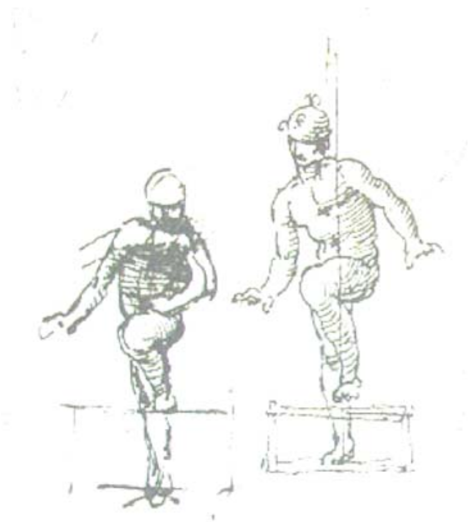


Ecole Normale Supérieure Lyon
Winter School Vision and Machine Learning
January 24-28, 2011

Motion and Human Actions

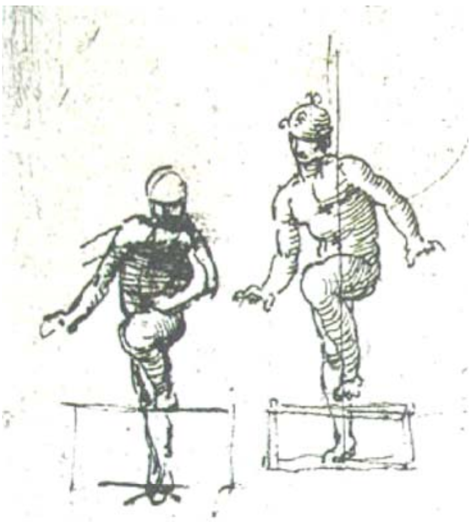
Ivan Laptev

ivan.laptev@inria.fr

INRIA, WILLOW, ENS/INRIA/CNRS UMR 8548
Laboratoire d'Informatique, Ecole Normale Supérieure, Paris

Includes slides from: Ondra Chum, Alyosha Efros, Mark Everingham, Pedro Felzenszwalb, Rob Fergus, Kristen Grauman, Bastian Leibe, Ivan Laptev, Fei-Fei Li, Marcin Marszalek, Pietro Perona, Deva Ramanan, Bernt Schiele, Jamie Shotton, Andrea Vedaldi and Andrew Zisserman

Class overview



Motivation

Historic review
Modern applications

Human Pose Estimation

Pictorial structures
Learning models from image data
Recent advances

Appearance-based methods

Motion history images
Active shape models
Tracking and motion priors

Motion-based methods

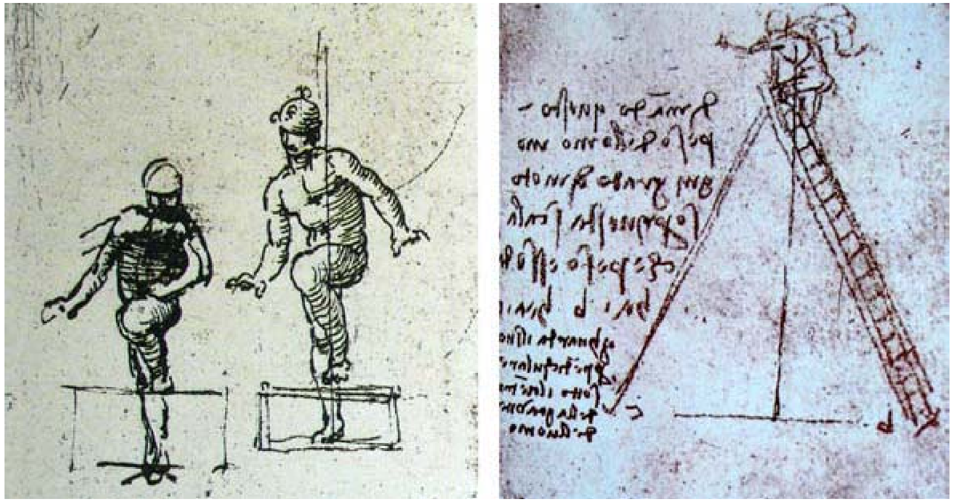
Generic and parametric Optical Flow
Motion templates

Motivation I: Artistic Representation

Early studies were motivated by human representations in Arts

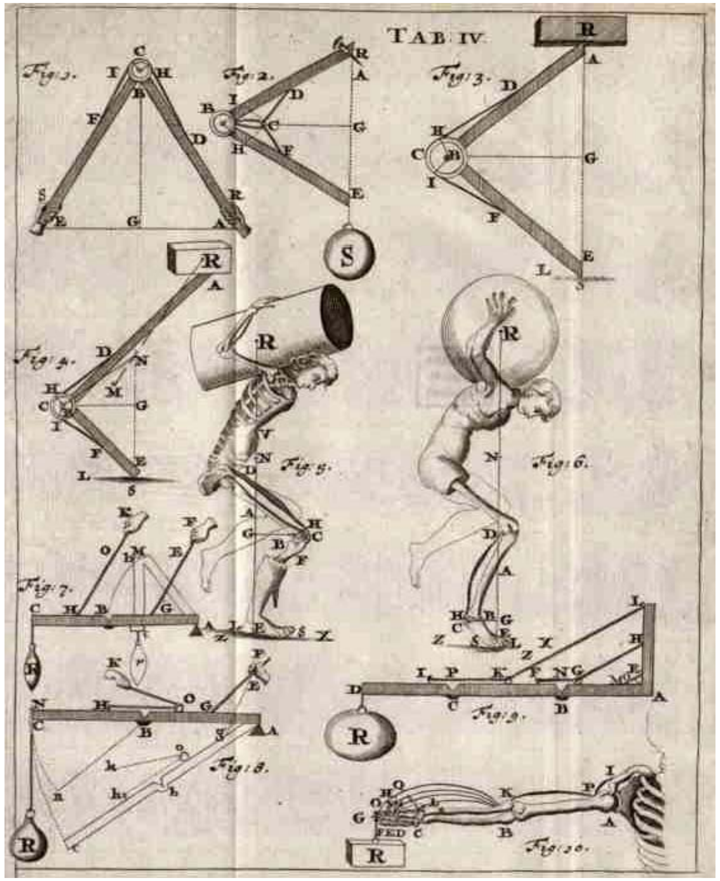
Da Vinci: “it is indispensable for a painter, to become totally familiar with the anatomy of nerves, bones, muscles, and sinews, such that he understands for their various motions and stresses, which sinews or which muscle causes a particular motion”

“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.

Motivation II: Biomechanics

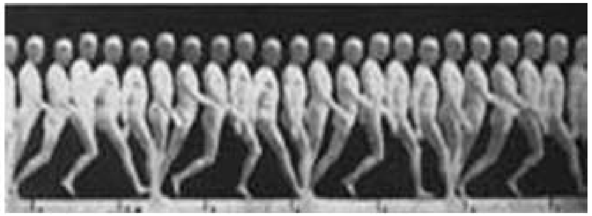


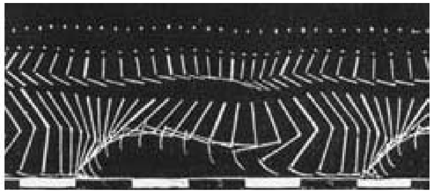
Giovanni Alfonso Borelli (1608–1679)

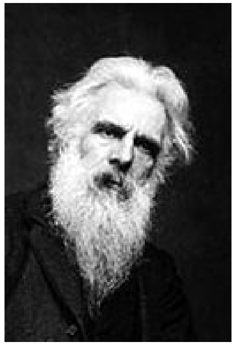
- The emergence of *biomechanics*
- Borelli applied to biology the analytical and geometrical methods, developed by Galileo Galilei
- He was the first to understand that bones serve as levers and muscles function according to mathematical principles
- His physiological studies included muscle analysis and a mathematical discussion of movements, such as running or jumping

Motivation III: Motion perception

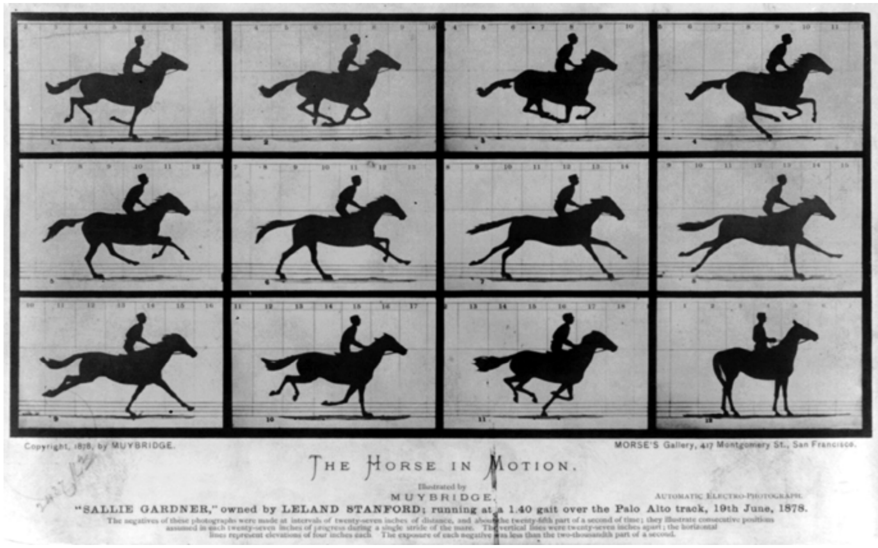
Etienne-Jules Marey: (1830–1904) made Chronophotographic experiments influential for the emerging field of *cinematography*





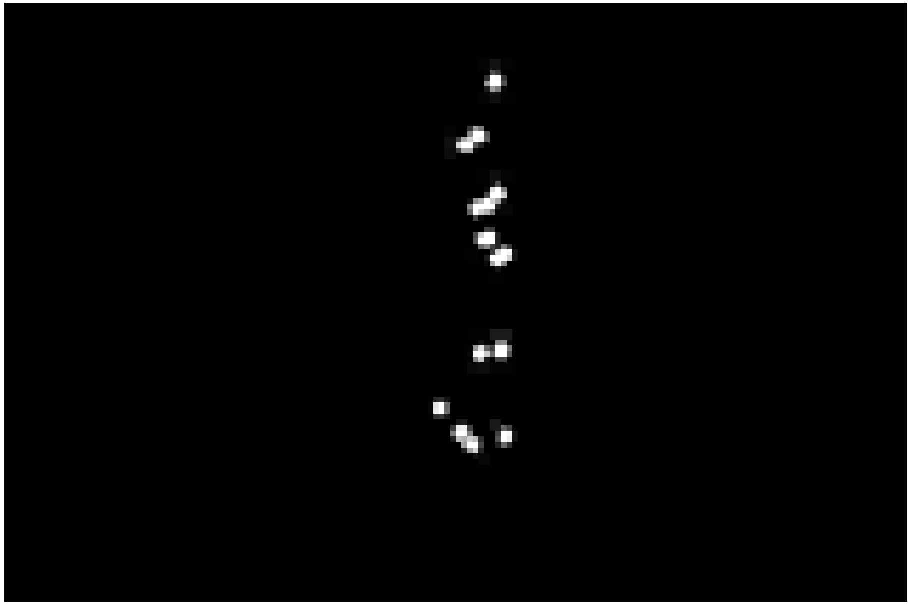


Eadweard Muybridge (1830–1904) invented a machine for displaying the recorded series of images. He pioneered motion pictures and applied his technique to movement studies



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.

