Motion and Human Actions

Josef Sivic
INRIA, WILLOW, ENS/INRIA/CNRS UMR 8548
Laboratoire d’Informatique, Ecole Normale Supérieure, Paris

With slides from Ivan Laptev
Announcements

• Check all assignments have been received: http://www.di.ens.fr/willow/teaching/recvis15/

• Assignment 3 is due today Nov. 17

• FP proposals are due next week Nov. 24
Final projects – evaluation and due dates

• **Project proposal** (due on Nov 24th). You will submit a 1-page project proposal indicating (i) your chosen topic, (ii) the plan of work, i.e. what are you going to implement, what data you are going to use, what experiments you are going to do, (iii) if working in a group, who are the members of the group and how you plan to share the work. *The project proposal will represent 10% of the final project grade.*

• **Project presentation** (on Jan 7/8, 2016). You will present your work in the class. *The project presentation will represent 20% of the final project grade.*

• **Project report** (due on Jan 11 2016). You will write a short report (<3 pages) summarizing your work. *The report will represent 70% of the final project grade.*
Class overview

Motivation
- Historic review
- Modern applications

Appearance-based methods
- Motion history images
- Active shape models
- Tracking and motion priors

Motion-based methods
- Generic and parametric Optical Flow
- Motion templates

Space-time methods
- Local space-time features
- Action classification and detection
- Weakly-supervised action learning
Motivation I: Artistic Representation

Early studies were motivated by human representations in Arts

Da Vinci: “it is indispensable for a painter, to become totally familiar with the anatomy of nerves, bones, muscles, and sinews, such that he understands for their various motions and stresses, which sinews or which muscle causes a particular motion”

“I ask for the weight [pressure] of this man for every segment of motion when climbing those stairs, and for the weight he places on b and on c. Note the vertical line below the center of mass of this man.”

Leonardo da Vinci (1452–1519): A man going upstairs, or up a ladder.
Giovanni Alfonso Borelli (1608–1679)

The emergence of *biomechanics*

Borelli applied to biology the analytical and geometrical methods, developed by Galileo Galilei

He was the first to understand that bones serve as levers and muscles function according to mathematical principles

His physiological studies included muscle analysis and a mathematical discussion of movements, such as running or jumping

Motivation II: Biomechanics
Motivation III: Motion perception

Etienne-Jules Marey: (1830–1904) made Chronophotographic experiments influential for the emerging field of cinematography.

Eadweard Muybridge (1830–1904) invented a machine for displaying the recorded series of images. He pioneered motion pictures and applied his technique to movement studies.
Motivation III: Motion perception

Gunnar Johansson [1973] pioneered studies on the use of image sequences for a programmed human motion analysis

“Moving Light Displays” (LED) enable identification of familiar people and the gender and inspired many works in computer vision.

Human actions: Historic overview

- 15th century studies of anatomy
- 17th century emergence of biomechanics
- 19th century emergence of cinematography
- 1973 studies of human motion perception

Modern computer vision
Modern applications: Motion capture and animation

Avatar (2009)
Modern applications: Motion capture and animation

Leonardo da Vinci (1452–1519)

Avatar (2009)
Modern applications: Video editing

*Space-Time Video Completion*

Y. Wexler, E. Shechtman and M. Irani, *CVPR* 2004
Modern applications: Video editing

Space-Time Video Completion
Y. Wexler, E. Shechtman and M. Irani, CVPR 2004
Modern applications: Video editing

Recognizing Action at a Distance
Alexei A. Efros, Alexander C. Berg, Greg Mori, Jitendra Malik, ICCV 2003
Modern applications: Video editing

Recognizing Action at a Distance
Alexei A. Efros, Alexander C. Berg, Greg Mori, Jitendra Malik, ICCV 2003
Applications: Unusual Activity Detection

e.g. for surveillance

*Detecting Irregularities in Images and in Video*  
Boiman & Irani, *ICCV* 2005
Why automatic video understanding?

- Huge amount of video is available and growing

- ~30M surveillance cameras in US  
  => ~700K video hours/day
Why video analysis?

