Lecture-7

Mann & Picard

Mann & Picard

Projective

Projective Flow (weighted)

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

Optical Flow const. equation

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

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

Projective transform

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

Projective Flow (weighted)

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

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

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

⇓ **minimize**

Projective Flow (weighted)

- (b) Homework 3 Derive this equation Due Sept 28

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

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

$$\phi^t = [f_x x, f_x y, f_x, f_y x, f_y y, f_y, xf_t - x^2 f_x - xyf_y, yf_t - xyf_x - y^2 f_y]$$

Projective Flow (unweighted)

Pseudo-Perspective

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

⇩ **Taylor Series**

$$x + u = a_1 + a_2 x + a_3 y + a_4 x^2 + a_5 xy$$

$$y + v = a_6 + a_7 x + a_8 y + a_4 xy + a_5 y^2$$

Bilinear

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

⇩ Taylor Series & remove Square terms

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

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

Projective Flow (unweighted)

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

Minimize

Bilinear and Pseudo-Perspective

$$(\sum \Phi\Phi^T)\mathbf{q} = -\sum f_t\Phi$$

(c) homewwork Derive these eqs Sept 28

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

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

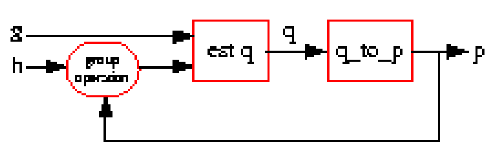
$$c_1 = x^2 f_x + xyf_x$$

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

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, y'_k)
  - estimate exact "p":



True Projective

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

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

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

$$\mathbf{a} = [a_1 \quad a_2 \quad b_1 \quad a_3 \quad a_4 \quad b_2 \quad c_1 \quad c_1]^T$$

$$\begin{bmatrix} x'_1 \\ y'_1 \\ \\ x'_k \\ y'_k \end{bmatrix} = \begin{bmatrix} x_1 & y_1 & 1 & 0 & 0 & 0 & -x_1 x'_1 & -y_1 x'_1 \\ 0 & 0 & 0 & x_1 & y_1 & 1 & -x_1 y'_1 & -y_1 y'_1 \\ \\ x_k & y_k & 1 & 0 & 0 & 0 & -x_k x' & -y_k x'_k \\ 0 & 0 & 0 & x_k & y_k & 1 & -x_k y'_k & -y_k y' \end{bmatrix} \mathbf{a}$$

$$\mathbf{P} = \mathbf{A}\mathbf{a}$$

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

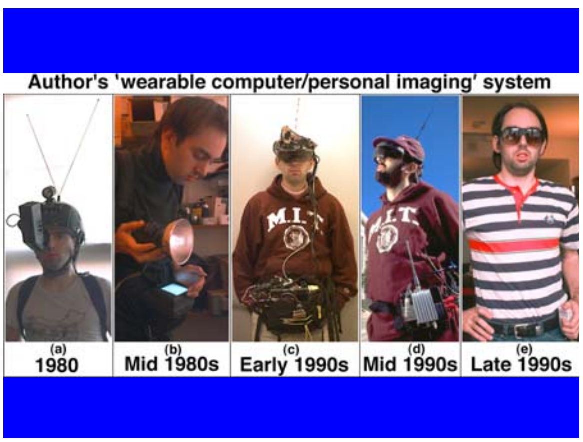
Steps in Generating A Mosaic

- 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



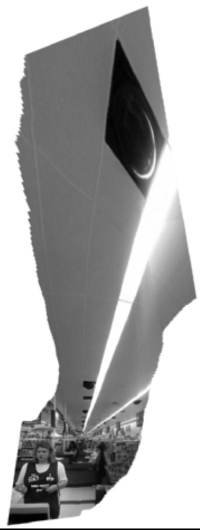
Sequence of Images

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 28

- (a) Derive linear system equation in Anandan's method Lecture 6, page 4, bottom slide.
- (b) Derive equations for Mann's method (weighted) Lecture 7, page 3 top slide.
- (c) Derive equations for Mann's method (un-weighted) Lecture 7, page 5 bottom slide.

Program-1 Due Oct 12

- (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
- .