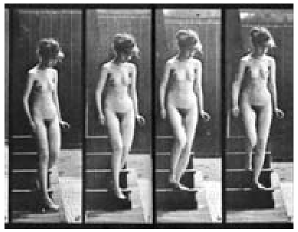


Gunnar Johansson, **Perception and Psychophysics,** 1973

Human actions: Historic overview

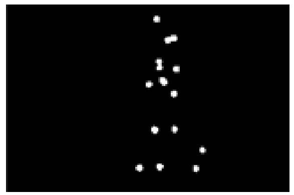
15th century
studies of
anatomy

17th century
emergence of
biomechanics



19th century
emergence of
cinematography

1973
studies of human
motion perception

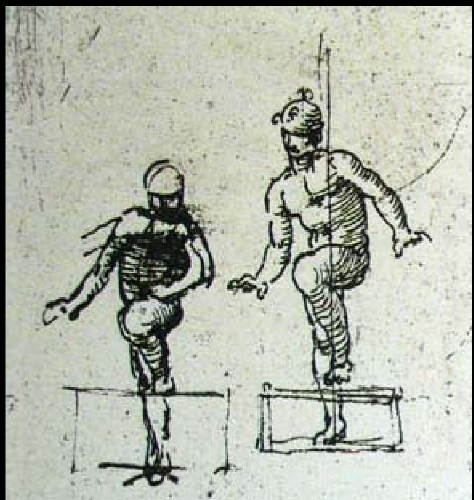


Modern computer vision

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

TV-channels recorded since 60's

>34K hours of video upload every day

~30M surveillance cameras in US
=> ~700K video hours/day

Why automatic video understanding?

- Video indexing and search is useful in TV production, entertainment, education, social studies, security,…

TV & Web: e.g. *“Fight in a parlament”*

Home videos: e.g. *“My daughter climbing”*

Sociology research:

Manually analyzed smoking actions in 900 movies

Surveillance: e.g. *“Woman throws cat into wheelie bin”*
260K views in 7 days

- … how much is it about people?

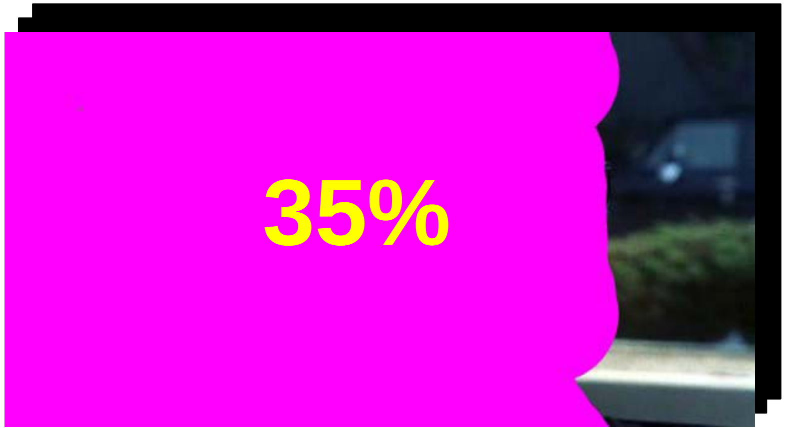
How many person-pixels are there?

Movies

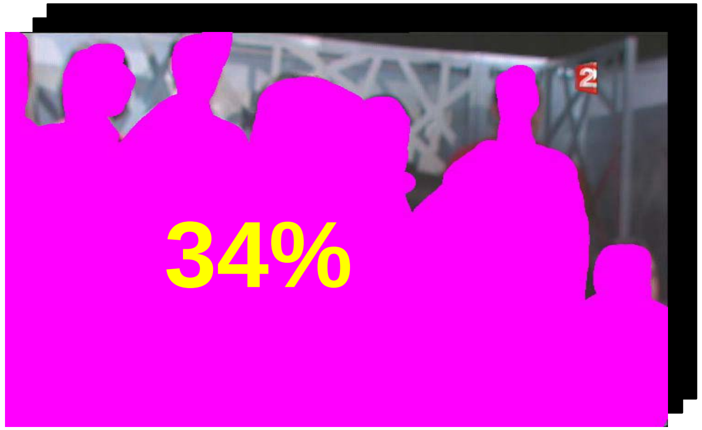
TV

YouTube

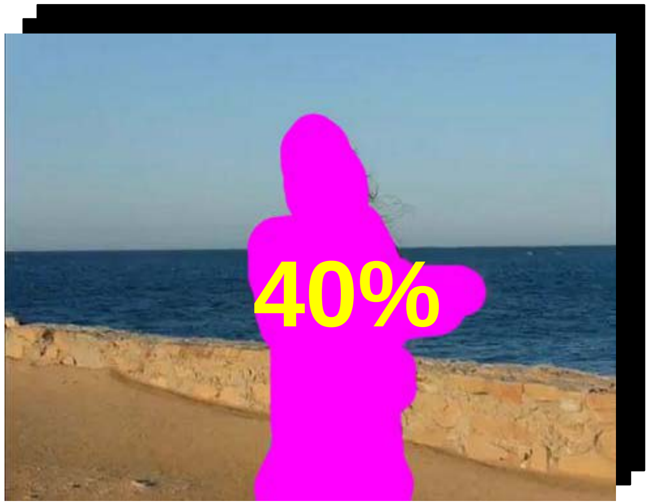
How many person-pixels are there?



Movies

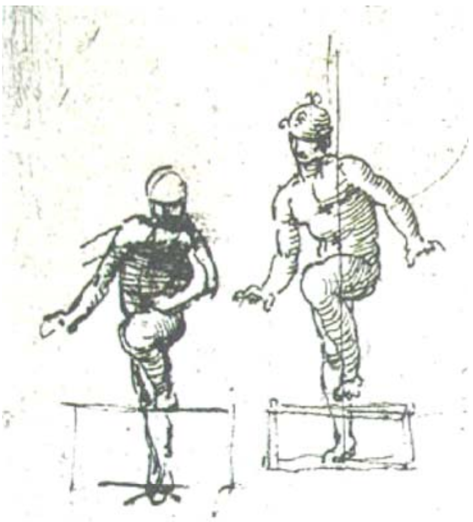


TV



YouTube

Class overview



Motivation

Historic review
Modern applications

Human Pose Estimation

Pictorial structures
Learning models from image data
Recent advances

Appearance-based methods

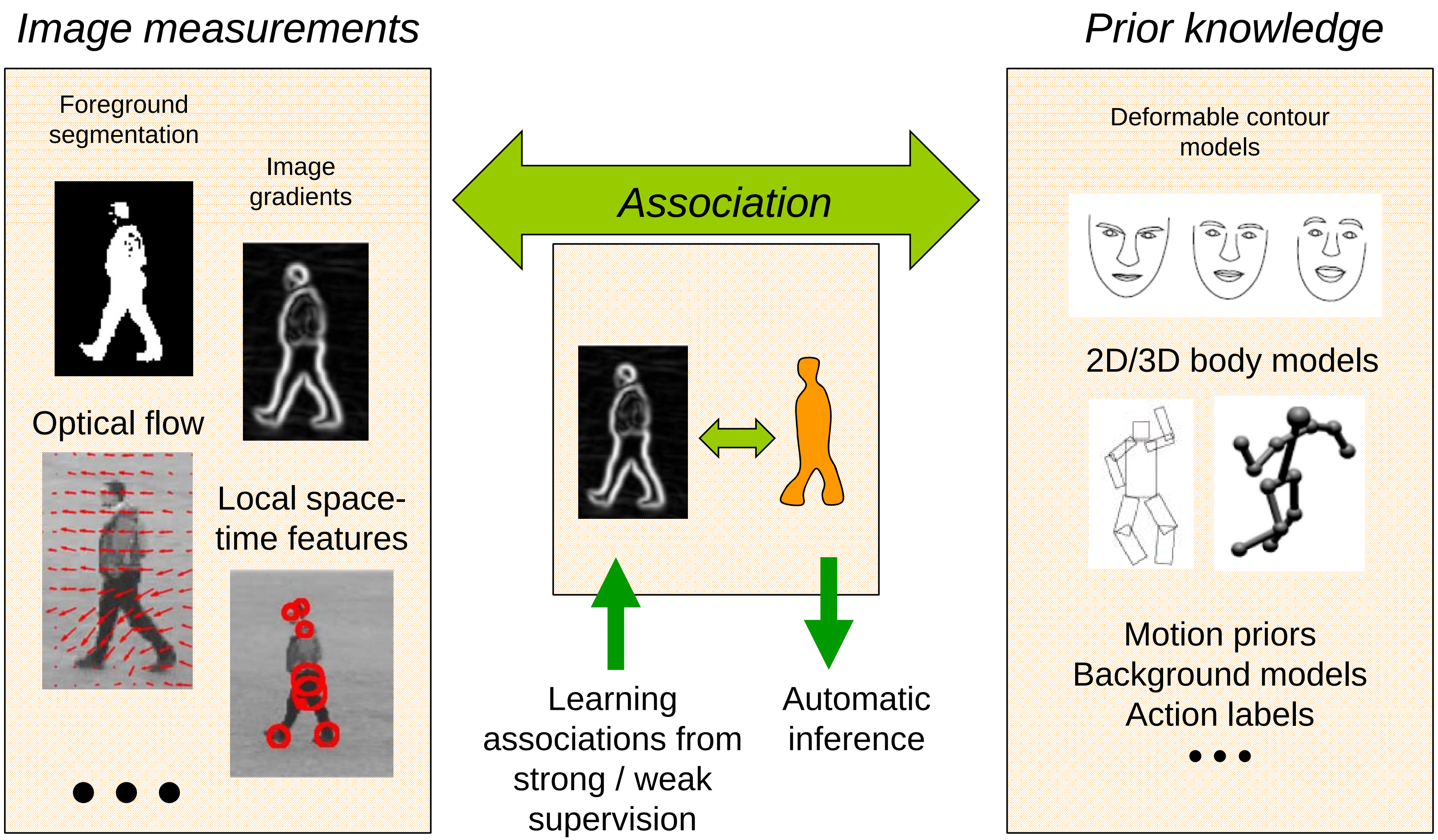
Motion history images
Active shape models
Motion priors

Motion-based methods

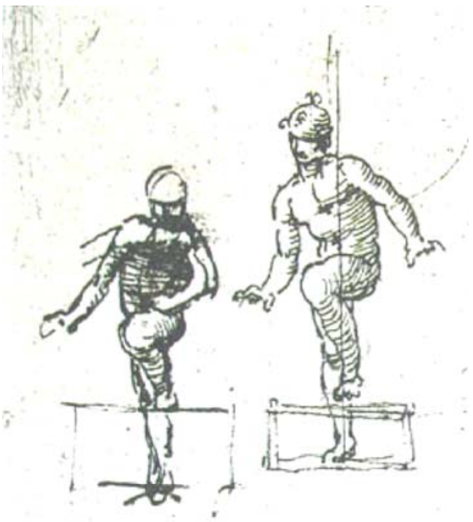
Generic and parametric Optical Flow
Motion templates

How to recognize actions?