Applications:

- First appearance of N. Sarkozy on TV
- Sociology research: Influence of character smoking in movies
- Education: How do I make a pizza?
- Where is my cat?
- Predicting crowd behavior
- Counting people
- Motion capture and animation
Why human actions?

How many person-pixels are in the video?

Movies

TV

YouTube
Why human actions?

How many person-pixels are in the video?

- Movies: 35%
- TV: 34%
- YouTube: 40%
How many person pixels in our daily life?

• Wearable camera data: Microsoft SenseCam dataset
How many person pixels in our daily life?

- Wearable camera data: Microsoft SenseCam dataset

$\sim 4\%$
### Challenges

- **Large variations in appearance:** occlusions, non-rigid motion, viewpoint changes, clothing…

  Action *Hugging*:

- **Manual collection of training samples is prohibitive:** many action classes, rare occurrence

- **Action vocabulary is not well-defined:** What is action granularity?

  Action *Open*:
What is action granularity?

Source: http://www.youtube.com/watch?v=eYdUZdan5i8

Do we want to learn person-throws-cat-into-trash-bin classifier?
How action recognition is related to computer vision?
We can recognize cars and roads, What’s next?
Class overview

Motivation
- Historic review
- Modern applications

Appearance-based methods
- Motion history images
- Active shape models
- Tracking and motion priors

Motion-based methods
- Generic and parametric Optical Flow
- Motion templates

Space-time methods
- Local space-time features
- Action classification and detection
- Weakly-supervised action learning
How to recognize actions?
Action understanding: Key components

**Image measurements**
- Foreground segmentation
- Image gradients
- Optical flow
- Local space-time features

**Prior knowledge**
- Deformable contour models
- 2D/3D body models
- Motion priors
  - Background models
  - Action labels

**Association**
- Learning associations from strong / weak supervision
- Automatic inference
Class overview

**Motivation**
- Historic review
- Modern applications

**Appearance-based methods**
- Motion history images
- Active shape models
- Tracking and motion priors

**Motion-based methods**
- Generic and parametric Optical Flow
- Motion templates

**Space-time methods**
- Local space-time features
- Action classification and detection
- Weakly-supervised action learning
Class overview

Motivation
- Historic review
- Modern applications

Appearance-based methods
- Motion history images
- Active shape models
- Tracking and motion priors

Motion-based methods
- Generic and parametric Optical Flow
- Motion templates

Space-time methods
- Local space-time features
- Action classification and detection
- Weakly-supervised action learning
Action understanding: Key components

**Image measurements**
- Foreground segmentation
- Image gradients
- Optical flow
- Local space-time features

**Prior knowledge**
- Deformable contour models
- 2D/3D body models
- Motion priors
- Background models
- Action labels

**Association**
- Learning associations from strong / weak supervision
- Automatic inference
Foreground segmentation

Image differencing: a simple way to measure motion/change

Better Background / Foreground separation methods exist:

- Modeling of color variation at each pixel with Gaussian Mixture
- Dominant motion compensation for sequences with moving camera
- Motion layer separation for scenes with non-static backgrounds

What are the potential problems?
Temporal Templates

\[ D(x, y, t) \quad t = 1, \ldots, T \]

Idea: summarize motion in the video in a 
*Motion History Image (MHI)*:

\[ H_\tau(x, y, t) = \begin{cases} 
\tau & \text{if } D(x, y, t) = 1 \\
\max (0, H_\tau(x, y, t - 1) - 1) & \text{otherwise}
\end{cases} \]

Descriptor: Hu moments of different orders

\[ m_{pq} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x^p y^q \rho(x, y) \, dx \, dy \]

[A.F. Bobick and J.W. Davis, PAMI 2001]
Aerobics dataset

Nearest Neighbor classifier: 66% accuracy
Temporal Templates: Summary

Pros:
+ Simple and fast
+ Works in controlled settings

Cons:
- Prone to errors of background subtraction
- Does not capture *interior* motion and shape

Not all shapes are valid
Restrict the space of admissible silhouettes

Variations in light, shadows, clothing…
What is the background here?

Silhouette tells little about actions
Active Shape Models of Cootes et al.

