Differential Motion Analysis

Ying Wu

Electrical Engineering and Computer Science
Northwestern University, Evanston, IL 60208

yingwu@ece.northwestern.edu

http://www.eecs.northwestern.edu/~yingwu

July 19, 2010

Outline

Brightness Constancy and Optical Flow

- Optical flow: the apparent motion of the brightness pattern
- Optical flow $\neq$ motion field
- Denote an image by $I(x, y, t)$, and the velocity of a pixel $\mathbf{m} = [x, y]^T$ is

$$\mathbf{v}_m = \dot{\mathbf{m}} = [v_x, v_y]^T = \begin{bmatrix} dx/dt \\ dy/dt \end{bmatrix}$$

- Brightness constancy: the intensity of $\mathbf{m}$ keeps the same during dt, i.e.,

$$I(x + v_x dt, y + v_y dt, t + dt) = I(x, y, t)$$

- Optical flow constraint:

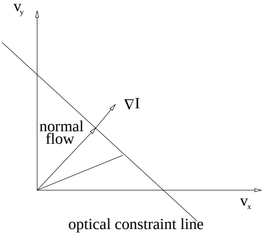
$$\frac{\partial I}{\partial x} v_x + \frac{\partial I}{\partial y} v_y + \frac{\partial I}{\partial t} = 0$$

i.e.

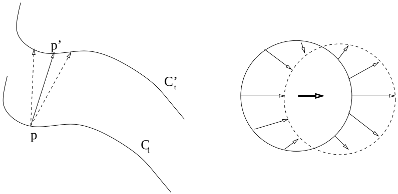
$$\nabla I \cdot \mathbf{v}_m + \frac{\partial I}{\partial t} = 0$$

The Aperture Problem

- For each pixel, one constraint equation, but two unknowns.
- Normal flow



- Aperture problem: the motion along the direction perpendicular to the image gradient cannot be determined



- Other constraints are needed.

Lucas-Kanade's Method

- Assume: a constant motion for a small image patch Ω.
- Define a weight function $W(\mathbf{m}), \mathbf{m} \in \Omega$, for the pixels.
- Weighted LS formulation

$$\min_{\mathbf{v}} E = \sum_{\mathbf{m}\in\Omega} W^2(\mathbf{m}) \left(\nabla I \cdot \mathbf{v} + \frac{\partial I}{\partial t} \right)^2$$

- WLS solution:

$$\mathbf{v} = (\mathbf{A}^T \mathbf{W}^2 \mathbf{A})^{-1} \mathbf{A}^T \mathbf{W}^2 \mathbf{b}$$

$$\mathbf{A} = \begin{bmatrix} \frac{\partial I_1}{\partial x_1} & \frac{\partial I_1}{\partial y_1} \\ \vdots & \vdots \\ \frac{\partial I_N}{\partial x_N} & \frac{\partial I_N}{\partial y_N} \end{bmatrix}, \quad \mathbf{W} = diag(W(\mathbf{m}_1), \ldots, W(\mathbf{m}_N))$$

$$\mathbf{v} = \left[\frac{\partial x}{\partial t}, \frac{\partial y}{\partial t} \right]^T = [v_x, v_y]^T, \ \ \mathbf{b} = -\left[\frac{\partial I_1}{\partial t}, \ldots, \frac{\partial I_N}{\partial t} \right]^T$$

- the intersection of all the flow constraint lines corresponding to the pixels in Ω.

Horn-Schunck's Method

- Assume: flow varies smoothly $\longleftarrow$ global regularization
- The measure of departure from smoothness can be written by:

$$\begin{aligned} e_s &= \int\int (||\nabla v_x||^2 + ||\nabla v_y||^2) dxdy \\ &= \int\int (\frac{\partial v_x}{\partial x})^2 + (\frac{\partial v_x}{\partial y})^2 + (\frac{\partial v_y}{\partial x})^2 + (\frac{\partial v_y}{\partial y})^2 dxdy \end{aligned}$$

- The error of optical flow is:

$$e_c = \int\int (\nabla I \cdot \mathbf{v}_m + \frac{\partial I}{\partial t})^2 dxdy$$

- Objective function:

$$\begin{aligned} e &= e_c + \lambda e_s \\ &= \int\int (\nabla I \cdot \mathbf{v}_m + \frac{\partial I}{\partial t})^2 + \lambda(||\nabla v_x||^2 + ||\nabla v_y||^2) dxdy \end{aligned}$$

Horn-Schunck's Method

- Fixed-point iteration:

$$
\begin{aligned}
v_x^{k+1} &= \bar{v}_x^k - \left[\frac{\left(\frac{\partial I}{\partial x}\right) \bar{v}_x^k + \left(\frac{\partial I}{\partial y}\right) \bar{v}_y^k + \frac{\partial I}{\partial t}}{\lambda + \left(\frac{\partial I}{\partial x}\right)^2 + \left(\frac{\partial I}{\partial y}\right)^2} \right] \frac{\partial I}{\partial x} \\
v_y^{k+1} &= \bar{v}_y^k - \left[\frac{\left(\frac{\partial I}{\partial x}\right) \bar{v}_x^k + \left(\frac{\partial I}{\partial y}\right) \bar{v}_y^k + \frac{\partial I}{\partial t}}{\lambda + \left(\frac{\partial I}{\partial x}\right)^2 + \left(\frac{\partial I}{\partial y}\right)^2} \right] \frac{\partial I}{\partial y}
\end{aligned}
$$

- Concisely, it is:

$$\mathbf{v}^{k+1} = \bar{\mathbf{v}}^k - \alpha(\nabla I)$$

- In each iteration, the new optical flow field is constrained by its local average and the optical flow constraints.

Parametric Flow: Affine flow

- Affine model is under two assumptions:
  - planar surface
  - orthographic projection

- We can write a 3D plane by $Z = AX + BY + C$. Then we have the 6-parameter affine flow model:

$$\begin{bmatrix} v_x \\ v_y \end{bmatrix} = \begin{bmatrix} a_1 & a_2 \\ a_3 & a_4 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} a_5 \\ a_6 \end{bmatrix}$$

- In this case, the flow can be determined by at least 3 points.