Action understanding: Key components



Class overview



Motivation

Historic review
Modern applications

Human Pose Estimation

Pictorial structures
Learning models from image data
Recent advances

Appearance-based methods

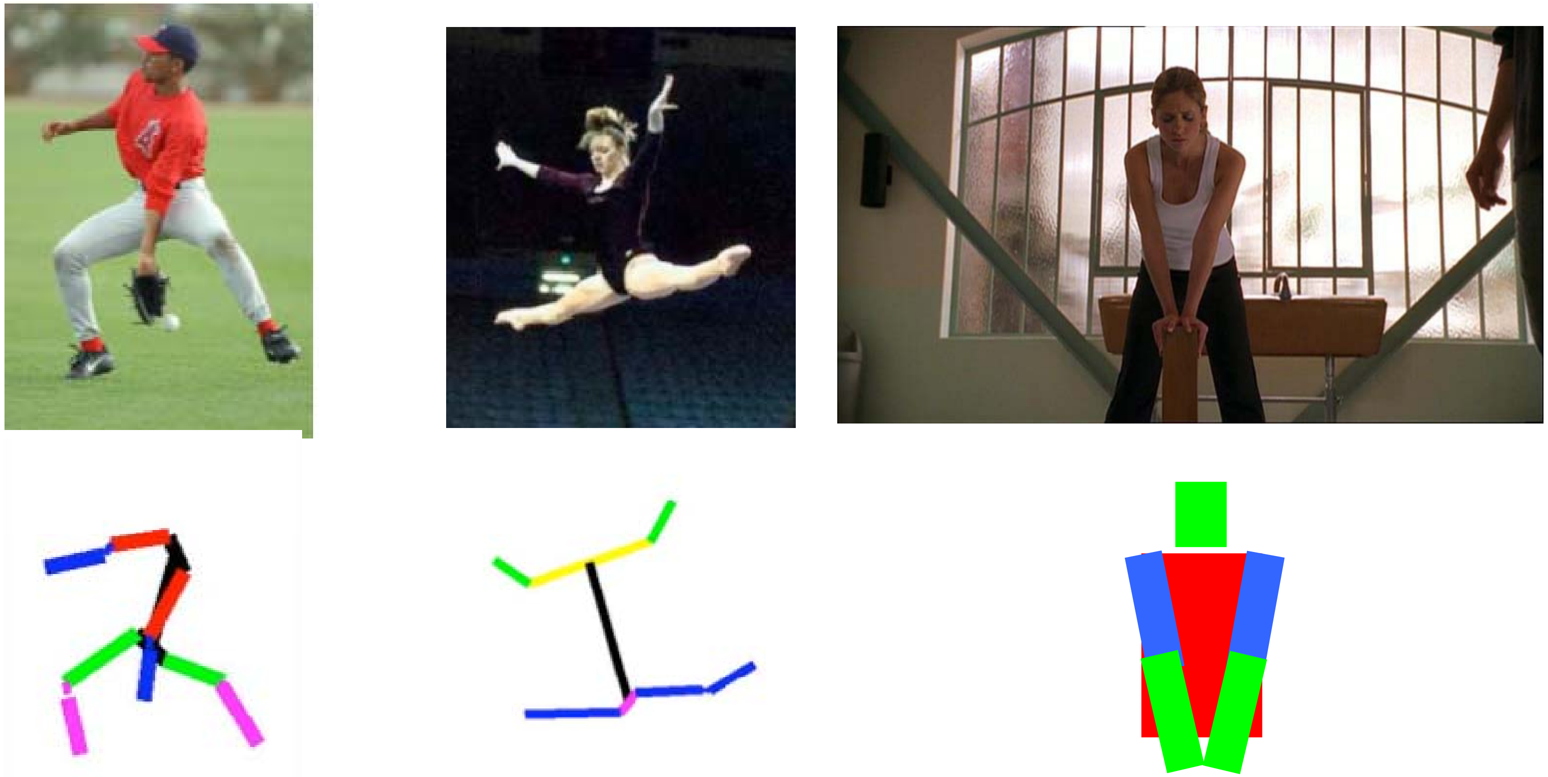
Motion history images
Active shape models
Motion priors

Motion-based methods

Generic and parametric Optical Flow
Motion templates

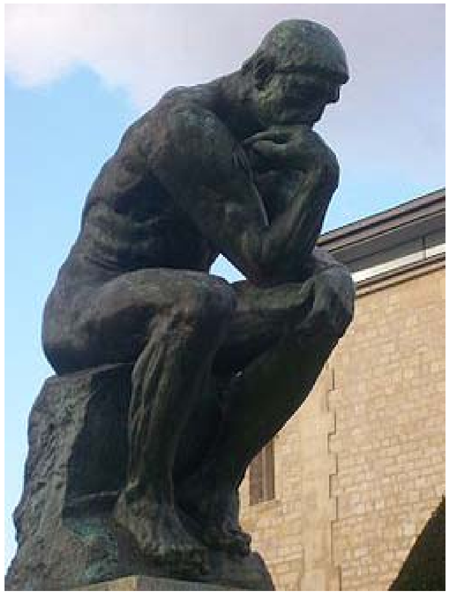
Objective and motivation

Determine human body pose (layout)



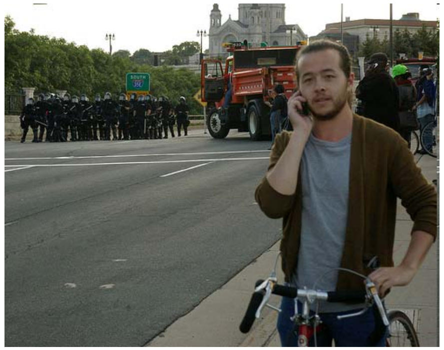
Why? To recognize poses, gestures, actions

Activities characterized by a pose



Activities characterized by a pose

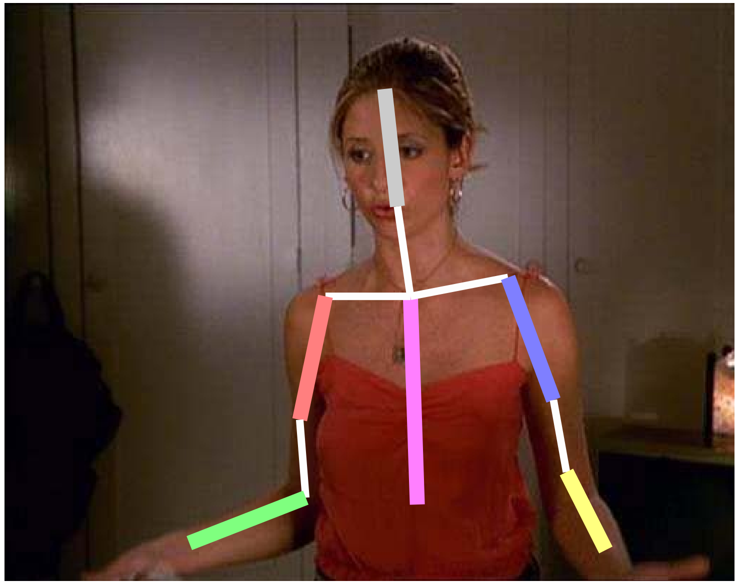
Activities characterized by a pose



Challenges: articulations and deformations

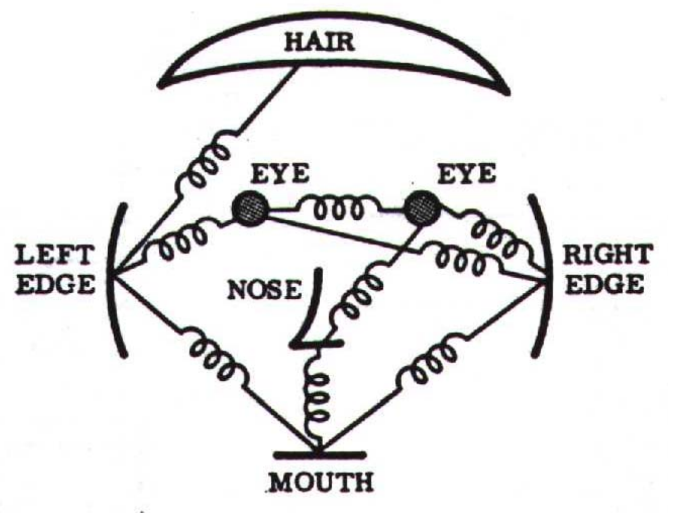
Challenges: of (almost) unconstrained images

varying illumination and low contrast; moving camera and background; multiple people; scale changes; extensive clutter; any clothing

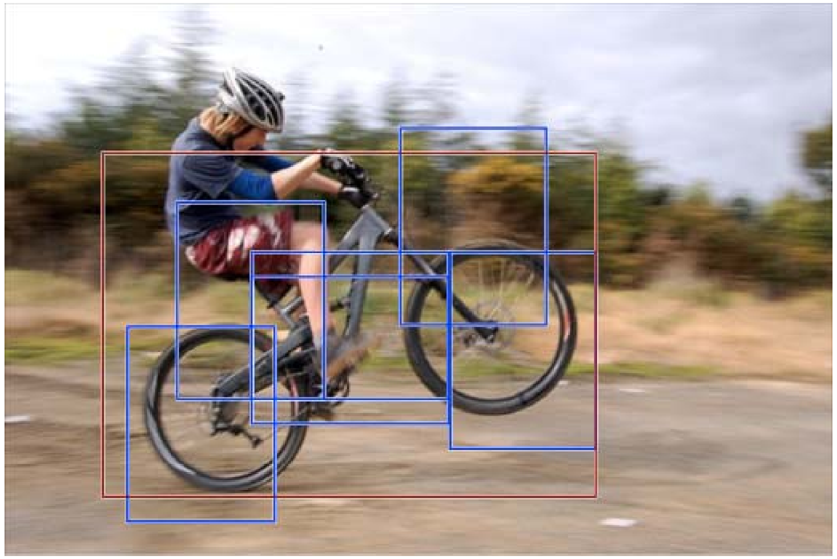
 

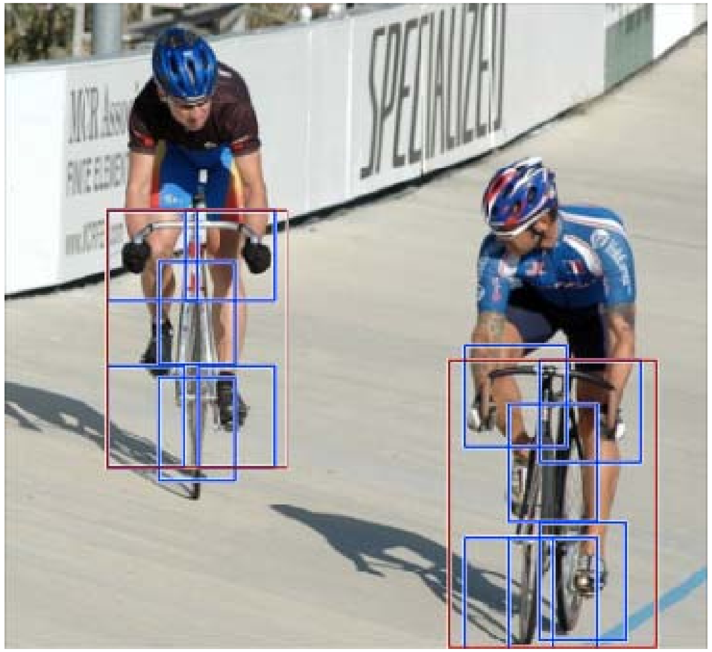
Pictorial Structures

- Intuitive model of an object
- Model has two components
  1. parts (2D image fragments)
  2. structure (configuration of parts)
- Dates back to Fischler & Elschlager 1973