Point Distribution Model

- Represent the shape of samples by a set of corresponding points or *landmarks*

\[
x = (x_1, \ldots, x_n, y_1, \ldots, y_n)^T
\]

- Assume each shape can be represented by the linear combination of basis shapes

\[
\Phi = (\phi_1 | \phi_2 | \ldots | \phi_t)
\]

such that

\[
x \approx \bar{x} + \Phi b
\]

for mean shape

\[
\bar{x} = \frac{1}{s} \sum_{i=1}^{s} x_i
\]

and some parameters \(b\)
Active Shape Models of Cootes et al.

- Basis shapes can be found as the main modes of variation of in the training data.

**2D Example:** (each point can be thought as a shape in 2N-Dim space)

Principle Component Analysis (PCA):

Covariance matrix \( S = \frac{1}{s-1} \sum_{i=1}^{s} (x_i - \bar{x})(x_i - \bar{x})^T \)

Eigenvectors \( \Phi = (\phi_1 | \phi_2 | \ldots | \phi_t) \) eigenvalues \( \lambda_1, \ldots, \lambda_t \)
Active Shape Models of Cootes et al.

- Back-project from shape-space \( \mathbf{b} \) to image space \( \mathbf{x} = \bar{\mathbf{x}} + \Phi \mathbf{b} \)

Three main modes of lips-shape variation:

\[
\mathbf{b} = (\mu \lambda_1, 0, 0, ...)^\top \\
\mathbf{b} = (0, \mu \lambda_2, 0, 0, ...)^\top \\
\mathbf{b} = (0, 0, \mu \lambda_3, 0, 0, ...)^\top
\]

\( \mu = -3, 1.5, 0, 1.5, 3 \)

Distribution of eigenvalues: \( \lambda_1, \lambda_2, \lambda_3, ... \)

A small fraction of basis shapes (eigenvectors) accounts for the most of shape variation (=> landmarks are redundant)
Active Shape Models of Cootes et al.

- $\Phi$ is orthonormal basis, therefore $\Phi^{-1} = \Phi^\top$
  
  Given estimate of $\mathbf{x}$ we can recover shape parameters $\mathbf{b}$
  
  $$\mathbf{b} = \Phi^\top (\mathbf{x} - \bar{\mathbf{x}})$$

- Projection onto the shape-space serves as a regularization

  $$\mathbf{x} \Rightarrow \mathbf{b} = \Phi^\top (\mathbf{x} - \bar{\mathbf{x}}) \Rightarrow \mathbf{x}_{\text{reg}} = \bar{\mathbf{x}} + \Phi \mathbf{b}$$
How to use Active Shape Models for shape estimation?

• Given initial guess of model points $\mathbf{x}$ estimate new positions $\mathbf{x}'$ using local image search, e.g. locate the closest edge point

• Re-estimate shape parameters

$$\mathbf{b}' = \Phi^\top (\mathbf{x}' - \bar{x})$$
To handle translation, scale and rotation, it is useful to normalize $x$ prior to shape estimation:

$$x = T(\bar{x} + \Phi b)$$

using similarity transformation

$$T(x_{\text{norm}}) = \begin{pmatrix} a & c \\ -c & a \end{pmatrix} x + \begin{pmatrix} t_x \\ t_y \end{pmatrix}$$

A simple way to estimate $T$ is to assign $(t_x, t_y)$ and $a$ to the mean position and the standard deviation of points in $x$ respectively and set $c = 0$. For more sophisticated normalization techniques see:

http://www.isbe.man.ac.uk/~bim/Models/app_model.ps.gz

Note: model parameters $\bar{x}$, $\Phi$ have to be computed using normalized image point coordinates $x_{\text{norm}} = T^{-1}(x)$
Active Shape Models of Cootes et al.

