_4 & b_2 & c_1 & c_1 \end{bmatrix}^T$$

$$\begin{bmatrix} x_1' \\ y_1' \\ \vdots \\ x_k' \\ y_k' \end{bmatrix} = \begin{bmatrix} x_1 & y_1 & 1 & 0 & 0 & 0 & -x_1x_1' & -y_1x_1' \\ 0 & 0 & 0 & x_1 & y_1 & 1 & -x_1y_1' & -y_1y_1' \\ & & & & \vdots & & & \\ x_k & y_k & 1 & 0 & 0 & 0 & -x_kx_k' & -y_kx_k' \\ 0 & 0 & 0 & x_k & y_k & 1 & -x_ky_k' & -y_ky_k' \end{bmatrix} \mathbf{a}$$

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

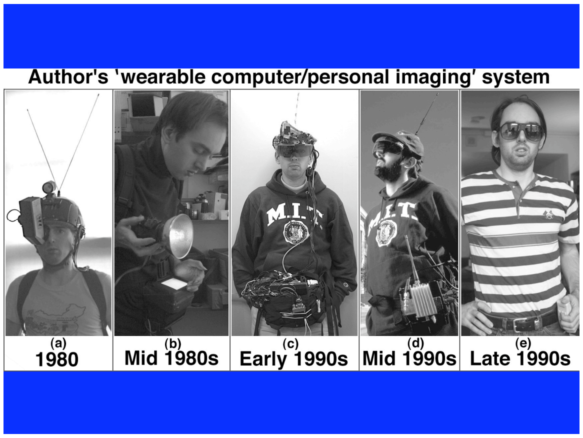
Steps in Generating A Mosaic

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

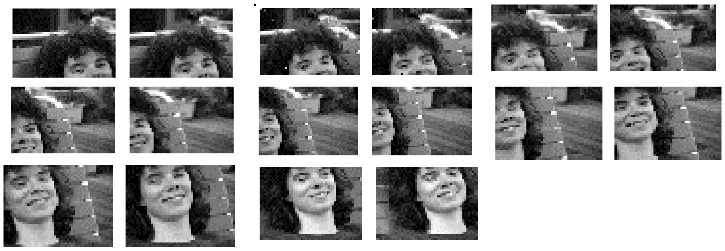
Applications of Mosaics

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

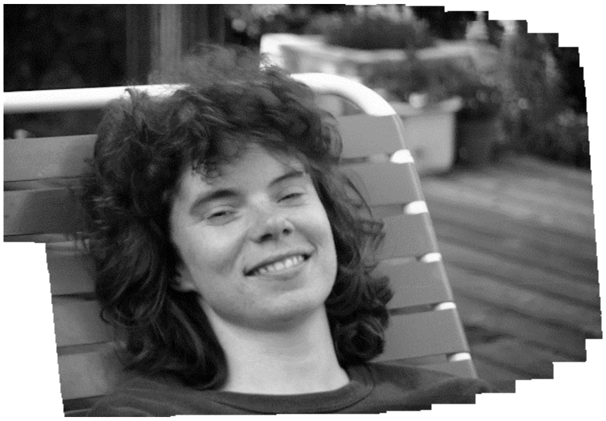
Steve Mann



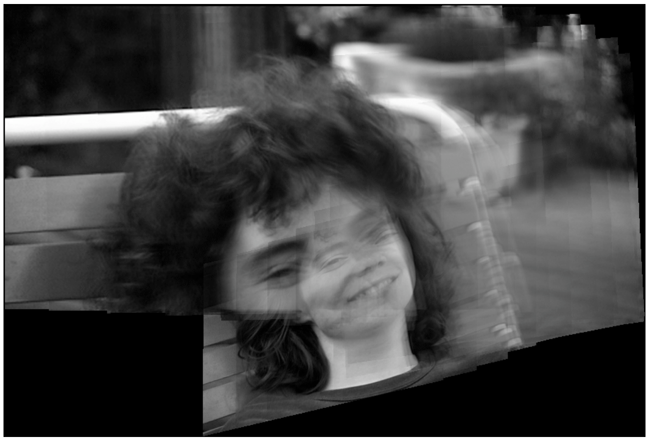
Sequence of Images



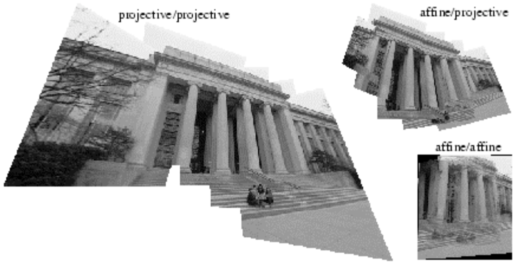
Projective Mosaic



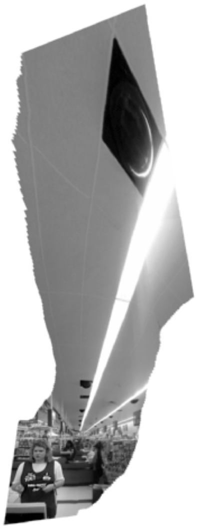
Affine Mosaic



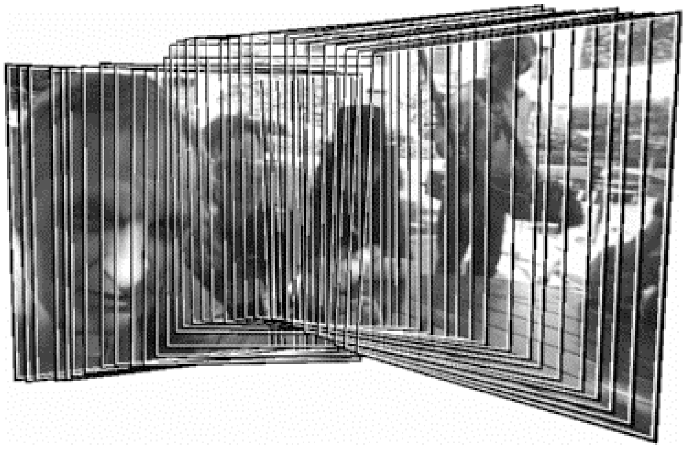
Building



Wal-Mart

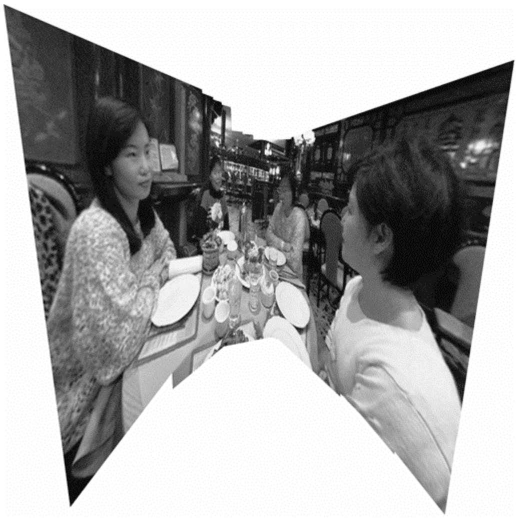


Scientific American Frontiers

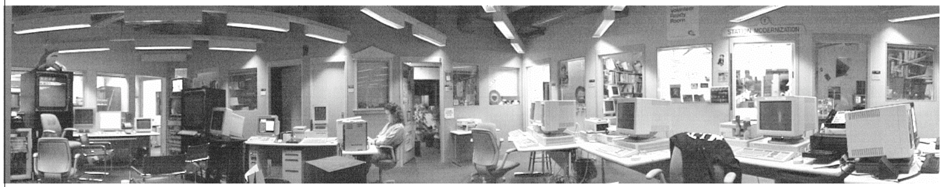


Scientific American Frontiers

Head-mounted Camera at Restaurant



MIT Media Lab



Webpages

- http://n1nlf1.eecg.toronto.edu/tip.ps.gz

  Video Orbits of the projective group, S. Mann and R. Picard.

- 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.

- M. Irani and P. Anandan, Video Indexing Based on Mosaic Representations. Proceedings of IEEE, May,1998.

- http://www.wisdom.weizmann.ac.il/~irani/abstracts/videoIndexing.html

Homework Due Sept 25

- (a) Derive linear system equation in Anandan’s method Lecture 5, page 14, top slide.
- (b) Derive equations for Mann’s method (weighted) Lecture 6, page 10.
- (c) Derive equations for Mann’s method (un-weighted) Lecture 6, page 13.

Program-1 Due Oct 2

- (a) Implement Anandan’s algorithm using affine transformation. To show the results generate a mosaic.
- (b) Implement Szeliski’s algorithm using projective transformation. To show the results generate a mosaic.
- (c) Implement Mann’s algorithm using projective transformation. To show the results generate a mosaic.
- Implement all four steps:
  - Pyramid construction
  - Motion estimation
  - Image warping
  - Coarse-to-fine refinement
- .