From last lecture: objects





Mixture of deformable part-based models

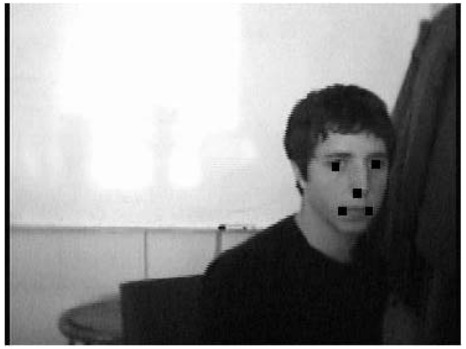
- One component per "aspect" e.g. front/side view

Each component has global template + deformable parts

Discriminative training from bounding boxes alone

Localize multi-part objects at arbitrary locations in an image

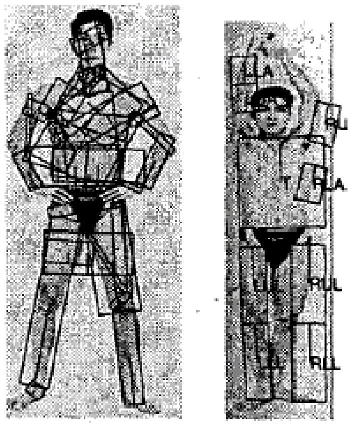
- Generic object models such as person or car
- Allow for articulated objects
- Simultaneous use of appearance and spatial information
- Provide efficient and practical algorithms



To fit model to image: minimize an energy (or cost) function that reflects both

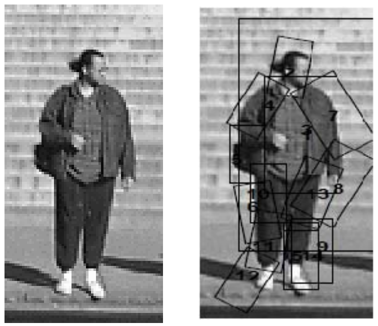
- Appearance: how well each part matches at given location
- Configuration: degree to which parts match 2D spatial layout

Long tradition of using pictorial structures for humans



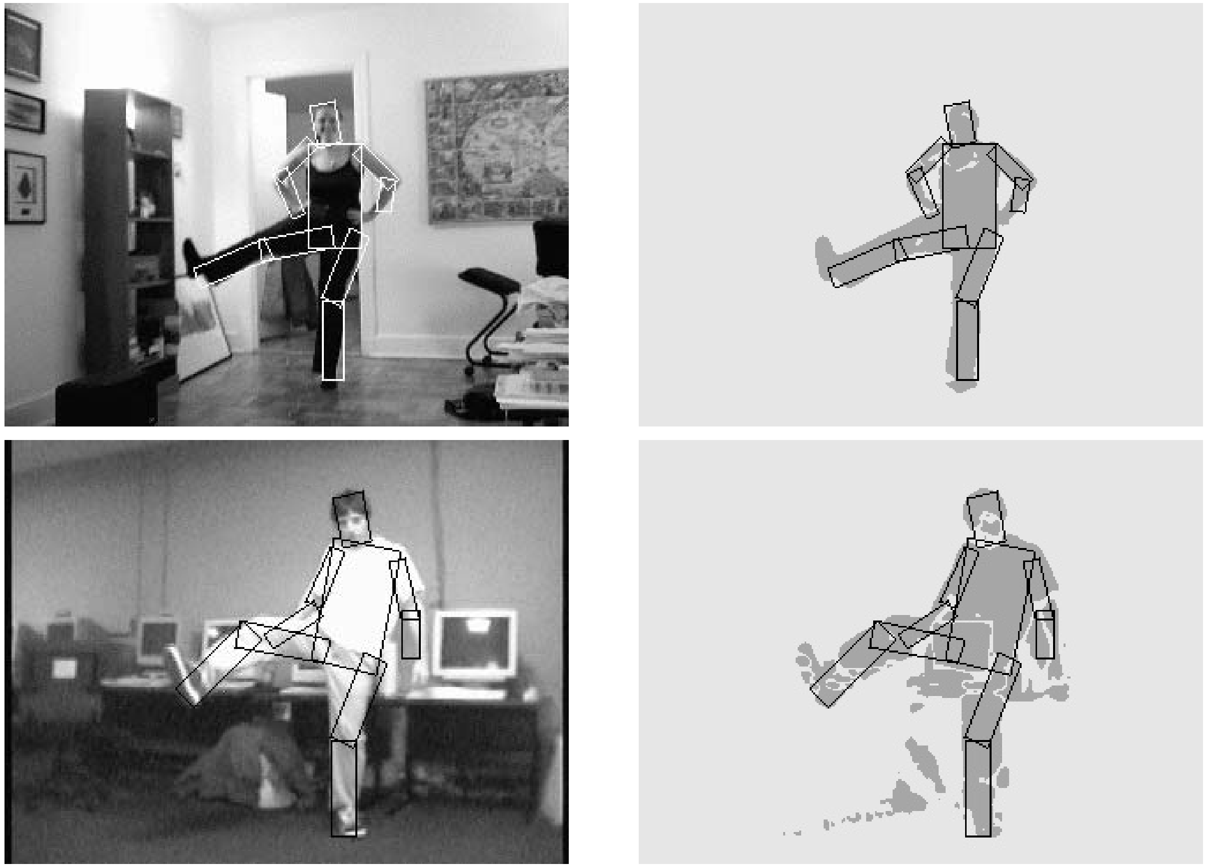
Finding People by Sampling
Ioffe & Forsyth, ICCV 1999

Pictorial Structure Models for Object Recognition
Felzenszwalb & Huttenlocher, 2000



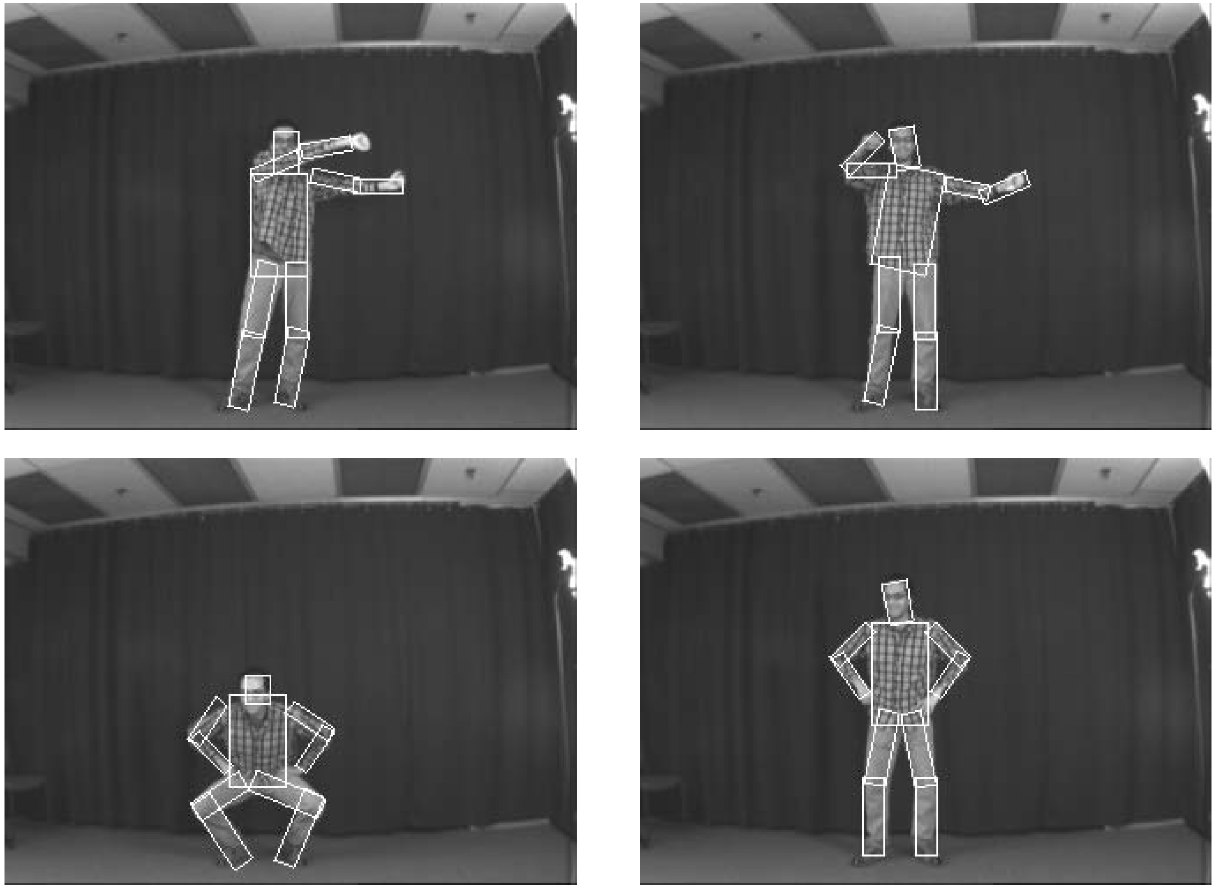
Learning to Parse Pictures of People
Ronfard, Schmid & Triggs, ECCV 2002