- Iterative ASM alignment algorithm
  1. Initialize with the reasonable guess of $T$ and $b = 0^T$
  2. Estimate $x'$ from image measurements
  3. Re-estimate $T, b$
  4. Unless $T, b$ converged, repeat from step 2

Example: face alignment

Illustration of face shape space

Active Shape Models: Their Training and Application
Active Shape Model tracking

Aim: to track ASM of time-varying shapes, e.g. human silhouettes

- Impose time-continuity constraint on model parameters. For example, for shape parameters $b_i$:
  \[ b_i^{(k)} = b_i^{(k-1)} + w_i^{k-1} \]
  \[ w_i \sim \mathcal{N}(0, \mu \lambda_i) \quad \text{Gaussian noise} \]

For similarity transformation $T$

\[ a^{(k)} = a^{(k-1)} + w_a^{k-1}, \quad w_a = \mathcal{N}(0, \sigma_a) \]
\[ t_x^{(k)} = t_x^{(k-1)} + v_x^{(k-1)} + w_x^{k-1}, \quad w_x|y = \mathcal{N}(0, \sigma_x|y) \]

- More complex dynamical models possible

- Update model parameters at each time frame using e.g. Kalman filter
Learning flexible models from image sequences
A. Baumberg and D. Hogg, ECCV 1994
Person Tracking

Learning flexible models from image sequences
A. Baumberg and D. Hogg, ECCV 1994
Active Shape Models: Summary

Pros:
- Shape prior helps overcoming segmentation errors
- Fast optimization
- Can handle interior/exterior dynamics

Cons:
- Optimization gets trapped in local minima
- Re-initialization is problematic

Possible improvements:
- Learn and use motion priors, possibly specific to different actions
Motion priors

- Accurate motion models can be used both to:
  - Help accurate tracking
  - Recognize actions

- Goal: formulate motion models for different types of actions and use such models for action recognition

Example:

Drawing with 3 action modes
- line drawing
- scribbling
- idle

[M. Isard and A. Blake, ICCV 1998]
Incorporating motion priors

**Image measurements**
- Foreground segmentation
- Image gradient
- Optical Flow

**Data Association**
- Particle filters

**Prior knowledge**
- Learning motion models for different actions

**Learning motion models for different actions**
Bayesian Tracking

General framework: recognition by synthesis;
    generative models;
    finding best explanation of the data

Notation:

$Z_i$ image data at time $i$
$X_i$ model parameters at time $i$ (e.g. shape and its dynamics)
$p(X_i)$ prior density for $X_i$
$p(Z_i|X_i)$ likelihood of data for the given model configuration

We search posterior defined by the Bayes’ rule

$$p(X|Z) \propto p(Z|X)p(X)$$

For tracking the Markov assumption gives the prior

$$p(X_i|X_{i-1})$$

Temporal update rule:

$$p(X_i|Z_i) \propto p(Z_i|X_i)p(X_i|X_{i-1})$$
Kalman Filtering

If all probability densities are uni-modal, specifically Gaussians, the posterior can be evaluated in closed form.

\[ p(X_{i-1}) \]

\[ p(x) \]

\[ p(x) \]

\[ p(x) \]

\[ p(x) \]

\[ p(x) \]

\[ p(x) \]

\[ p(Z_i | X_i)p(X_i | X_{i-1}) \]

\[ p(X_i | X_{i-1}) \]
Particle Filtering

In reality probability densities are almost always *multi-modal*

\[
p(X_{i-1})
\]

\[
p(z) \quad p(x) \quad p(z) \quad p(x)
\]

\[
p(X_i|Z_i) \propto p(Z_i|X_i)p(X_i|X_{i-1})
\]

\[
p(X_i|X_{i-1})
\]
Particle Filtering

Represent a multi-modal distribution by a set of N particles: $s^{(n)}$ each with weight $\pi^{(n)}$, chosen to be proportional to $p(z|x)$. 