Parametric Flow: Quadratic flow

- Quadratic model is under two assumptions:
  - planar surface
  - perspective projection

- Under perspective projection, a plane can be written as

$$\frac{1}{Z} = \frac{1}{C} - \frac{A}{C}X - \frac{B}{C}Y$$

- So, we have

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

- In this case, if we know at least 4 points on a planar object, we can also $\{a_1, \ldots, a_8\}$.

Parametric Flow: Parametric flow fitting

- LS formulation

$$\min_{\theta} \sum_{\Omega} ||I(x + v_x(\Theta)dt, y + v_y(\Theta)dt, t + dt) - I(x, y, t)||^2$$

- or

$$\min_{\theta} \sum_{\Omega} \left[\nabla I^T \mathbf{v}(\Theta) + I_t\right]^2$$

- denote by $\nabla I_\theta = \nabla I^T \nabla \mathbf{v}(\Theta)$,

$$\min_{\theta} \sum_{\Omega} \left[\nabla I_\theta^T \Theta + I_t\right]^2$$

- Easy to figure out the LS solution.

Exercises

- **Exercise 1**: Recovering rotation
  Assume the motion is a pure rotation, i.e.,

$$\begin{bmatrix} x_2 \\ y_2 \end{bmatrix} = R(\theta) \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \end{bmatrix}$$

$$\min_{\theta} \sum \left[I\left(R(\theta) \begin{bmatrix} x_1 \\ y_1 \end{bmatrix}, t + dt\right) - I(x, y, t) \right]^2$$

- **Exercise 2**: Recovering 2D affine motion

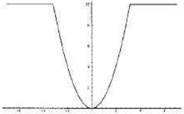
$$\begin{bmatrix} v_x \\ v_y \end{bmatrix} = \begin{bmatrix} a_1 & a_2 \\ a_3 & a_4 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} a_5 \\ a_6 \end{bmatrix}$$

$$\min_{A} \sum ||\nabla I^T \mathbf{v}(A) + I_t||^2$$

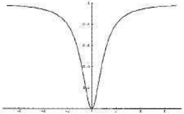
Robust Flow Computation[1]

- Motivation
  - Brightness constancy $\stackrel{\text{violation}}{\longleftarrow}$ specular reflection
  - Spatial smoothness $\stackrel{\text{violation}}{\longleftarrow}$ motion discontinuities
- Outliers ruin LS estimation
- One solution: influence function $\rho(x, \sigma)$

$$\rho(x, \alpha, \lambda) = \begin{cases} \lambda x^2 & \text{if } |x| < \frac{\sqrt{\alpha}}{\sqrt{\lambda}}, \\ \alpha & \text{otherwise.} \end{cases}$$



$$\rho(x, \sigma) = \frac{x^2}{\sigma + x^2}$$



- Applying influence function to flow estimation

$$\min_v \sum_{\Omega} \rho(I(x, y, t) - I(x + v_x dt, y + v_y dt, t + dt), \sigma)$$

$$\min_v \sum_{\Omega} \rho_c(\nabla I^T \mathbf{v}(\theta) + I_t, \sigma_c) + \lambda[\rho_s(v_x, \sigma_s) + \rho_s(v_y, \sigma_s)]$$

[1] Michael Black and P. Anandan, "The Robust Estimation of Multiple Motions: Parametric and Piecewise-Smooth Flow Fields", CVIU, vol.63. no.1, pp.75-104, 1996

Multi-frame Optical Flow[2]

- For a static scene, the flow induced by camera motion in multiple frames lie in a low-dimensional subspace
- Example: 3D planar scene under orthographic projection

$$\begin{bmatrix} v_x \\ v_y \end{bmatrix} = \begin{bmatrix} a_5 & a_1 & a_2 \\ a_6 & a_3 & a_4 \end{bmatrix} \begin{bmatrix} 1 \\ x \\ y \end{bmatrix}$$

- We have F frames and all have the same N points. Denote by $[v_x^{ij} \; v_y^{ij}]^T$ the flow for the i-th point at the j-th frame, w.r.t. a reference frame. Collect all the flows:

$$\mathbf{U} = \begin{bmatrix} v_x^{11} & v_x^{21} & \dots & v_x^{N1} \\ \vdots & \vdots & \dots & \vdots \\ v_x^{1F} & v_x^{2F} & \dots & v_x^{NF} \end{bmatrix} = \begin{bmatrix} a_5^1 & a_1^1 & a_2^1 \\ \vdots & \vdots & \vdots \\ a_5^F & a_1^F & a_2^F \end{bmatrix} \begin{bmatrix} 1 & 1 & \dots & 1 \\ x_1 & x_2 & \dots & x_N \\ y_1 & y_2 & \dots & y_N \end{bmatrix}$$

[2] Michal Irani, "Multi-Frame Optical Flow Estimation Using Subspace Constraints", ICCV'99

Multi-frame Optical Flow

- And

$$\mathbf{V} = \begin{bmatrix} v_y^{11} & v_y^{21} & \dots & v_y^{N1} \\ \vdots & \vdots & \dots & \vdots \\ v_y^{1F} & v_y^{2F} & \dots & v_y^{NF} \end{bmatrix} = \begin{bmatrix} a_6^1 & a_3^1 & a_4^1 \\ \vdots & \vdots & \vdots \\ a_6^F & a_3^F & a_4^F \end{bmatrix} \begin{bmatrix} 1 & 1 & \dots & 1 \\ x_1 & x_2 & \dots & x_N \\ y_1 & y_2 & \dots & y_N \end{bmatrix}$$

