Differential Motion Analysis (I)

Ying Wu

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

yingwu@northwestern.edu

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

Optical Flow

- Brightness Constancy and Optical Flow Constraint
- Lucas-Kanade’s Method
- Horn-Schunck’s Method
- Parametric Flow
- Robust Flow and Multi-frame Flow

Beyond Basic Optical Flow

- Considering Lighting Variations
- Considering Appearance Variations
- Considering Spatial-Appearance Variations
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

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

- 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 $\Omega$.
▶ Define a weight function $W(m)$, $m \in \Omega$, for the pixels.
▶ Weighted LS formulation

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

▶ WLS solution:

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

$$
A = \begin{bmatrix}
\frac{\partial I_1}{\partial x_1} & \frac{\partial I_1}{\partial y_1} \\
\vdots & \vdots \\
\frac{\partial I_N}{\partial x_N} & \frac{\partial I_N}{\partial y_N}
\end{bmatrix}, \quad W = \text{diag}(W(m_1), \ldots, W(m_N))
$$

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

▶ the intersection of all the flow constraint lines corresponding to the pixels in $\Omega$. 

Horn-Schunck’s Method

- Assume: flow varies smoothly \(\leftarrow\) global regularization
- The measure of departure from smoothness can be written by:

\[
es = \int \int (|\nabla v_x|^2 + |\nabla v_y|^2) dxdy
\]

\[
= \int \int (\frac{\partial v_x}{\partial x})^2 + (\frac{\partial v_x}{\partial y})^2 + (\frac{\partial v_y}{\partial x})^2 + (\frac{\partial v_y}{\partial y})^2 dxdy
\]

- The error of optical flow is:

\[
e_c = \int \int (\nabla I \cdot v_m + \frac{\partial I}{\partial t})^2 dxdy
\]

- Objective function:

\[
e = e_c + \lambda e_s
\]

\[
= \int \int (\nabla I \cdot v_m + \frac{\partial I}{\partial t})^2 + \lambda (|\nabla v_x|^2 + |\nabla v_y|^2) dxdy
\]
Horn-Schunck’s Method

- **Fixed-point iteration:**

  \[
  v_{x}^{k+1} = \bar{v}_{x}^{k} - \frac{\left( \frac{\partial I}{\partial x} \right) \bar{v}_{x}^{k} + \left( \frac{\partial I}{\partial y} \right) \bar{v}_{y}^{k} + \frac{\partial I}{\partial t}}{\lambda + \left( \frac{\partial I}{\partial x} \right)^2 + \left( \frac{\partial I}{\partial y} \right)^2} \left( \frac{\partial I}{\partial x} \right)
  \]

  \[
  v_{y}^{k+1} = \bar{v}_{y}^{k} - \frac{\left( \frac{\partial I}{\partial x} \right) \bar{v}_{x}^{k} + \left( \frac{\partial I}{\partial y} \right) \bar{v}_{y}^{k} + \frac{\partial I}{\partial t}}{\lambda + \left( \frac{\partial I}{\partial x} \right)^2 + \left( \frac{\partial I}{\partial y} \right)^2} \left( \frac{\partial I}{\partial y} \right)
  \]

- **Concisely, it is:**

  \[
  v^{k+1} = \bar{v}^{k} - \alpha(\nabla l)
  \]

- 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

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

- 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**

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

- or

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

- denote by \( \nabla l_\theta = \nabla I^T \nabla v(\Theta) \),

\[
\min_{\theta} \sum_{\Omega} \left[ \nabla l_\theta^T \Theta + l_t \right]^2
\]

- Easy to figure out the LS solution.
Exercises

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

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

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

► **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

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

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

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

- **Applying influence function to flow estimation**

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

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

---

Multi-frame Optical Flow\textsuperscript{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:

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

\textsuperscript{2}Michal Irani, “Multi-Frame Optical Flow Estimation Using Subspace Constraints”, ICCV'99
Multi-frame Optical Flow

And

\[
\mathbf{V} = \begin{bmatrix}
V_{y11} & V_{y21} & \cdots & V_{yN1} \\
\vdots & \vdots & \ddots & \vdots \\
V_{y1F} & V_{y2F} & \cdots & V_{yNF}
\end{bmatrix} = \begin{bmatrix}
a_1^1 & a_1^3 & a_1^4 \\
\vdots & \vdots & \vdots \\
a_6^F & a_3^F & a_4^F
\end{bmatrix} \begin{bmatrix}
1 & 1 & \cdots & 1 \\
x_1 & x_2 & \cdots & x_N \\
y_1 & y_2 & \cdots & y_N
\end{bmatrix}
\]

It is clear that

\[
\text{rank} \begin{bmatrix}
\mathbf{U} \\
\mathbf{V}
\end{bmatrix} \leq 3
\]

\[
\text{rank} \begin{bmatrix}
\mathbf{U} & \mathbf{V}
\end{bmatrix} \leq 6
\]
Outline