---

*Figure 3.* Factored sampling: a set of points $s^{(n)}$, the centres of the blobs in the figure, is sampled randomly from a prior density $p(x)$. Each sample is assigned a weight $\pi_i$ (depicted by blob area) in proportion to the value of the observation density $p(z|x = s^{(n)})$. The weighted point-set then serves as a representation of the posterior density $p(x|z)$, suitable for sampling. The one-dimensional case illustrated here extends naturally to the practical case that the density is defined over several position and shape variables.
Particle Filtering

In reality probability densities are almost always *multi-modal*

Approximate distributions with weighted particles
Particle Filtering

Tracking examples:

\( X \) describes leave shape

\( X \) describes head shape

CONDENSATION - conditional density propagation for visual tracking
A. Blake and M. Isard *IJCV* 1998
Learning dynamic prior

- Dynamic model: 2\textsuperscript{nd} order Auto-Regressive Process

State:

\[
\mathcal{X}_k = \begin{pmatrix} X_{k-1} \\ X_k \end{pmatrix}
\]

Update rule:

\[
\mathcal{X}_k - \bar{X} = A(\mathcal{X}_{k-1} - \bar{X}) + B w_k
\]

Model parameters:

\[
A = \begin{pmatrix} 0 & I \\ A_2 & A_1 \end{pmatrix}, \quad \bar{X} = \begin{pmatrix} \bar{X} \\ \bar{X} \end{pmatrix} \quad \text{and} \quad B = \begin{pmatrix} 0 \\ B_0 \end{pmatrix}
\]

Learning scheme:

Shape Space

Hand-built dynamics

Training sequence slow, clutter-free

Fast test sequences

Faster training sequence

Infer dynamical model

Iterate
Learning dynamic prior

Learning point sequence

Random simulation of the learned dynamical model

Statistical models of visual shape and motion
A. Blake, B. Bascle, M. Isard and J. MacCormick, Phil.Trans.R.Soc. 1998
Learning dynamic prior

Random simulation of the learned gait dynamics
Dynamics with discrete states

Introduce “mixed” state \( \mathcal{X}_k^+ = \begin{pmatrix} \mathcal{X}_k \\ y_k \end{pmatrix} \)

Continuous state space (as before)

Discrete variable identifying dynamical model \( y_k = 1, 2, \ldots, n \)

Transition probability matrix

\[
P(y_k = j \mid y_{k-1} = i) = T_{i,j},
\]

or more generally

\[
P(y_k = j \mid y_{k-1} = i, \mathcal{X}_{k-1}) = T_{i,j}(\mathcal{X}_{k-1})
\]

Incorporation of the mixed-state model into a particle filter is straightforward, simply use \( \mathcal{X}_k^+ \) instead of \( \mathcal{X}_k \) and the corresponding update rules
Dynamics with discrete states

Example: Drawing

Transition probability matrix

\[
T = \begin{pmatrix}
0.9800 & 0.0015 & 0.0185 \\
0.0850 & 0.9000 & 0.0150 \\
0.0050 & 0.0150 & 0.9800
\end{pmatrix}
\]

Result: simultaneously improved tracking and gesture recognition

A mixed-state Condensation tracker with automatic model-switching
M. Isard and A. Blake, ICCV 1998
Dynamics with discrete states

Similar method illustrated on gesture recognition in the context of a visual black-board interface

[M.J. Black and A.D. Jepson, ECCV 1998]
Motion priors & Tracking: Summary

Pros:

+ more accurate tracking using specific motion models
+ Simultaneous tracking and motion recognition with discrete state dynamical models

Cons:

- Local minima is still an issue
- Re-initialization is still an issue
## Class overview

### Motivation
- Historic review
- Modern applications

<table>
<thead>
<tr>
<th>Appearance-based methods</th>
</tr>
</thead>
<tbody>
<tr>
<td>Motion history images</td>
</tr>
<tr>
<td>Active shape models</td>
</tr>
<tr>
<td>Tracking and motion priors</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Motion-based methods</th>
</tr>
</thead>
<tbody>
<tr>
<td>Generic and parametric Optical Flow</td>
</tr>
<tr>
<td>Motion templates</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Space-time methods</th>
</tr>
</thead>
<tbody>
<tr>
<td>Local space-time features</td>
</tr>
<tr>
<td>Action classification and detection</td>
</tr>
<tr>
<td>Weakly-supervised action learning</td>
</tr>
</tbody>
</table>
Class overview

Motivation
- Historic review
- Modern applications

Appearance-based methods
- Motion history images
- Active shape models
- Tracking and motion priors