- It is clear that

$$\texttt{rank} \begin{bmatrix} \mathbf{U} \\ \mathbf{V} \end{bmatrix} \leq 3$$

$$\texttt{rank} \begin{bmatrix} \mathbf{U} & \mathbf{V} \end{bmatrix} \leq 6$$

Outline

Considering lighting models

- Brightness constancy assumption is too restrictive
- Are there constraints for lighting?
- For a pure Lambertain surface
  - if no shadowing, then all images under varying illumination lie in a 3-D subspace in $\mathbb{R}^N$
  - with shadowing, the dimension will be higher, but we may learn it
- The subspace can be learnt from a set of training images by PCA, so we have the basis $\mathbf{B} = [B_1, B_2, \ldots, B_m]$ (note: $\mathbf{B}^T\mathbf{B} = \mathbf{I}$).
- Then the appearance of the template at t is modeled by

$$I(x, y, t) + \mathbf{B}\Lambda, \qquad \textit{where } \Lambda = \begin{bmatrix} \lambda_1 \\ \vdots \\ \lambda_m \end{bmatrix}$$

Considering lighting models[3]

- therefore, we have

$$
\begin{aligned}
E(\Theta,\Lambda) &= \sum_{\Omega} ||I(x+v_x(\Theta)dt, y+v_y(\Theta)dt, t+dt) - I(x,y,t) - \mathbf{B}\Lambda||^2 \\
\textit{or} \quad E(\Theta,\Lambda) &= \sum_{\Omega} ||\nabla^T \mathbf{v}(\Theta) + I_t - \mathbf{B}\Lambda||^2 \\
\textit{or} \quad E(\Theta,\Lambda) &= \sum_{\Omega} ||\nabla_\theta^T \Theta + I_t - \mathbf{B}\Lambda||^2
\end{aligned}
$$

- denote by $\nabla I_\theta^T = \mathbf{M}$, we have

$$
\begin{bmatrix} \mathbf{M} & -\mathbf{B} \end{bmatrix} \begin{bmatrix} \Theta \\ \Lambda \end{bmatrix} = -I_t
$$

[3] Gregory Hager and Peter Belhumeur, "Real-Time Tracking of Image Regions with Changes in Geometry and Illumination", CVPR'96

Considering lighting models

- So, we have

$$\begin{bmatrix}\Theta \\ \Lambda\end{bmatrix} = \begin{bmatrix}\mathbf{M} & -\mathbf{B}\end{bmatrix}^{\dagger} I_t$$

$$= -\begin{bmatrix}\mathbf{M}^T\mathbf{M} & -\mathbf{M}^T\mathbf{B} \\ -\mathbf{B}^T\mathbf{M} & \mathbf{B}^T\mathbf{B}\end{bmatrix}^{-1} \begin{bmatrix}\mathbf{M}^T \\ -\mathbf{B}^T\end{bmatrix} I_t$$

- easy to see

$$\Theta = -\left[\mathbf{M}^T(\mathbf{I} - \mathbf{B}\mathbf{B}^T)\mathbf{M}\right]^{-1} \mathbf{M}^T(\mathbf{I} + \mathbf{B}\mathbf{B}^T) I_t$$

Considering appearance variations

- In-class appearance variations
- Low-level matching $\longrightarrow$ high-level matching
- If we know the target, we may learn its appearance variations
- We may build a classifier for matching

$$\min_{v} S\left(I(u + v_x dt, v + v_y dt, t + dt) : \Lambda\right),$$

where Λ are parameters of the classifier
- E.g., using an SVM classifier

$$\sum_{j=1}^{n} y_j \alpha_j k(\mathbf{I}, \mathbf{x}_j) + b$$

- Let's maximize the SVM matching score

$$\max_{u,v} \sum_{j=1}^{n} y_j \alpha_j k(I + uI_x + vI_y, \mathbf{x}_j)$$

Considering appearance variations[4]

- Let use a 2nd order polynomial kernel $k(\mathbf{x}, \mathbf{x}_j) = (\mathbf{x}^T \mathbf{x}_j)^2$
- so, we have

$$E(u, v) = \sum_{j=1}^{n} y_j \alpha_j \left[(I + uI_x + vI_y)^T \mathbf{x}_j \right]^2$$

$$\begin{aligned} \frac{\partial E}{\partial u} &= \sum y_j \alpha_j I_x^T \mathbf{x}_j (I + uI_x + vI_y)^T \mathbf{x}_j = 0 \\ \frac{\partial E}{\partial v} &= \sum y_j \alpha_j I_y^T \mathbf{x}_j (I + uI_x + vI_y)^T \mathbf{x}_j = 0 \end{aligned}$$

- the solution is:

$$\begin{bmatrix} \sum \alpha_j y_j (\mathbf{x}_j^T I_x)^2 & \sum \alpha_j y_j (\mathbf{x}_j^T I_x)(\mathbf{x}_j I_y) \\ \sum \alpha_j y_j (\mathbf{x}_j^T I_x)(\mathbf{x}_j^T I_y) & \sum \alpha_j y_j (\mathbf{x}_j^T I_y)^2 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} -\sum \alpha_j y_j (\mathbf{x}_j^T I_x)(\mathbf{x}_j^T I) \\ -\sum \alpha_j y_j (\mathbf{x}_j^T I_y)(\mathbf{x}_j^T I) \end{bmatrix}$$

[4] Shai Avidan, "Subset Selection for Efficient SVM Tracking", CVPR'03

Spatial-appearance model (SAM)[5]

- Denote by $\mathbf{y} = [\mathbf{x} \ c(\mathbf{x})]$, where $\mathbf{x}$ is the location and c color
- Assume a Gaussian component be factorized

$$g(\mathbf{y}; \mu_k, \Sigma_k) = g(\mathbf{x}; \mu_{ks}, \Sigma_{ks}) g(c(\mathbf{x}); \mu_{kc}, \Sigma_{kc})$$

- For a pixel, the likelihood is a mixture

$$p(\mathbf{y}|\Theta) = \sum_{k=1}^{K} p_k g(\mathbf{y}; \mu_k, \Sigma_k)$$

- Let's use an affine motion here

$$T(\mathbf{x}; a_t) = \begin{bmatrix} a_{1t} & a_{2t} \\ a_{3t} & a_{4t} \end{bmatrix} \mathbf{x} + \begin{bmatrix} a_{5t} \\ a_{6t} \end{bmatrix}$$