Felzenszwalb & Huttenlocher

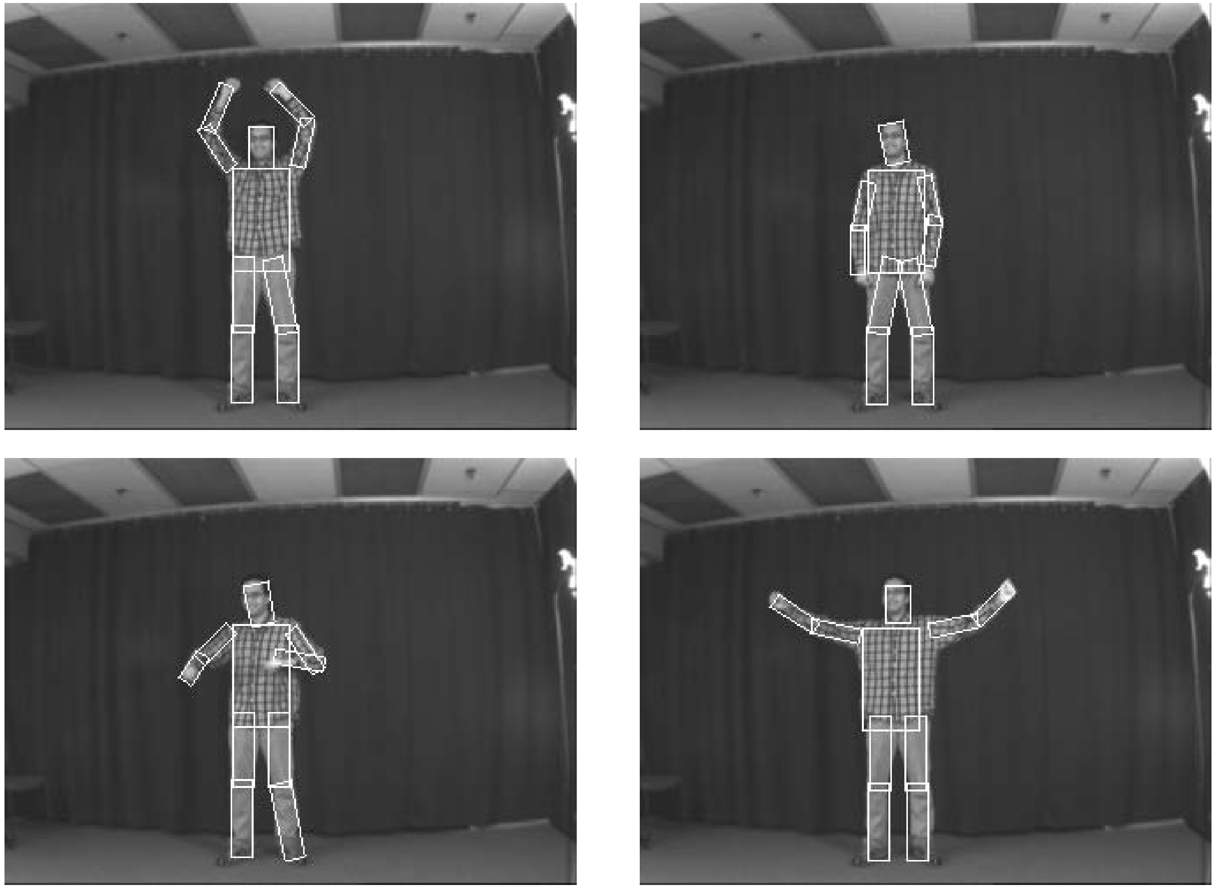


NB: requires background subtraction

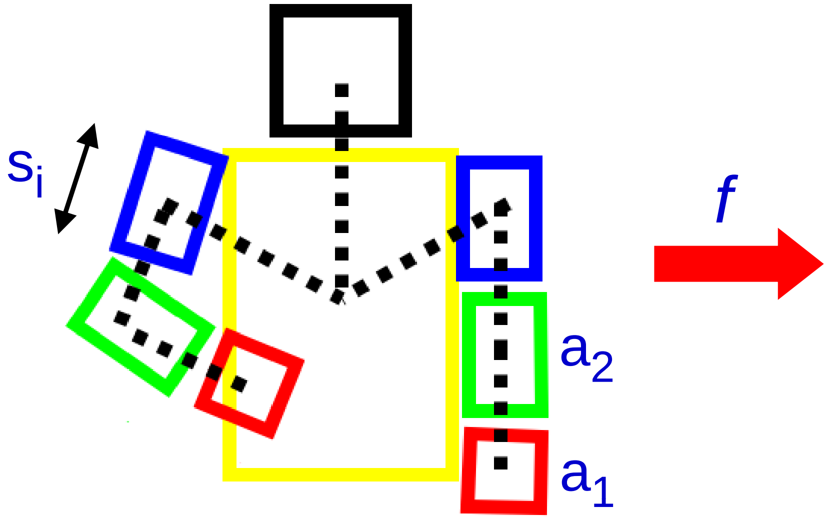
Variety of Poses

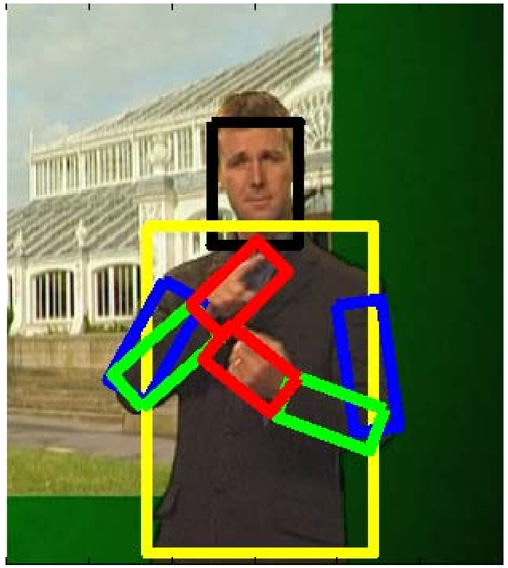


Variety of Poses



Objective: detect human and determine upper body pose (layout)

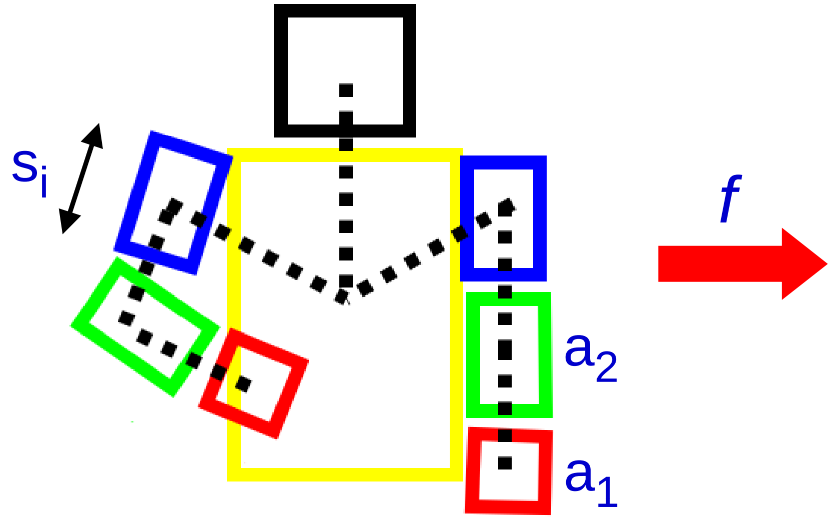


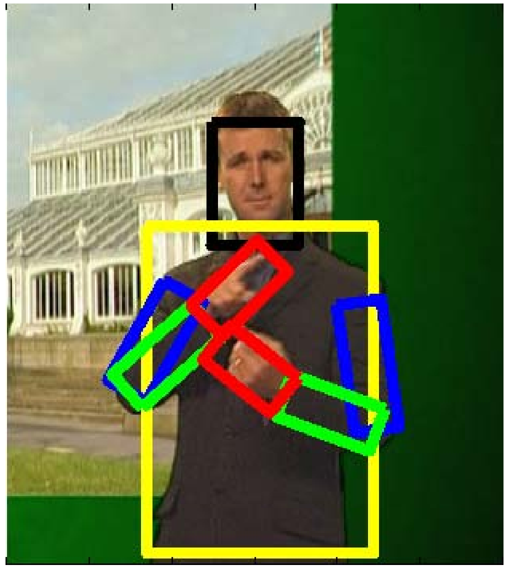


Model as a graph labelling problem

- Vertices $\mathcal{V}$ are parts, $a_i, i = 1, \cdots, n$
- Edges $\mathcal{E}$ are pairwise linkages between parts
- For each part there are h possible poses $\mathbf{p}_j = (x_j, y_j, \phi_j, s_j)$
- Label each part by its pose: $f : \mathcal{V} \longrightarrow \{1, \cdots, h\}$, i.e. part a takes pose $\mathbf{p}_{f(a)}$.

Pictorial structure model – CRF





- Each labelling has an energy (cost):

$$E(f) = \underbrace{\sum_{a \in \mathcal{V}} \theta_{a;f(a)}}_{\substack{\text{unary terms} \\ \text{(appearance)}}} + \underbrace{\sum_{(a,b) \in \mathcal{E}} \theta_{ab;f(a)f(b)}}_{\substack{\text{pairwise terms} \\ \text{(configuration)}}}$$