Motion-based methods
- Generic and parametric Optical Flow
- Motion templates

Space-time methods
- Local space-time features
- Action classification and detection
- Weakly-supervised action learning
Shape and Appearance vs. Motion

• Shape and appearance in images depends on many factors: clothing, illumination contrast, image resolution, etc…

• Motion field (in theory) is invariant to shape and can be used directly to describe human actions

[Efros et al. 2003]
Motion estimation: Optical Flow

• Classic problem of computer vision [Gibson 1955]

• Goal: estimate motion field

  How? We only have access to image pixels
  Estimate pixel-wise correspondence between frames = Optical Flow

• Brightness constancy assumption: corresponding pixels preserve their intensity (color)

  ▶ Useful assumption in many cases

  ▶ Breaks at occlusions and illumination changes

  ▶ Physical and visual motion may be different
Problem definition: optical flow

How to estimate pixel motion from image $H$ to image $I$?
  - Solve pixel correspondence problem
    - given a pixel in $H$, look for nearby pixels of the same color in $I$

Key assumptions
  - **color constancy**: a point in $H$ looks the same in $I$
    - For grayscale images, this is **brightness constancy**
  - **small motion**: points do not move very far

This is called the **optical flow** problem
Optical flow constraints (grayscale images)

Let's look at these constraints more closely

- brightness constancy: Q: what's the equation?
  \[ H(x, y) = I(x + u, y + v) \]

- small motion: \((u\text{ and }v \text{ are less than 1 pixel})\)
  - suppose we take the Taylor series expansion of \(I\):
    \[ I(x + u, y + v) = I(x, y) + \frac{\partial I}{\partial x} u + \frac{\partial I}{\partial y} v + \text{higher order terms} \]
    \[ \approx I(x, y) + \frac{\partial I}{\partial x} u + \frac{\partial I}{\partial y} v \]
Optical flow equation

Combining these two equations

\[ 0 = I(x + u, y + v) - H(x, y) \]

\[ \approx I(x, y) + I_x u + I_y v - H(x, y) \]

\[ \approx (I(x, y) - H(x, y)) + I_x u + I_y v \]

\[ \approx I_t + I_x u + I_y v \]

\[ \approx I_t + \nabla I \cdot [u \ v] \]

In the limit as \( u \) and \( v \) go to zero, this becomes exact

\[ 0 = I_t + \nabla I \cdot \left[ \frac{\partial x}{\partial t} \frac{\partial y}{\partial t} \right] \]

shorthand: \( I_x = \frac{\partial I}{\partial x} \)

Slide: F. Durand, A. Efros, B. Freeman, S. Seitz, R. Szeliski
Optical flow equation

\[ 0 = I_t + \nabla I \cdot [u \ v] \]

Q: how many unknowns and equations per pixel?

2 unknowns, one equation

Intuitively, what does this constraint mean?

- The component of the flow in the gradient direction is determined
- The component of the flow parallel to an edge is unknown

This explains the Barber Pole illusion

http://en.wikipedia.org/wiki/Barber's_pole

Aperture problem
Aperture problem
Solving the aperture problem

How to get more equations for a pixel?

• Basic idea: impose additional constraints
  – most common is to assume that the flow field is smooth locally
  – one method: pretend the pixel’s neighbors have the same \((u,v)\)
    » If we use a 5x5 window, that gives us 25 equations per pixel!

\[
0 = I_t(p_i) + \nabla I(p_i) \cdot [u \ v]
\]

\[
\begin{bmatrix}
  I_x(p_1) & I_y(p_1) \\
  I_x(p_2) & I_y(p_2) \\
  \vdots & \vdots \\
  I_x(p_{25}) & I_y(p_{25})
\end{bmatrix}
\begin{bmatrix}
  u \\
  v
\end{bmatrix}
= -
\begin{bmatrix}
  I_t(p_1) \\
  I_t(p_2) \\
  \vdots \\
  I_t(p_{25})
\end{bmatrix}
\]