- then, we have

$$p(T(\mathbf{y}; a_t)|\Theta) = p(T(\mathbf{x}; a_t), c(T(\mathbf{x}; a_t))|\Theta)$$

$$= \sum_{k=1}^{K} p_k g(\mathbf{x}; \mu_{ks}, \Sigma_{ks}) g(c(T(\mathbf{x}; a_t)); \mu_{kc}, \Sigma_{kc}) \triangleq \sum_{k=1}^{K} q(k, \mathbf{y}_i; a_t)$$

[5] Ting Yu and Ying Wu, "Differential Tracking based on Spatial-Appearance Model (SAM)", CVPR'06

Spatial-appearance model (SAM)

- For an image region

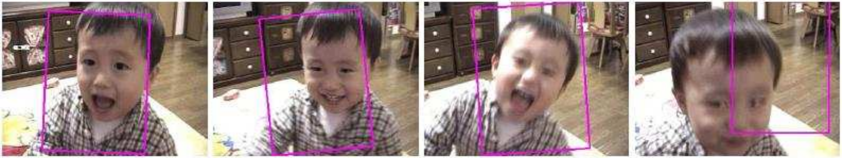
$$E(a_t;\Theta) = \sum_{x_i \in \Omega} \log p(T(\mathbf{y}_i; a_t)|\Theta) = \sum_{x_i \in \Omega} \log \left(\sum_{k=1}^{K} q(k, \mathbf{y}_i; a_t) \right)$$

- our task is to

$$\max_{a_t} E(a_t;\Theta)$$

- Solution: similar to the general EM algorithm

Spatial-appearance model (SAM)



(a) tracking with template matching.

(b) differential tracking via SAM, iterative motion estimations of each frame.

(c) differential tracking via SAM, final tracking result of each frame overlapped by spatial mixture components.

Outline

Representation

- The target is represented by a feature histogram $\mathbf{q} = [q_1, q_2, \dots, q_m]^T \in \mathbb{R}^m$, where

$$q_u = \frac{1}{C} \sum_{i=1}^{n} K(\mathbf{s}_i - \mathbf{x}) \delta(b(\mathbf{s}_i), u)$$

- its matrix form

$$\mathbf{q}(\mathbf{x}) = \mathbf{U}^T \mathbf{K}(\mathbf{x})$$

$$\mathbf{U} = \begin{bmatrix} \delta(b(\mathbf{s}_1), u_1) & \dots & \delta(b(\mathbf{s}_1), u_m) \\ \vdots & \vdots & \vdots \\ \delta(b(\mathbf{s}_n), u_1) & \dots & \delta(b(\mathbf{s}_n), u_m) \end{bmatrix} \in \mathbb{R}^{n \times m}, \quad \mathbf{K} = \frac{1}{C} \begin{bmatrix} K(\mathbf{s}_1 - \mathbf{x}) \\ \vdots \\ K(\mathbf{s}_n - \mathbf{x}) \end{bmatrix} \in \mathbb{R}^n.$$

- Kernel profile: $K(x) = k(||x||^2)$
- denote by $g(x) = -k'(x)$

Formulation

- The target is initially at $\mathbf{x}$
- to find the optimal motion $\Delta\mathbf{x}^*$ by

$$\min_{\Delta x} O(\mathbf{q}, \mathbf{p}(\mathbf{x} + \Delta\mathbf{x}))$$

  where $\mathbf{q}$ is the target model, and $\mathbf{p}$ is the image observation
- choices of $O(\cdot, \cdot)$
  - Bhattachayya coefficient

$$O_B(\Delta\mathbf{x}) \triangleq -\langle\sqrt{\mathbf{q}}, \sqrt{\mathbf{p}(\mathbf{x} + \Delta\mathbf{x})}\rangle = -\sqrt{\mathbf{q}}^T \sqrt{\mathbf{p}(\mathbf{x} + \Delta\mathbf{x})}.$$

  - Matusita metric

$$O_M(\Delta\mathbf{x}) \triangleq \|\sqrt{\mathbf{q}} - \sqrt{\mathbf{p}(\mathbf{x} + \Delta\mathbf{x})}\|^2.$$

Mean-shift tracking[6]

- $O_B = -\sum_{u=1}^{m} \sqrt{p_u(\mathbf{x}+\Delta\mathbf{x}) q_u}$
- first order approximation

$$-O_B(\Delta\mathbf{x}) = \frac{1}{2}\sum_{u=1}^{m}\sqrt{p_u(\mathbf{x})q_u} + \frac{1}{2C}\sum_{i=1}^{n} w_i K(||\frac{\mathbf{x}+\Delta\mathbf{x}-\mathbf{s}_i}{h}||^2)$$

where $w_i = \sum_{u=1}^{m}\delta(b(\mathbf{s}_i), u)\sqrt{\frac{q_u}{p_u(\mathbf{x})}}$ is the weight for $\mathbf{s}_i$

[6] D. Comaniciu, V. Ramesh and P. Meer, "Real-Time Tracking of Non-Rigid Objects using Mean Shift", CVPR'00

Mean-shift tracking

- So, we have

$$\min_{\Delta x} O_B(\Delta \mathbf{x}) \Longrightarrow \max_{\Delta x} \sum_{i=1}^{n} w_i K(||\frac{\mathbf{x} + \Delta \mathbf{x} - \mathbf{s}_i}{h}||^2)$$

- The solution is an iterative mean-shift procedure

$$\mathbf{x}' \longleftarrow \frac{\sum_{i=1}^{n} \mathbf{s}_i w_i g(||\frac{\mathbf{x} - \mathbf{s}_i}{h}||^2)}{\sum_{i=1}^{n} w_i g(||\frac{\mathbf{x} - \mathbf{s}_i}{h}||^2)}$$

SSD kernel-based tracking[7]

- let's use $O_M(\Delta\mathbf{x}) = \|\sqrt{\mathbf{q}} - \sqrt{\mathbf{p}(\mathbf{x}+\Delta\mathbf{x})}\|^2$

