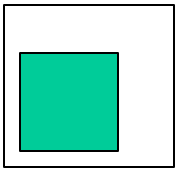
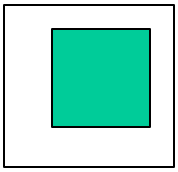
Szeliski

Projective

Projective



$f(X',t-1)$



$f(X,t)$

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

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

Szeliski (Levenberg-Marquadet)

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

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

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

min

Szeliski (Levenberg-Marquadet)

Motion Vector:

$$\mathbf{m} = [a_1 \quad a_2 \quad a_3 \quad a_4 \quad b_1 \quad b_2 \quad c_1 \quad c_2]^T$$

Szeliski (Levenberg-Marquadet)

Hessian $a_{kl} = \sum \frac{\partial e}{\partial m_k} \frac{\partial e}{\partial m_l}$ $\qquad b_k = -\sum e \frac{\partial e}{\partial m_k}$

gradient

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

•Homework 1.2 derive expressions for Hessian and gradient vector.

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

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

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

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

Hessian

$$A=\begin{bmatrix}
\sum_x \frac{\partial^2 e}{\partial a_1 \partial a_1} & \sum_x \frac{\partial^2 e}{\partial a_1 \partial a_2} & \sum_x \frac{\partial^2 e}{\partial a_1 \partial a_3} & \sum_x \frac{\partial^2 e}{\partial a_1 \partial a_4} & \sum_x \frac{\partial^2 e}{\partial a_1 \partial b_1} & \sum_x \frac{\partial^2 e}{\partial a_1 \partial b_2} & \sum_x \frac{\partial^2 e}{\partial a_1 \partial c_1} & \sum_x \frac{\partial^2 e}{\partial a_1 \partial c_2} \\
\sum_x \frac{\partial^2 e}{\partial a_2 \partial a_1} & \sum_x \frac{\partial^2 e}{\partial a_2 \partial a_2} & \sum_x \frac{\partial^2 e}{\partial a_2 \partial a_3} & \sum_x \frac{\partial^2 e}{\partial a_2 \partial a_4} & \sum_x \frac{\partial^2 e}{\partial a_2 \partial b_1} & \sum_x \frac{\partial^2 e}{\partial a_2 \partial b_2} & \sum_x \frac{\partial^2 e}{\partial a_2 \partial c_1} & \sum_x \frac{\partial^2 e}{\partial a_2 \partial c_2} \\
\sum_x \frac{\partial^2 e}{\partial a_3 \partial a_1} & \sum_x \frac{\partial^2 e}{\partial a_3 \partial a_2} & \sum_x \frac{\partial^2 e}{\partial a_3 \partial a_3} & \sum_x \frac{\partial^2 e}{\partial a_3 \partial a_4} & \sum_x \frac{\partial^2 e}{\partial a_3 \partial b_1} & \sum_x \frac{\partial^2 e}{\partial a_3 \partial b_2} & \sum_x \frac{\partial^2 e}{\partial a_3 \partial c_1} & \sum_x \frac{\partial^2 e}{\partial a_3 \partial c_2} \\
\sum_x \frac{\partial^2 e}{\partial a_4 \partial a_1} & \sum_x \frac{\partial^2 e}{\partial a_4 \partial a_2} & \sum_x \frac{\partial^2 e}{\partial a_4 \partial a_3} & \sum_x \frac{\partial^2 e}{\partial a_4 \partial a_4} & \sum_x \frac{\partial^2 e}{\partial a_4 \partial b_1} & \sum_x \frac{\partial^2 e}{\partial a_4 \partial b_2} & \sum_x \frac{\partial^2 e}{\partial a_4 \partial c_1} & \sum_x \frac{\partial^2 e}{\partial a_4 \partial c_2} \\
\sum_x \frac{\partial^2 e}{\partial b_1 \partial a_1} & \sum_x \frac{\partial^2 e}{\partial b_1 \partial a_2} & \sum_x \frac{\partial^2 e}{\partial b_1 \partial a_3} & \sum_x \frac{\partial^2 e}{\partial b_1 \partial a_4} & \sum_x \frac{\partial^2 e}{\partial b_1 \partial b_1} & \sum_x \frac{\partial^2 e}{\partial b_1 \partial b_2} & \sum_x \frac{\partial^2 e}{\partial b_1 \partial c_1} & \sum_x \frac{\partial^2 e}{\partial b_1 \partial c_2} \\
\sum_x \frac{\partial^2 e}{\partial b_2 \partial a_1} & \sum_x \frac{\partial^2 e}{\partial b_2 \partial a_2} & \sum_x \frac{\partial^2 e}{\partial b_2 \partial a_3} & \sum_x \frac{\partial^2 e}{\partial b_2 \partial a_4} & \sum_x \frac{\partial^2 e}{\partial b_2 \partial b_1} & \frac{\partial^2 e}{\partial b_2 \partial b_2} & \sum_x \frac{\partial^2 e}{\partial b_2 \partial c_1} & \sum_x \frac{\partial^2 e}{\partial b_2 \partial c_2} \\
\sum_x \frac{\partial^2 e}{\partial c_1 \partial a_1} & \sum_x \frac{\partial^2 e}{\partial c_1 \partial a_2} & \sum_x \frac{\partial^2 e}{\partial c_1 \partial a_3} & \sum_x \frac{\partial^2 e}{\partial c_1 \partial a_4} & \sum_x \frac{\partial^2 e}{\partial c_1 \partial b_1} & \sum_x \frac{\partial^2 e}{\partial c_1 \partial b_2} & \sum_x \frac{\partial^2 e}{\partial c_1 \partial c_1} & \sum_x \frac{\partial^2 e}{\partial c_1 \partial c_2} \\
\sum_x \frac{\partial^2 e}{\partial c_2 \partial a_1} & \sum_x \frac{\partial^2 e}{\partial c_2 \partial a_2} & \sum_x \frac{\partial^2 e}{\partial c_2 \partial a_3} & \sum_x \frac{\partial^2 e}{\partial c_2 \partial a_4} & \sum_x \frac{\partial^2 e}{\partial c_2 \partial b_1} & \sum_x \frac{\partial^2 e}{\partial c_2 \partial b_2} & \sum_x \frac{\partial^2 e}{\partial c_2 \partial c_1} & \sum_x \frac{\partial^2 e}{\partial c_2 \partial c_2}
\end{bmatrix}$$