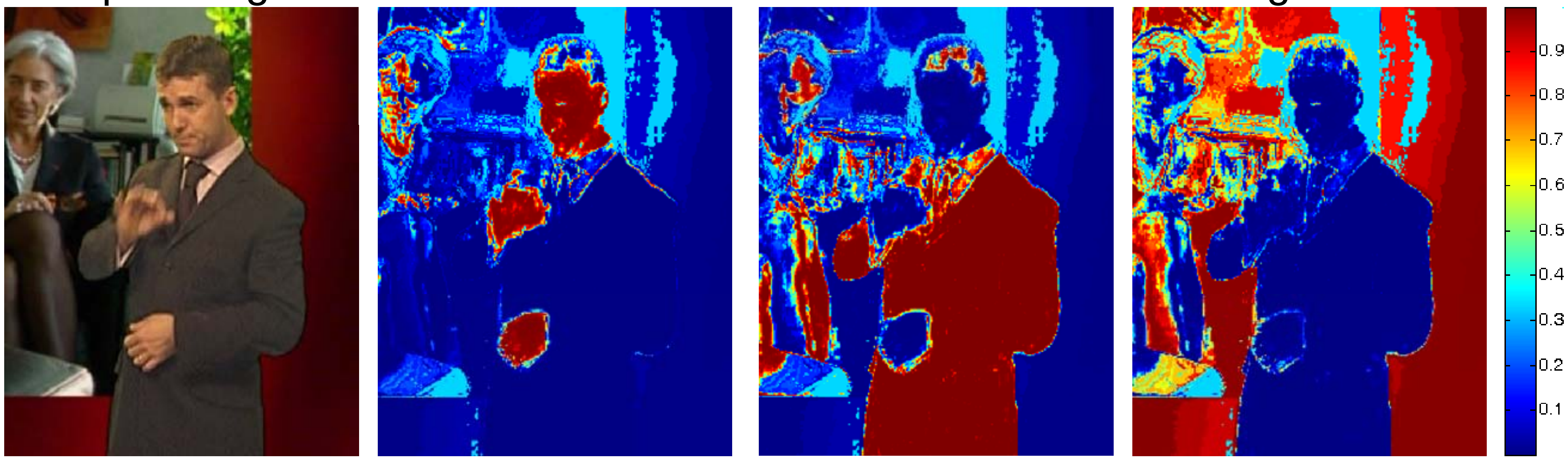
Features for unary:
- colour
- HOG

for limbs/torso

- Fit model (inference) as labelling with lowest energy

Unary term: appearance feature I - colour

input image skin torso background

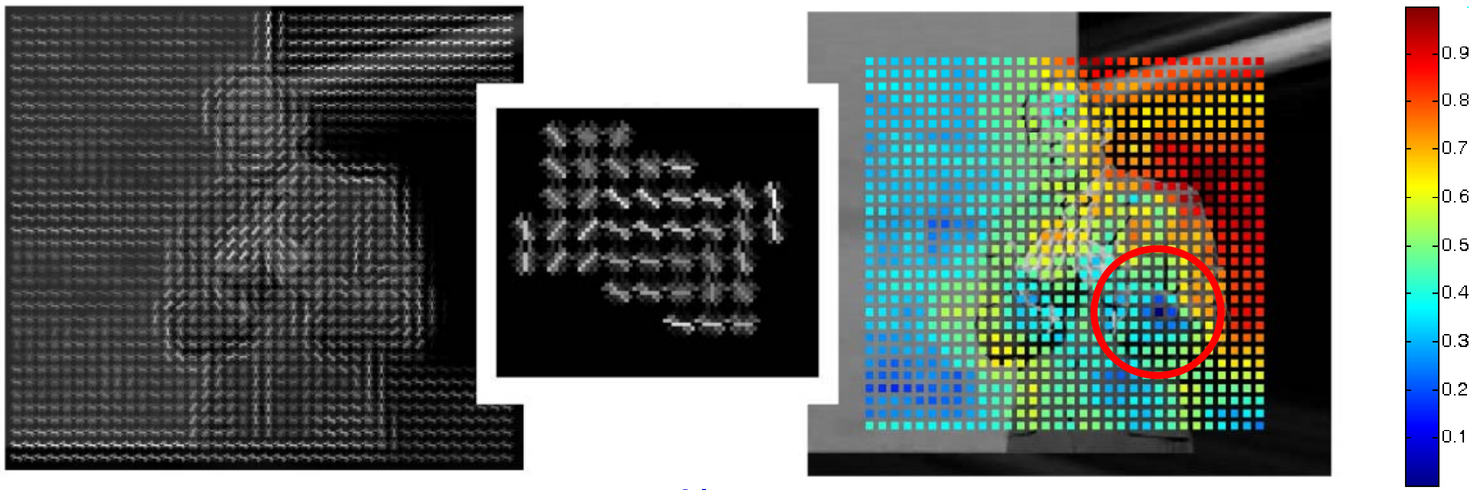


colour posteriors

Unary term: appearance feature II - HOG

Dalal & Triggs, CVPR 2005

Histogram of oriented gradients (HOG)



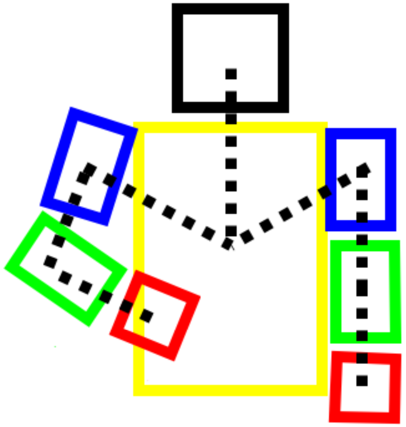
HOG of image

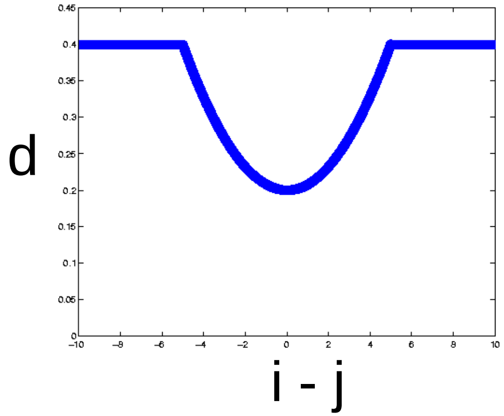
HOG of lower arm template (learned)

L2 Distance

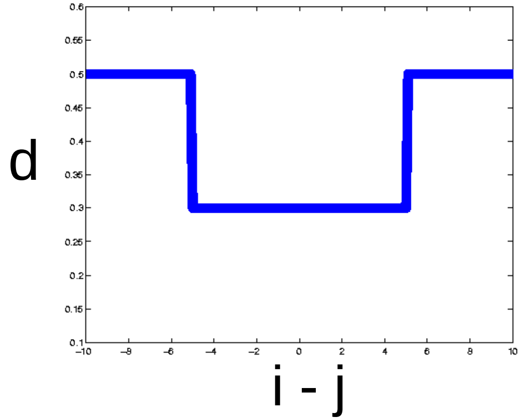
Pairwise terms: kinematic layout

$$\theta_{ab;ij} = w_{ab} d(|i-j|)$$



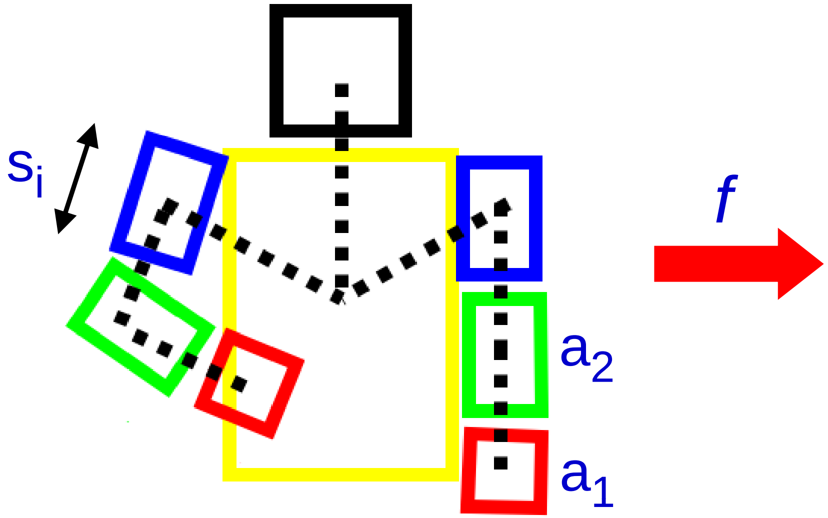


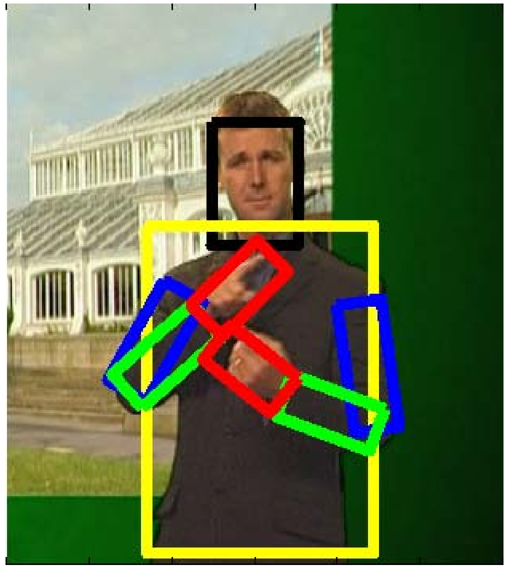
Truncated Quadratic



Potts

Pictorial structure model – CRF





- Each labelling has an energy (cost):

$$E(f) = \underbrace{\sum_{a \in \mathcal{V}} \theta_{a;f(a)}}_{\substack{\text{unary terms} \\ \text{(appearance)}}} + \underbrace{\sum_{(a,b) \in \mathcal{E}} \theta_{ab;f(a)f(b)}}_{\substack{\text{pairwise terms} \\ \text{(configuration)}}}$$

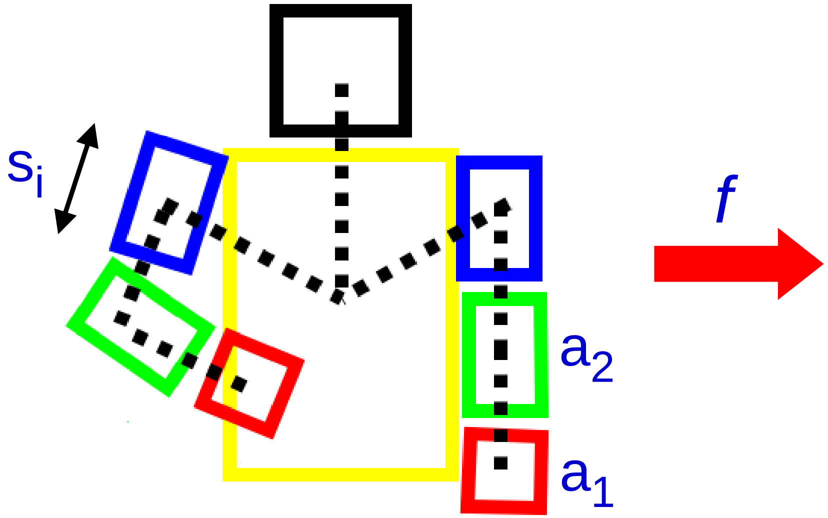
Features for unary:

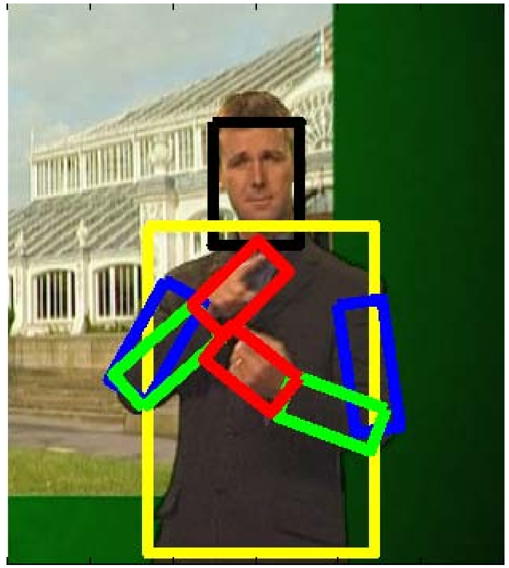
- colour
- HOG

for limbs/torso

- Fit model (inference) as labelling with lowest energy

Complexity



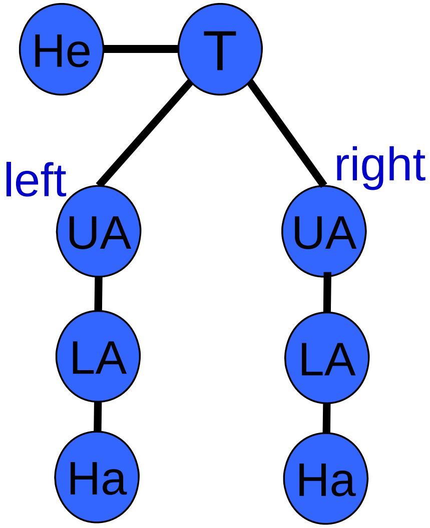


- n parts
- For each part there are h possible poses $\mathbf{p}_j = (x_j, y_j, \phi_j, s_j)$
- There are h^n possible labellings

Problem: any reasonable discretization (e.g. 12 scales and 36 angles for upper and lower arm, etc) gives a number of configurations 10^12 – 10^14

→ Brute force search not feasible

Are trees the answer?