- Linearization

$$\sqrt{\mathbf{p}(\mathbf{x}+\Delta\mathbf{x})} \approx \sqrt{\mathbf{p}(\mathbf{x})} + \frac{1}{2} d(\mathbf{p}(\mathbf{x}))^{-\frac{1}{2}} \mathbf{U}^T \mathbf{J}_k(\mathbf{x}) \Delta\mathbf{x}$$

where

$$d(\mathbf{p}(\mathbf{x})) = diag(p_1(\mathbf{x}), \ldots, p_m(\mathbf{x}))$$

$$\mathbf{J}_k = \begin{bmatrix} \frac{\partial \mathbf{K}}{\partial u} & \frac{\partial \mathbf{K}}{\partial v} \end{bmatrix} = \begin{bmatrix} \nabla_c K(\mathbf{s}_1 - \mathbf{x}) \\ \vdots \\ \nabla_c K(\mathbf{s}_n - \mathbf{x}) \end{bmatrix}$$

[7] G. Hager, M. Dewan and C. Stewart, "Multiple Kernel Tracking with SSD", CVPR'04

SSD kernel-based tracking

- So the objective function is

$$O_M(\Delta\mathbf{x}) = ||\sqrt{\mathbf{q}} - \sqrt{\mathbf{p}(\mathbf{x})} - \frac{1}{2}d(\mathbf{p}(\mathbf{x}))^{-\frac{1}{2}}\mathbf{U}^T\mathbf{J}_k(\mathbf{x})\Delta\mathbf{x}||^2$$

- Denote $\mathbf{M}(\mathbf{x}) = \frac{1}{2}d(\mathbf{p}(\mathbf{x}))^{-\frac{1}{2}}\mathbf{U}^T\mathbf{J}_k(\mathbf{x})$

- we have a linear system

$$\mathbf{M}\Delta\mathbf{x} = \sqrt{\mathbf{q}} - \sqrt{\mathbf{p}(\mathbf{x})}$$

- the solution is clear

$$\Delta\mathbf{x} = \mathbf{M}^{\dagger}(\sqrt{\mathbf{q}} - \sqrt{\mathbf{p}(\mathbf{x})})$$

Singularities

- It is clear that $\mathbf{M}$ is in the form of

$$\mathbf{M} = \begin{bmatrix} d_x^1 & d_y^1 \\ \vdots & \vdots \\ d_x^m & d_y^m \end{bmatrix}$$

where

$$\begin{bmatrix} d_x^j & d_y^j \end{bmatrix} = \left[\frac{1}{2\sqrt{\mathbf{p}}_j} \sum_i (\mathbf{s}_i^j - \mathbf{x}) g \left(\left\| \frac{\mathbf{s}_i^j - \mathbf{x}}{h} \right\|^2 \right) \right]$$

which is the center of mass for feature j.

- If $\left\{ \begin{bmatrix} d_x^j & d_y^j \end{bmatrix}, j = 1, \ldots, m \right\}$ are linearly dependent, then $\texttt{rank}(\mathbf{M}) = 1$ and the solution is not unique.

Optimal Kernel Placement[8]

- Different image regions have different properties. Some of them are singular, and some are far from singular.
- How can we find those that are far from singular?
- Checking the property of $\mathbf{M}$.
- The Schatten 1-norm: $\|\mathbf{A}\|_S = \sum \sigma_i$
- The S-norm condition number

$$\kappa_S(\mathbf{A}) = (\sum \sigma_i)^2 / \prod \sigma_i$$

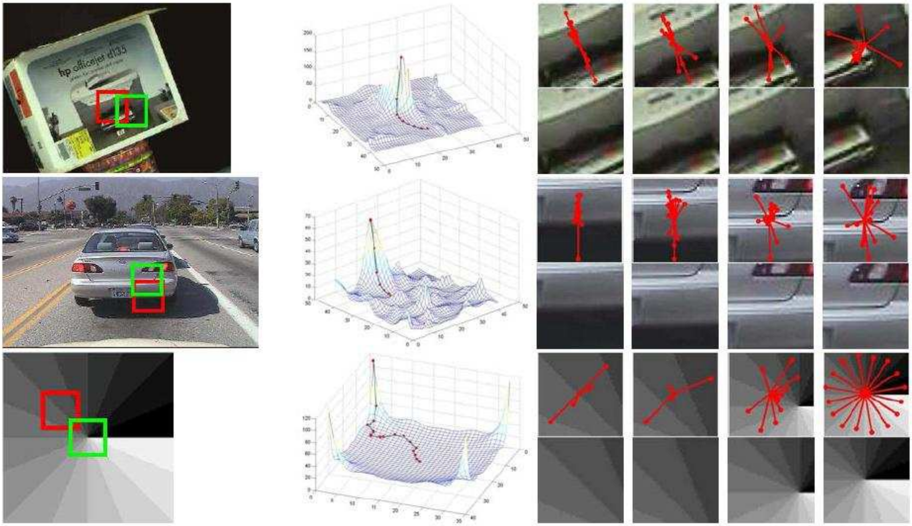
- we can compute in a closed form

$$\kappa_S(\mathbf{M}^T\mathbf{M}) = \|(\mathbf{M}^T\mathbf{M})\|_S \|(\mathbf{M}^T\mathbf{M})\|_S = \frac{(\sum (d_x^j)^2 + \sum (d_y^j)^2)^2}{\sum (d_x^j)^2 \sum (d_y^j)^2 - (\sum (d_x^j d_y^j))^2}$$

- exhaustive search v.s. gradient-based search

[8] Zhimin Fan, Ming Yang, Ying Wu, Gang Hua and Ting Yu, Efficient Optimal Kernel Placement for Reliable Visual Tracking", CVPR'06

Optimal Kernel Placement



Kernel Concatenation

- Concatenate multiple kernels to increase the dimensionality of measurement
- the same as using more features
- a set of K kernels $\mathbf{p}_i(\mathbf{x}) = \mathbf{U}^T \mathbf{K}_i(\mathbf{x})$
- stacking histograms into $\bar{\mathbf{p}}$ and $\bar{\mathbf{q}}$.
- the objective function is

$$\min_{\Delta x} \sum_{i=1}^{K} ||\sqrt{\mathbf{q}} - \sqrt{\mathbf{p}_i(\mathbf{x} + \Delta \mathbf{x})}||^2$$