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

Szeliski (Levenberg-Marquadet)

- Start with some initial value of m, and ë=.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 $\frac{\partial e}{\partial m_k} = \frac{\partial f}{\partial x'}\frac{\partial x'}{\partial m_k} + \frac{\partial f}{\partial y'}\frac{\partial y'}{\partial m_k}$

Szeliski (Levenberg-Marquadet)

-Compute A **and** b

-Solve system

$$(A - lI)\Delta m = b$$

-Update

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

Szeliski (Levenberg-Marquadet)

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

Mann & Picard

Projective

Projective Flow (weighted)

$$u_f f_x + v_f f_y + f_t = 0$$

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

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

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

Projective Flow (weighted)

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

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

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

⇩ **minimize**

Projective Flow (weighted)

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

$$a = [a_{11}, a_{12}, b_1, a_{21}, a_{22}, b_2, c_1, c_2]^T$$

$$f^t = [f_x x,\ f_x y,\ f_x,\ f_y x,\ f_y y,\ f_y,\ x f_t - x^2 f_x - xy f_y,\ y f_t - xy f_x - y^2 f_y]$$

Projective Flow (unweighted)

Bilnear

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

Taylor Series

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

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

Pseudo-Perspective

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

Taylor Series

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

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

Projective Flow (unweighted)

$$e_{flow} = \sum (\mathbf{u_m^T f_X} + f_t)^2$$

⇩ **Minimize**

Bilinear and Pseudo-Perspective

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

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

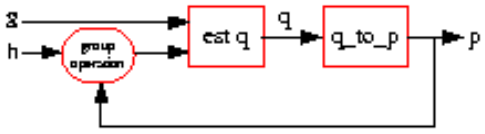
$$\Phi^T = \left[f_x(x, y, 1) \quad f_y(x, y, 1) \quad c_1 \quad c_2 \right]$$ **Pseudo p**

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

$$c_2 = xy f_x + y^2 f_y$$

Algorithm

- 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

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

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

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

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

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

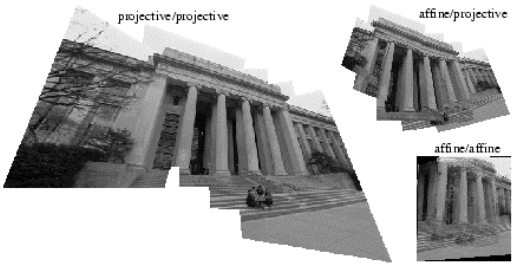
Steve Mann

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.