- With n parts and h possible discrete locations per part, $O(h^n)$
- For a tree, using dynamic programming this reduces to $O(nh^2)$
- If model is a tree and has certain edge costs, then complexity reduces to O(nh) using a distance transform [Felzenszwalb & Huttenlocher, 2000, 2005]

Problems with tree structured pictorial structures

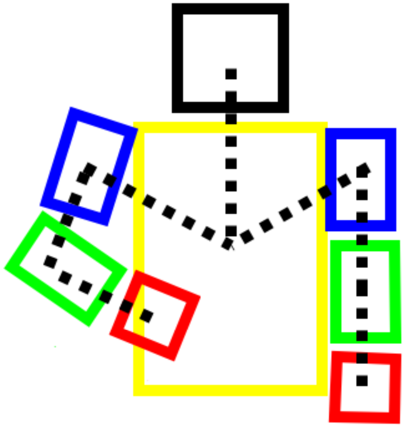
- Layout model defines the foreground,
i.e. it chooses the pixels to "explain"

- ignores skin and strong edge in background

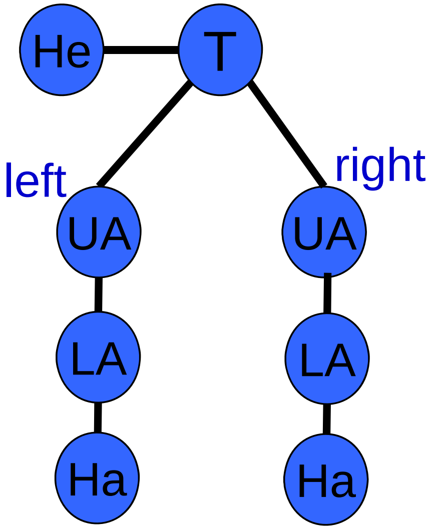
- "double counting"

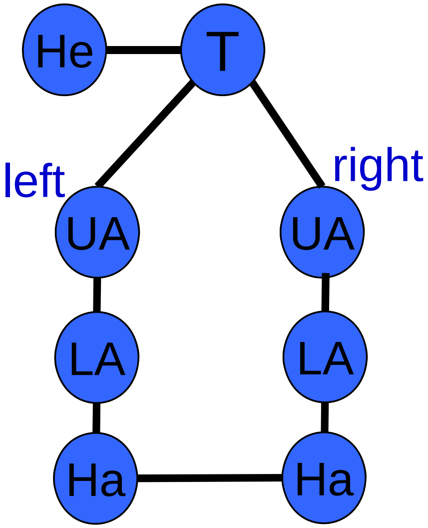
Generative model of foreground only

Kinematic structure vs graphical (independence) structure



Graph G = (V,E)

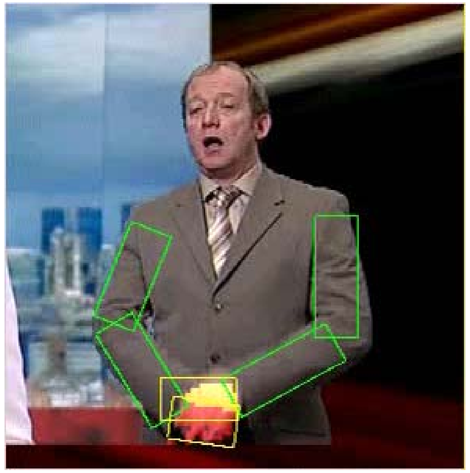




Requires more connections than a tree

Some recent results

- Detect hands and arms of person signing British Sign Language
- Hour long sequences



- Strong but minimal supervision

[Buehler, Everingham, Zisserman CVPR09]

Search space reduction by upper body human detection

(1) detect human; (2) reduce search from h^n

Train

Idea

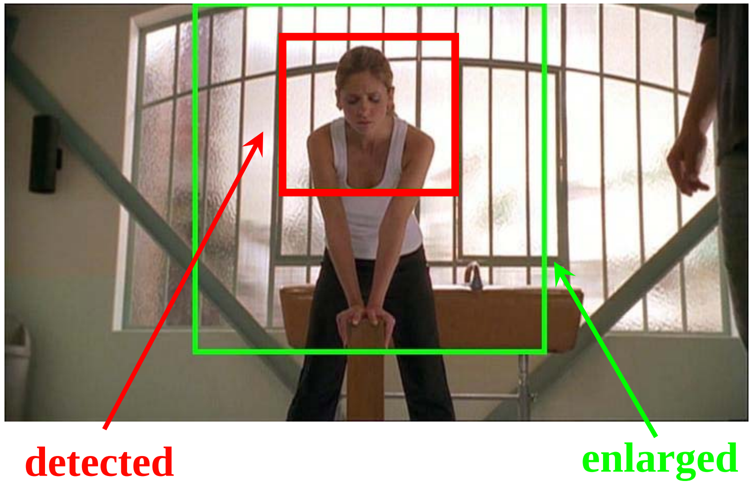
get approximate location and scale with a detector generic over pose and appearance

Building an upper-body detector

- based on Dalal and Triggs CVPR 2005

- train = 96 frames X 12 perturbations

Test



Benefits for pose estimation

+ fixes scale of body parts

+ sets bounds on x,y locations

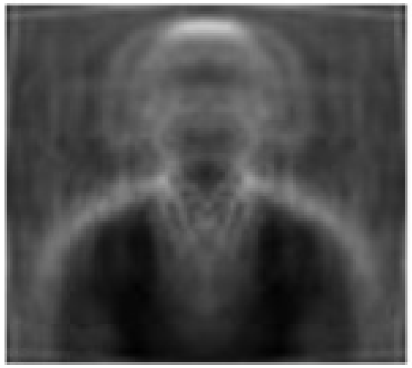
+ detects also back views

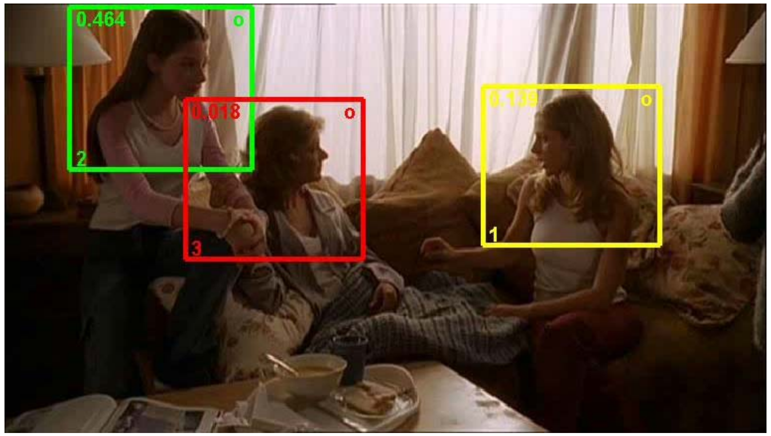
+ fast

- little info about pose (arms)

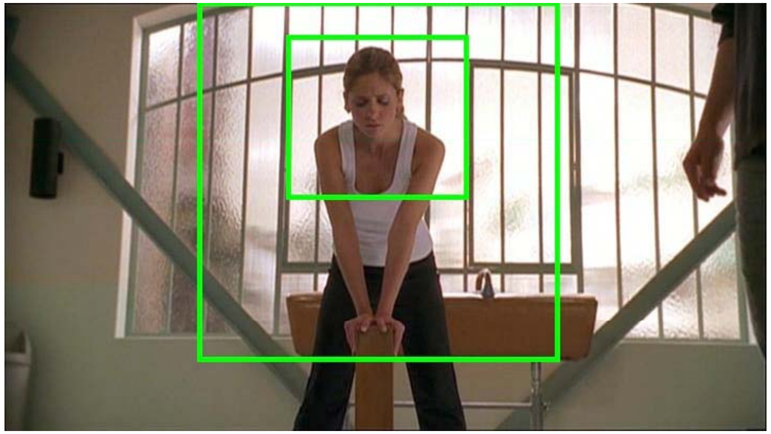
Upper body detector – using HOGs

average training data





Search space reduction by foreground highlighting



initialization

output

Idea

exploit knowledge about structure of search area to initialize Grabcut

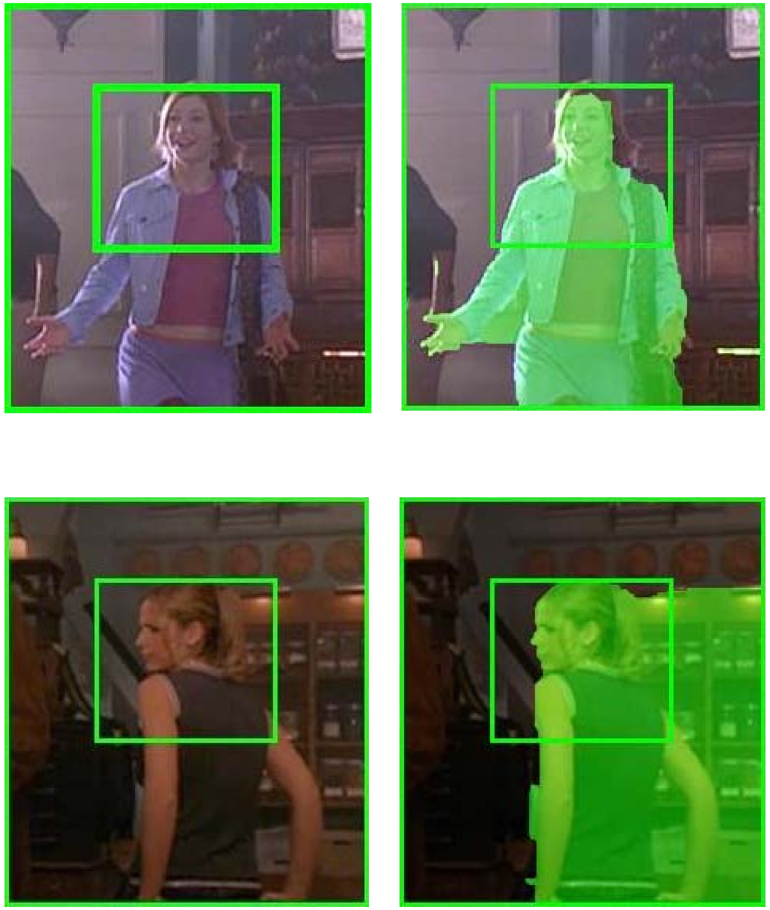
Initialization

- learn fg/bg models from regions where person likely present/absent
- clamp central strip to fg
- don't clamp bg (arms can be anywhere)

Benefits for pose estimation

+ further reduce clutter
+ conservative (no loss 95.5% times)
+ needs no knowledge of background
+ allows for moving background

Search space reduction by foreground highlighting



Idea

exploit knowledge about structure of search area to initialize Grabcut

Initialization

- learn fg/bg models from regions where person likely present/absent
- clamp central strip to fg
- don't clamp bg (arms can be anywhere)

Benefits for pose estimation

+ further reduce clutter
+ conservative (no loss 95.5% times)
+ needs no knowledge of background
+ allows for moving background

Pose estimation by image parsing - Ramanan NIPS 06

edge parse | appearance | edge + col parse

Goal

estimate posterior of part configuration

$$E(f) = \underbrace{\textstyle\sum_{a\in\mathcal{V}} \theta_{a;f(a)}}_{\substack{\text{unary terms}\\ \text{(edges/colour)}}} + \underbrace{\textstyle\sum_{(a,b)\in\mathcal{E}} \theta_{ab;f(a)f(b)}}_{\substack{\text{pairwise terms}\\ \text{(configuration)}}}$$