- easy to see the solution

$$\mathbf{M} \Delta \mathbf{x} = \sqrt{\bar{\mathbf{q}}} - \sqrt{\bar{\mathbf{p}}_i(\mathbf{x})}$$

where

$$\mathbf{M} = \frac{1}{2} \mathrm{d}(\overline{\mathbf{p}})^{-\frac{1}{2}} \begin{bmatrix} \mathbf{U}^T & & \\ & \ddots & \\ & & \mathbf{U}^T \end{bmatrix} \begin{bmatrix} \mathbf{J}_{\mathbf{K}_1} \\ \vdots \\ \mathbf{J}_{\mathbf{K}_w} \end{bmatrix}$$

Kernel Combination

- Aggregating histograms to produce new features

$$\overline{\mathbf{q}} = \sum_{i=1}^{K} \mathbf{U}^T \mathbf{K}_i, \quad \overline{\mathbf{p}} = \sum_{i=1}^{K} \mathbf{U}^T \mathbf{K}_i(\mathbf{c}).$$

- The objective function is

$$\min_{\Delta x} \| \sqrt{\sum_{i=1}^{K} \mathbf{q}} - \sqrt{\sum_{i=1}^{K} \mathbf{p}_i(\mathbf{x} + \Delta \mathbf{x})} \|^2$$

- The corresponding linear system is:

$$\sqrt{\sum_{i=1}^{K} \mathbf{q}_i} - \sqrt{\sum_{i=1}^{K} \mathbf{p}_i(\mathbf{x})} = \mathbf{M}\Delta\mathbf{x},$$

  where

$$\mathbf{M} = \frac{1}{2} \mathrm{d}(\overline{\mathbf{p}})^{-\frac{1}{2}} \mathbf{U}^T \sum_{i=1}^{K} \mathbf{J}_{\mathbf{K}_i} = \sum_{i=1}^{K} \mathbf{M}_i$$

Collaborative Multiple Kernels[9]

- Relaxed motion representation $\underline{\Delta \mathbf{x}} = \begin{bmatrix} \Delta \mathbf{x}_1 \\ \vdots \\ \Delta \mathbf{x}_k \end{bmatrix}$
- Consider a structural constrain

$$\Omega(\mathbf{x}_1, \ldots, \mathbf{x}_k) = 0$$

- Objective function

$$O(\mathbf{x}_1, \ldots, \mathbf{x}_k) = \sum_{i=1}^{k} ||\sqrt{\mathbf{q}_i} - \sqrt{\mathbf{p}_i(\mathbf{x}_i)}||^2 + \gamma ||\Omega(\mathbf{x}_1, \ldots, \mathbf{x}_k)||^2$$

- This is equivalent to a linear system

$$\begin{cases} I & = & \mathbf{G}\underline{\Delta \mathbf{x}} + \omega_1 \\ \mathbf{y} & = & \mathbf{M}\underline{\Delta \mathbf{x}} + \omega_2 \end{cases},$$

[9] Zhimin Fan, Ying Wu and Ming Yang, "Multiple Collaborative Kernel Tracking", CVPR'05

Collaborative Multiple Kernels

- where

$$\mathbf{y} = \begin{bmatrix} \sqrt{\mathbf{q}_1} - \sqrt{\mathbf{p}(\mathbf{x}_1)} \\ \vdots \\ \sqrt{\mathbf{q}_k} - \sqrt{\mathbf{p}(\mathbf{x}_k)} \end{bmatrix}, \quad \mathbf{M} = \begin{bmatrix} \mathbf{M}_1 & 0 & 0 & 0 \\ 0 & \mathbf{M}_2 & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & \mathbf{M}_k \end{bmatrix},$$

$$\mathbf{G} = \begin{bmatrix} \frac{\partial \Omega}{\partial \mathbf{x}_1} & \frac{\partial \Omega}{\partial \mathbf{x}_2} & \cdots & \frac{\partial \Omega}{\partial \mathbf{x}_k} \end{bmatrix}, \qquad l = -\Omega(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_k)$$

- We have

$$\mathtt{rank}\left(\begin{bmatrix} \mathbf{M} \\ \sqrt{\gamma}\mathbf{G} \end{bmatrix} \right) \geq \mathtt{rank}(\mathbf{M})$$

- it enhances the motion observability

An Example

- special case: $\Delta \mathbf{x}_1 = \Delta \mathbf{x}_2 = \ldots = \Delta \mathbf{x}_k$, and γ is chosen as the optimal Lagrangian multiplier, then

$$\mathbf{G} = \begin{bmatrix} \mathbf{I} & -\mathbf{I} & & & \\ & \mathbf{I} & -\mathbf{I} & & \\ & & \ddots & \ddots & \\ & & & \mathbf{I} & -\mathbf{I} \end{bmatrix}, \quad \text{and} \quad l = 0.$$

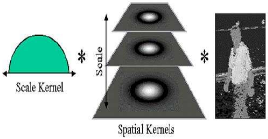
- we have $\texttt{rank}(\mathbf{G}) = (k-1) \times \texttt{dim}(\mathbf{x}_1)$.
- E.g., supposing $k = 10$ and $\texttt{dim}(\mathbf{x}_1) = 2$, this implies that the motion resides in a 2-D manifold in $\mathbb{R}^{20}$.
- Thus, as long as $\texttt{rank}(\mathbf{M}) \geq \texttt{dim}(\mathbf{x}_1)$, all the motion parameters are observable, or can be uniquely determined.
- It is be easily satisfied if
  - any of the $\mathbf{x}_i$ is observable through its kernel,
  - there are a number of $\texttt{dim}(\mathbf{x}_1)$ motion parameters that are observable through multiple kernels.