Optical Flow
- Brightness Constancy and Optical Flow Constraint
- Lucas-Kanade’s Method
- Horn-Schunck’s Method
- Parametric Flow
- Robust Flow and Multi-frame Flow

Beyond Basic Optical Flow
- Considering Lighting Variations
- Considering Appearance Variations
- Considering Spatial-Appearance Variations
Considering lighting models

- Brightness constancy assumption is too restrictive
- Are there constraints for lighting?
- For a pure Lambertian 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

$$I(x, y, t) + \mathbf{B} \Lambda,$$

where $\Lambda = \begin{bmatrix} \lambda_1 \\ \vdots \\ \lambda_m \end{bmatrix}$
Considering lighting models\(^3\)

\[ E(\Theta, \Lambda) = \sum_{\Omega} ||I(x + v_x(\Theta)dt, y + v_y(\Theta)dt, t + dt) - I(x, y, t) - B\Lambda||^2 \]

or

\[ E(\Theta, \Lambda) = \sum_{\Omega} ||\nabla^T v(\Theta) + I_t - B\Lambda||^2 \]

or

\[ E(\Theta, \Lambda) = \sum_{\Omega} ||\nabla^T_\theta \Theta + I_t - B\Lambda||^2 \]

\[ \Rightarrow \text{denote by } \nabla l^T_\theta = M, \text{ we have} \]

\[ \begin{bmatrix} M & -B \end{bmatrix} \begin{bmatrix} \Theta \\ \Lambda \end{bmatrix} = -I_t \]

\(^3\)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

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

\[
= -\begin{bmatrix} M^T M & -M^T B \\ -B^T M & B^T B \end{bmatrix}^{-1} \begin{bmatrix} M^T \\ -B^T \end{bmatrix} I_t
\]

easy to see

\[
\Theta = -\begin{bmatrix} M^T (I - BB^T) M \end{bmatrix}^{-1} M^T (I + BB^T) I_t
\]
Considering appearance variations

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

$$\min_v S \left( l(u + v_x dt, v + v_y dt, t + dt) : \Lambda \right),$$

where $\Lambda$ are parameters of the classifier

- E.g., using an SVM classifier

$$\sum_{j=1}^n y_j \alpha_j k(l, x_j) + b$$

- Let’s maximize the SVM matching score

$$\max_{u,v} \sum_{j=1}^n y_j \alpha_j k(l + ul_x + vl_y, x_j)$$
Considering appearance variations\(^4\)

- Let use a 2nd order polynomial kernel \(k(x, x_j) = (x^T x_j)^2\)
- so, we have

\[
E(u, v) = \sum_{j=1}^{n} y_j \alpha_j \left[(l + ul_x + vl_y)^T x_j\right]^2
\]

\[
\frac{\partial E}{\partial u} = \sum y_j \alpha_j l_x^T x_j (l + ul_x + vl_y)^T x_j = 0
\]

\[
\frac{\partial E}{\partial v} = \sum y_j \alpha_j l_y^T x_j (l + ul_x + vl_y)^T x_j = 0
\]

- the solution is:

\[
\begin{bmatrix}
\sum \alpha_j y_j (x_j^T l_x)^2 & \sum \alpha_j y_j (x_j^T l_x)(x_j^T l_y) \\
\sum \alpha_j y_j (x_j^T l_x)(x_j^T l_y) & \sum \alpha_j y_j (x_j^T l_y)^2
\end{bmatrix}
\begin{bmatrix}
u
\end{bmatrix}
= \begin{bmatrix}
- \sum \alpha_j y_j (x_j^T l_x)(x_j^T l_x) \\
- \sum \alpha_j y_j (x_j^T l_y)(x_j^T l_y)
\end{bmatrix}
\]

\(^4\) Shai Avidan, “Subset Selection for Efficient SVM Tracking”, CVPR’03
Spatial-appearance model (SAM)\(^5\)

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

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

- For a pixel, the likelihood is a mixture

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

- Let’s use an affine motion here

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

- then, we have

\[
p(T(y; a_t)|\Theta) = p(T(x; a_t), c(T(x; a_t))|\Theta)
\]

\[
= \sum_{k=1}^{K} p_k g(x; \mu_{ks}, \Sigma_{ks})g(c(T(x; a_t)); \mu_{kc}, \Sigma_{kc}) \triangleq \sum_{k=1}^{K} q(k, y_i; a_t)
\]

\(^5\)Ting Yu and Ying Wu, “Differential Tracking based on Spatial- Appearance Model (SAM)”, CVPR’06
Spatial-appearance model (SAM)

- For an image region

\[
E(a_t; \Theta) = \sum_{x_i \in \Omega} \log p(T(y_i; a_t)|\Theta) = \sum_{x_i \in \Omega} \log \left( \sum_{k=1}^{K} q(k, y_i; a_t) \right)
\]

- 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.