25x2

25x1
Lukas-Kanade flow

Prob: we have more equations than unknowns

\[ A \begin{pmatrix} d \\ 25 \times 2 \end{pmatrix} = b \begin{pmatrix} 25 \times 1 \end{pmatrix} \rightarrow \text{minimize } \|Ad - b\|^2 \]

Solution: solve least squares problem

- minimum least squares solution given by solution (in \(d\)) of:

\[
\begin{pmatrix} A^T A \end{pmatrix} \begin{pmatrix} d \end{pmatrix} = A^T b
\]

\[
\begin{pmatrix} \sum I_x I_x & \sum I_x I_y \\ \sum I_x I_y & \sum I_y I_y \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix} = - \begin{pmatrix} \sum I_x I_t \\ \sum I_y I_t \end{pmatrix}
\]

- The summations are over all pixels in the \(K \times K\) window
- This technique was first proposed by Lukas & Kanade (1981)
Conditions for solvability

• Optimal (u, v) satisfies Lucas-Kanade equation

\[
\begin{bmatrix}
\sum I_x I_x & \sum I_x I_y \\
\sum I_x I_y & \sum I_y I_y
\end{bmatrix}
\begin{bmatrix}
u \\
v
\end{bmatrix} =
- \begin{bmatrix}
\sum I_x I_t \\
\sum I_y I_t
\end{bmatrix}
\]

When is This Solvable?

• \(A^T A\) should be invertible

• \(A^T A\) should not be too small due to noise
  – eigenvalues \(\lambda_1\) and \(\lambda_2\) of \(A^T A\) should not be too small

• \(A^T A\) should be well-conditioned
  – \(\lambda_1 / \lambda_2\) should not be too large (\(\lambda_1 = \) larger eigenvalue)

\(A^T A\) is solvable when there is no aperture problem

\[
A^T A = \begin{bmatrix}
\sum I_x I_x & \sum I_x I_y \\
\sum I_x I_y & \sum I_y I_y
\end{bmatrix} = \sum \begin{bmatrix}
I_x \\
I_y
\end{bmatrix} [I_x \ I_y] = \sum \nabla I (\nabla I)^T
\]
Local Patch Analysis
\[ \sum \nabla I (\nabla I)^T \]
- large gradients, all the same
- large \( \lambda_1 \), small \( \lambda_2 \)
Low texture region

\[ \sum \nabla I(\nabla I)^T \]

- gradients have small magnitude
- small \( \lambda_1 \), small \( \lambda_2 \)
High textured region

\[ \sum \nabla I (\nabla I)^T \]
- gradients are different, large magnitudes
  - large \( \lambda_1 \), large \( \lambda_2 \)
Generic Optical Flow

- Brightness Constancy Constraint Equation (BCCE)

\[(\nabla I)^\top v + I_t = 0\]

\[v = (v_x, v_y)^\top\] Optical flow
\[\nabla I = (I_x, I_y)^\top\] Image gradient

One equation, two unknowns => cannot be solved directly

Integrate several measurements in the local neighborhood and obtain a Least Squares Solution [Lucas & Kanade 1981]

\[< \nabla I (\nabla I)^\top > v = -< \nabla I I_t >\]

\[
\begin{pmatrix}
< I_x^2 > & < I_x I_y > \\
< I_x I_y > & < I_y^2 >
\end{pmatrix}
\begin{pmatrix}
v_x \\
v_y
\end{pmatrix}
= -
\begin{pmatrix}
< I_x I_t > \\
< I_y I_t >
\end{pmatrix}
\]

\[< \cdot >\] Denotes integration over a spatial (or spatio-temporal) neighborhood of a point

Second-moment matrix, the same one used to compute Harris interest points!
Generic Optical Flow

- The solution of \( \langle \nabla I (\nabla I)^\top \rangle v = -\langle \nabla I I_t \rangle \) assumes
  1. Brightness change constraint holds in \( \langle \cdot \rangle \)
  2. Sufficient variation of image gradient in \( \langle \cdot \rangle \)
  3. Approximately constant motion in \( \langle \cdot \rangle \)

Motion estimation becomes inaccurate if any of assumptions 1-3 is violated.