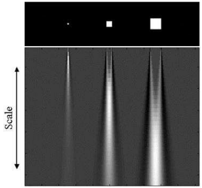
Solution and Collaboration

- The solution
$$\underline{\Delta\mathbf{x}} = (\mathbf{M}^T\mathbf{M} + \gamma\mathbf{G}^T\mathbf{G})^{-1}(\mathbf{M}^T\mathbf{y} + \gamma\mathbf{G}^T l).$$
- A more efficient solution
$$\underline{\Delta\mathbf{x}} = (\mathbf{I} - \mathbf{D})(\mathbf{M}^T\mathbf{M})^{-1}(\mathbf{M}^T\mathbf{y} + \gamma\mathbf{G}^T l),$$
where
$$\mathbf{D} = \gamma(\mathbf{M}^T\mathbf{M})^{-1}\mathbf{G}^T(\gamma\mathbf{G}(\mathbf{M}^T\mathbf{M})^{-1}\mathbf{G}^T + \mathbf{I})^{-1}\mathbf{G}$$
- Notice that
$$\underline{\Delta\mathbf{x}}_u = (\mathbf{M}^T\mathbf{M})^{-1}\mathbf{M}^T\mathbf{y} = \mathbf{M}^\dagger\mathbf{y},$$
is the solution to the independent kernels, and
$$\underline{\Delta\mathbf{x}} = (\mathbf{I} - \mathbf{D})\underline{\Delta\mathbf{x}}_u + \mathbf{z}(\mathbf{x})$$
- The collaboration through a fixed-point recursion
$$\underline{\Delta\mathbf{x}}^{k+1} \longleftarrow (\mathbf{I} - \mathbf{D}^k)[\mathbf{M}(\underline{\Delta\mathbf{x}}^k)]^\dagger\mathbf{y}^k + \mathbf{z}^k,$$

MKL for scale[10]

- Determining the scale of the target is an important issue
- It is related to the scale of the kernel
- Basic idea: using mean-shift in the spatial-scale space $(\mathbf{x}, \sigma)$
- Algorithm: alternating a spatial mean-shift and a scale one
  1. initial states $(\mathbf{x}_0, \sigma_0)$;
  2. fix σ_0, perform a 2-D spatial mean-shift to obtain $\mathbf{x}'$;
  3. fix $\mathbf{x}'$, perform a 1-D scale mean-shift to obtain σ';
  4. repeat 2 and 3 until convergence.

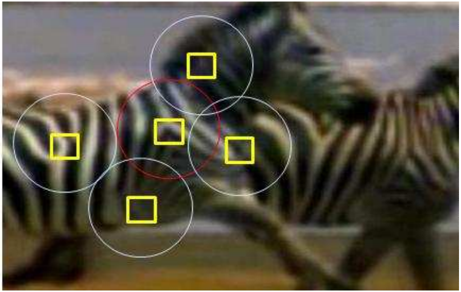




[10] Robert Collins, "Mean-shift Blob Tracking through Scale Space", CVPR'03

Outline

Distraction and Matching Ambiguity



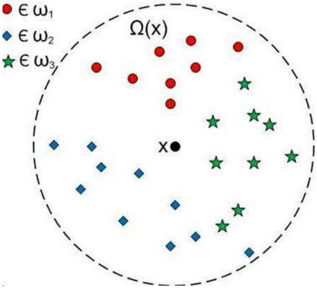
- Spatial context can reduce matching ambiguity

- Questions:
  - Modeling context for motion analysis?
  - Methods resilient to local variations?

Spatial Context (for object recognition)

- Structure-stiff (e.g., template and filters)
- Structure-flexible
  - random fields
  - deformable templates
  - shape context, AutoContext
- Structure-free
  - bag-of-words or bag-of-features

Modeling Spatial Context

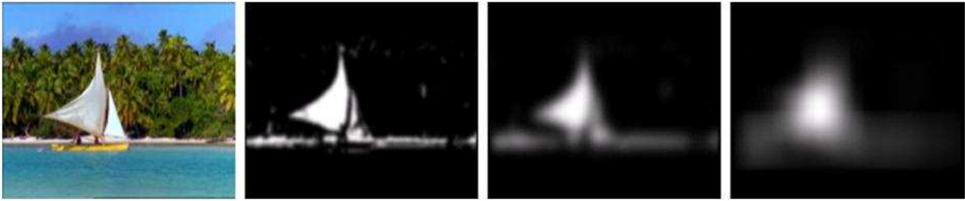


- Location $\mathbf{x}$ is associated with features $\mathbf{f}(\mathbf{x})$
- feature class $\{\omega_1, \ldots, \omega_N\}$
- *individual context*:

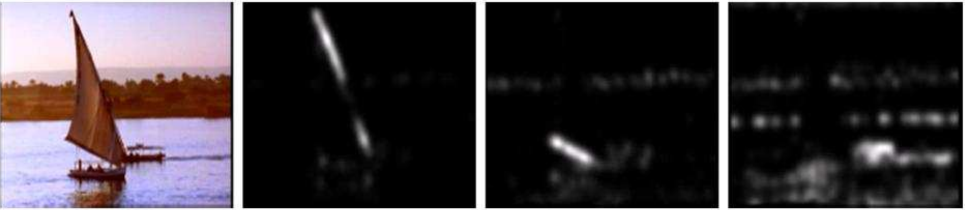
$$\mathcal{C}_i = \{\mathbf{y} | \mathbf{f}(\mathbf{y}) \in \omega_i, \mathbf{y} \in \Omega(\mathbf{x})\},$$

- *total context*: $\mathcal{C} = \bigcup_{i=1}^{N} \mathcal{C}_i$.
- context representation: $p(\omega_i|\mathbf{x}) \propto p(\mathbf{x}|\omega_i)p(\omega_i)$

Contextual Maps



"white" color contexts at 3 different scales



edge contexts at 3 different orientations

Brightness Constancy $\longrightarrow$ Context Constancy[11]

- Context constancy

$$p(\omega_i|\mathbf{x} + \Delta\mathbf{x}, t + \Delta t, \mathcal{C}) = p(\omega_i|\mathbf{x}, t, \mathcal{C})$$

- The motion $\Delta\mathbf{x}$ shall not change the context

- More flexible than constant brightness
  - insensitive to lighting
  - insensitive to local deformation

- Let's impose a small motion assumption ...

[11] Ying Wu and Jialue Fan, "Contextual Flow", CVPR'09

A Differential Form

$$\underbrace{\nabla_x^T p(\omega_i|\mathbf{x}, t)}_{\texttt{contextual gradient}} \Delta\mathbf{x} + \underbrace{\nabla_t p(\omega_i|\mathbf{x}, t)}_{\texttt{contextual frame difference}} \Delta t = 0$$

- Contextual frame difference is approximated by

$$p(\omega_i|\mathbf{x}, t + \Delta t) - p(\omega_i|\mathbf{x}, t)$$

- Contextual gradient (details follow)

$$\begin{aligned}
\nabla_x p(\omega_i|\mathbf{x}) &= \nabla_x \left\{ p(\omega_i) \frac{p(\mathbf{x}|\omega_i)}{p(\mathbf{x})} \right\} \\
&= \frac{1}{c} p(\omega_i|\mathbf{x}) \Big[\mu_i(\mathbf{x}) - \mu_0(\mathbf{x}) \Big]
\end{aligned}$$

Context Gradient

- Conditional Shift

$$\mu_i(\mathbf{x}) \triangleq E\{(\mathbf{y}-\mathbf{x})|\mathbf{y}\in\omega_i\} = \frac{1}{Z_i(\mathbf{x})}\int_\Omega (\mathbf{y}-\mathbf{x})p(\mathbf{y}|\omega_i)d\mathbf{y}$$

- After simple manipulation

$$\mu_i(\mathbf{x}) = c\frac{\nabla_x p(\mathbf{x}|\omega_i)}{p(\mathbf{x}|\omega_i)}$$

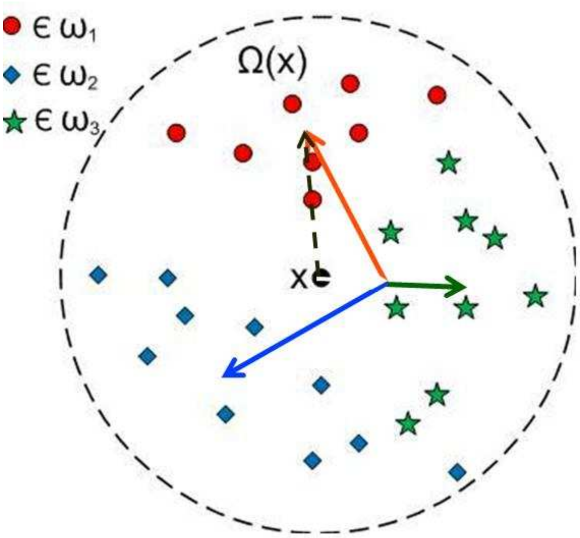
- Total shift

$$\mu_0(\mathbf{x}) \triangleq E\{(\mathbf{y}-\mathbf{x})|\mathbf{y}\in\Omega\} = c\frac{\nabla_x p(\mathbf{x})}{p(\mathbf{x})}$$

- so we have

$$\nabla_x p(\omega_i|\mathbf{x}) = \frac{1}{c}p(\omega_i|\mathbf{x})\Big[\mu_i(\mathbf{x}) - \mu_0(\mathbf{x})\Big]$$

Illustration: Contextual Gradient



Context Flow Constraint

- It is easy to see:

$$\underbrace{\left[\mu_i(\mathbf{x}) - \mu_0(\mathbf{x})\right]^T}_{\bar{\mu}_i(\mathbf{x})} \Delta\mathbf{x} + \underbrace{c\left[\frac{p(\omega_i|\mathbf{x}, t+1)}{p(\omega_i|\mathbf{x}, t)} - 1\right]}_{-b_i} = 0$$

- $\bar{\mu}_i(\mathbf{x})$ is the *centered shift*

- b_i is the change of context ratio

- Contextual flow constraint

$$\boxed{\bar{\mu}_i(\mathbf{x})^T \Delta\mathbf{x} - b_i = 0}$$

Local Contextual System

- Each context gives a constrain weighted by $W_i(\mathbf{x}) = p(\omega_i|\mathbf{x}, t)$, and

$$\mathbf{W}(\mathbf{x}) \triangleq diag[W_1(\mathbf{x}), \dots, W_N(\mathbf{x})]$$

- Denote by

$$\mathbf{U}_r(\mathbf{x}) \triangleq \left[\bar{\mu}_1(\mathbf{x}), \dots, \bar{\mu}_N(\mathbf{x})\right]^T, \quad \mathbf{b}_r(\mathbf{x}) \triangleq [b_1, b_2 \dots, b_N]^T,$$

$$\mathbf{U}(\mathbf{x}) = \mathbf{W}(\mathbf{x})\mathbf{U}_r(\mathbf{x}), \qquad \mathbf{b}(\mathbf{x}) = \mathbf{W}(\mathbf{x})\mathbf{b}_r(\mathbf{x})$$

- we have a linear contextual system

$$\mathbf{U}(\mathbf{x})\Delta\mathbf{x} = \mathbf{b}(\mathbf{x}), \text{ or simply } \quad \mathbf{U}\Delta\mathbf{x} = \mathbf{b}$$

Extended Lucas-Kanade Method

- If $\mathbf{U}$ is rank deficient, we have an aperture problem as well
- Considering the nearby locations $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_m\}$
- each of which is associated with a contextual system

$$\mathbf{U}_i(\mathbf{x}_i)\Delta\mathbf{x}_i = \mathbf{b}(\mathbf{x}_i), \text{ or simply } \mathbf{U}_i\Delta\mathbf{x}_i = \mathbf{b}_i$$

  where $\Delta\mathbf{x}_i$ is the motion for location $\mathbf{x}_i$.
- If they share the same motion, i.e., $\Delta\mathbf{x}_i = \Delta\mathbf{x}$, then
- Extended Lucas-Kanade method

$$\begin{bmatrix} \mathbf{U}_1 \\ \ldots \\ \mathbf{U}_m \end{bmatrix} \Delta\mathbf{x} \triangleq \mathbf{U}_c\Delta\mathbf{x} = \begin{bmatrix} \mathbf{b}_1 \\ \ldots \\ \mathbf{b}_m \end{bmatrix}$$