- Solutions:
  1. Insufficient gradient variation known as aperture problem
     - Increase integration neighborhood
  2. Non-constant motion in \( \langle \cdot \rangle \)
     - Use more sophisticated motion model
Parameterized Optical Flow

- Constant velocity model: \( \mathbf{v} = \begin{pmatrix} v_x \\ v_y \end{pmatrix} \)

- Upgrade to affine motion model: \( \mathbf{v} = \begin{pmatrix} a_1 & a_2 \\ a_3 & a_4 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} v_x \\ v_y \end{pmatrix} \)

Now motion depends on the position \((x, y)\) inside the neighborhood

Examples of Affine motion models for different parameters:

- Can be formulated as Least Squares approach to estimate \( \mathbf{v} \) as before!
Another extension of the constant motion model is to compute PCA basis flow fields from training examples

1. Compute standard Optical Flow for many examples
2. Put velocity components into one vector
   \[ w = (v_x^1, v_y^1, v_x^2, v_y^2, ..., v_x^n, v_y^n) \]
3. Do PCA on \( w \) and obtain most informative PCA flow basis vectors

---

**Learning Parameterized Models of Image Motion**

M.J. Black, Y. Yacoob, A.D. Jepson and D.J. Fleet, CVPR 1997
Parameterized Optical Flow

• Use PCA flow bases to *regularize* solution of motion estimation
• Motion estimation for test samples can be computed *without* explicit computation of optical flow

Solution formulation e.g. in terms of Least Squares

Direct flow recovery:

*Learning Parameterized Models of Image Motion*
M.J. Black, Y. Yacoob, A.D. Jepson and D.J. Fleet, CVPR 1997
Parameterized Optical Flow

- Estimated coefficients of PCA flow bases can be used as action descriptors

Training Smile

Test Smile

Learning Parameterized Models of Image Motion
M.J. Black, Y. Yacoob, A.D. Jepson and D.J. Fleet, CVPR 1997
Parameterized Optical Flow

- Estimated coefficients of PCA flow bases can be used as action descriptors.

Optical flow seems to be an interesting descriptor for motion/action recognition.
Spatial Motion Descriptor

Image frame

Optical flow $F_{x,y}$

$F_x, F_y$

$F_x^-, F_x^+, F_y^-, F_y^+$

blurred $F_x^-, F_x^+, F_y^-, F_y^+$
Spatio-Temporal Motion Descriptor

Temporal extent $E$

Sequence A

Sequence B

Temporal extent $E$

Frame-to-frame similarity matrix

Motion-to-motion similarity matrix

Slide credit: A. Efros
Football Actions: matching

Input Sequence

Matched Frames

input  matched

Slide credit: A. Efros
Football Actions: classification

10 actions; 4500 total frames; 13-frame motion descriptor
16 Actions; 24800 total frames; 51-frame motion descriptor. Men used to classify women and vice versa.
Classifying Tennis Actions

6 actions; 4600 frames; 7-frame motion descriptor
Woman player used as training, man as testing.

[Alexei A. Efros, Alexander C. Berg, Greg Mori, Jitendra Malik, ICCV 2003]
What we have seen so far?

**Temporal templates:**
- Simple, fast
- Sensitive to segmentation errors

**Active shape models:**
- Shape regularization
- Sensitive to initialization and tracking failures

**Tracking with motion priors:**
- Improved tracking and simultaneous action recognition
- Sensitive to initialization and tracking failures

**Motion-based recognition:**
- Generic descriptors; less depends on appearance
- Sensitive to localization/tracking errors
Class overview

Motivation
- Historic review
- Modern applications

Appearance-based methods
- Motion history images
- Active shape models
- Tracking and motion priors

Motion-based methods
- Generic and parametric Optical Flow
- Motion templates

Space-time methods
- Local space-time features
- Action classification and detection
- Weakly-supervised action learning
Class overview

Motivation
- Historic review
- Modern applications

Appearance-based methods
- Motion history images
- Active shape models
- Tracking and motion priors

Motion-based methods
- Generic and parametric Optical Flow
- Motion templates

Space-time methods
- Local space-time features
- Action classification and detection
- Weakly-supervised action learning