Algorithm

1. inference with edges unary
2. learn appearance models of body parts and background
3. inference with edges + colour unary

Advantages of space reduction

+ much more robust
+ much faster (10x-100x)

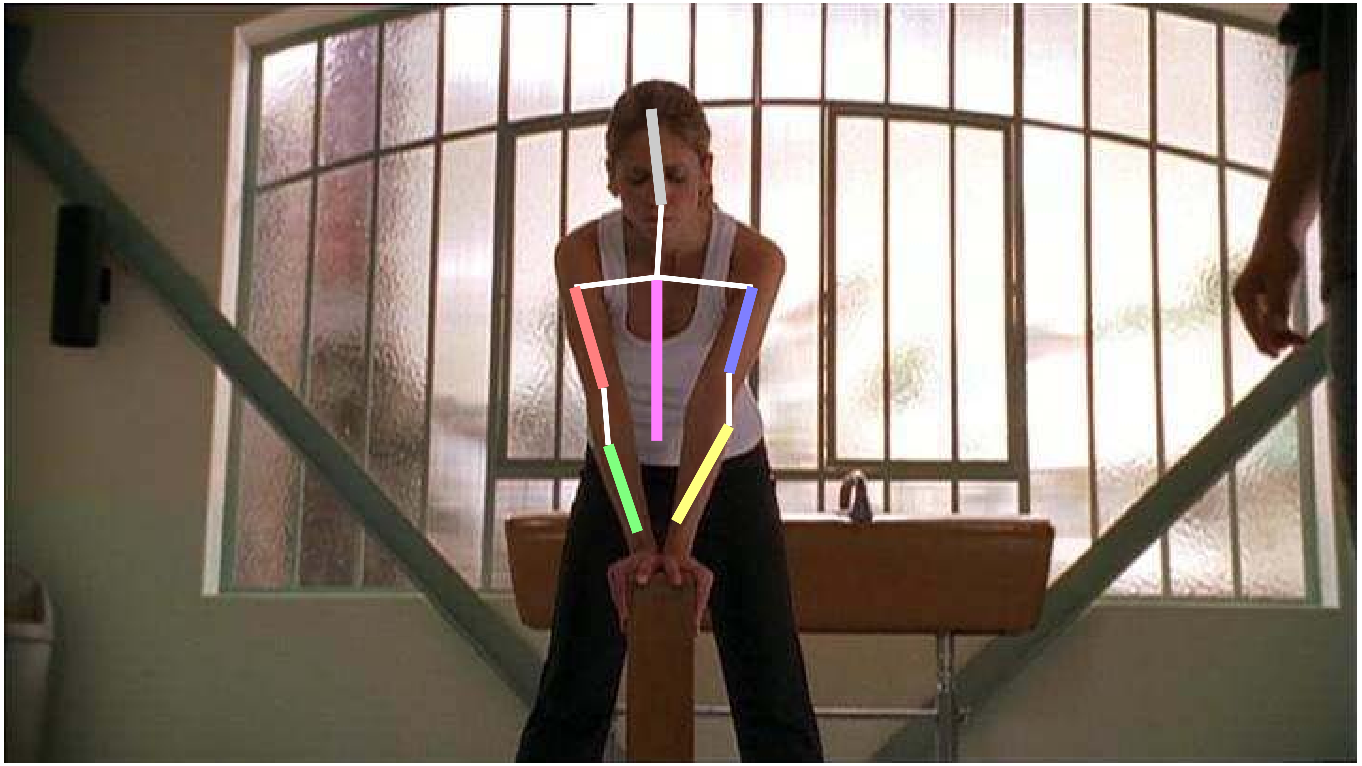
Failure of direct pose estimation

Ramanan NIPS 2006 unaided

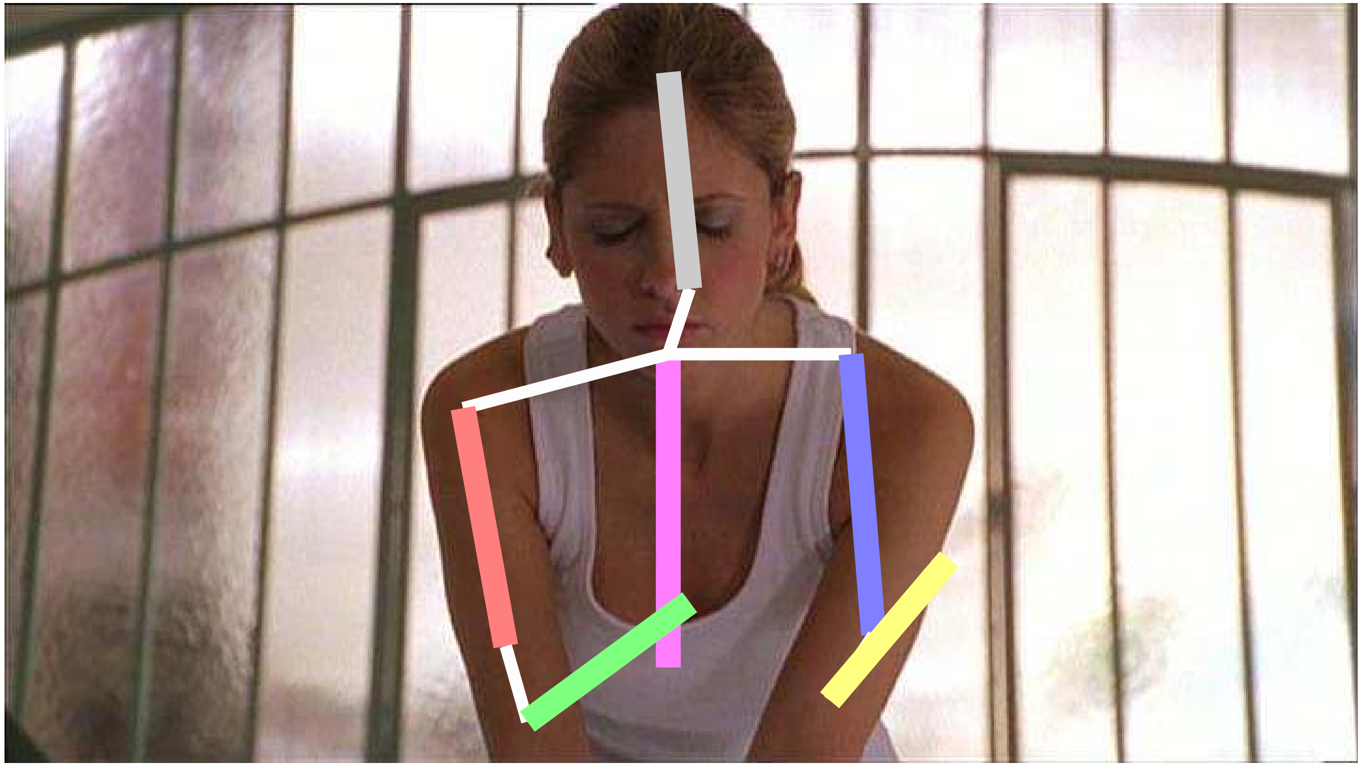
Results on Buffy frames

Results on PASCAL flickr images

What is missed?

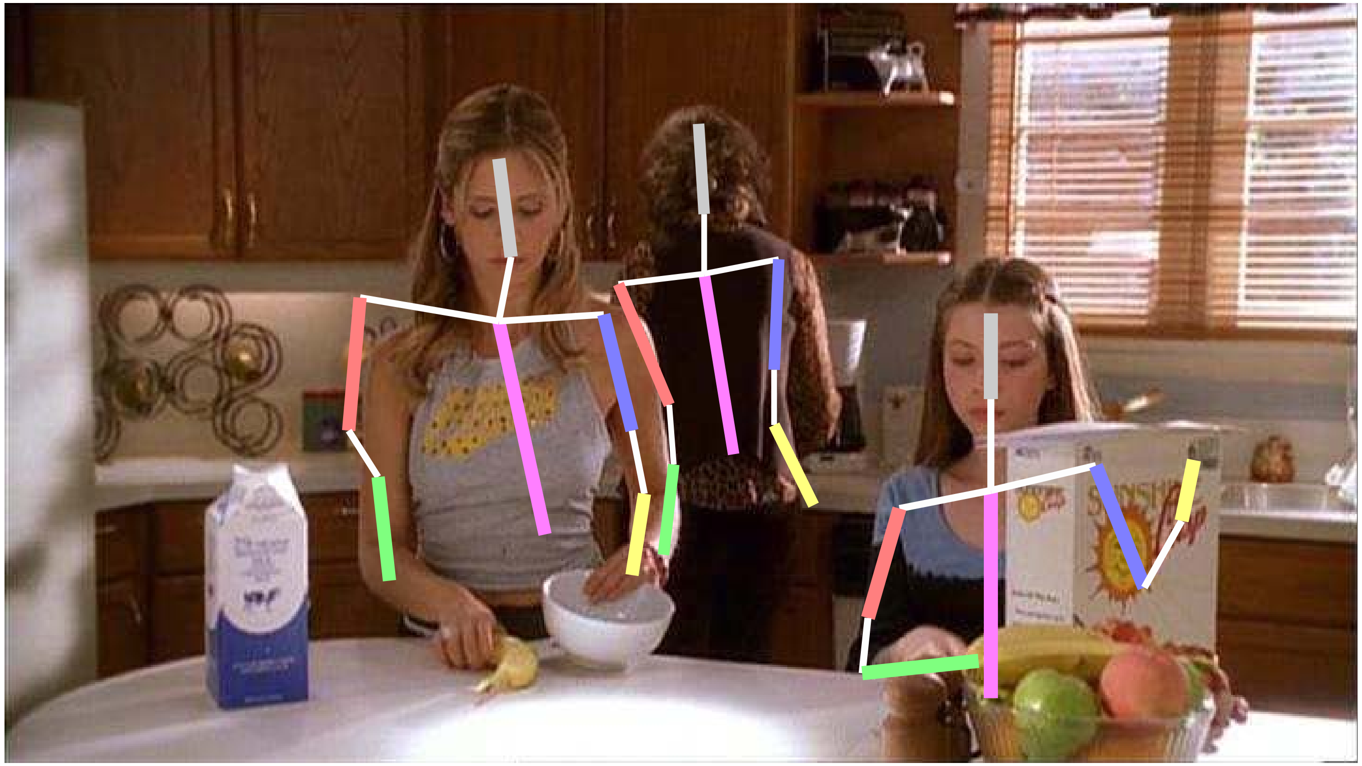


What is missed?



truncation is not modelled

What is missed?



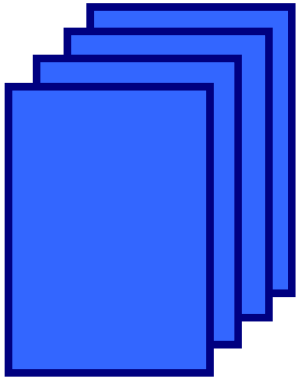
occlusion is not modelled

Application: Pose Search

Given user-selected
query frame+person …

query

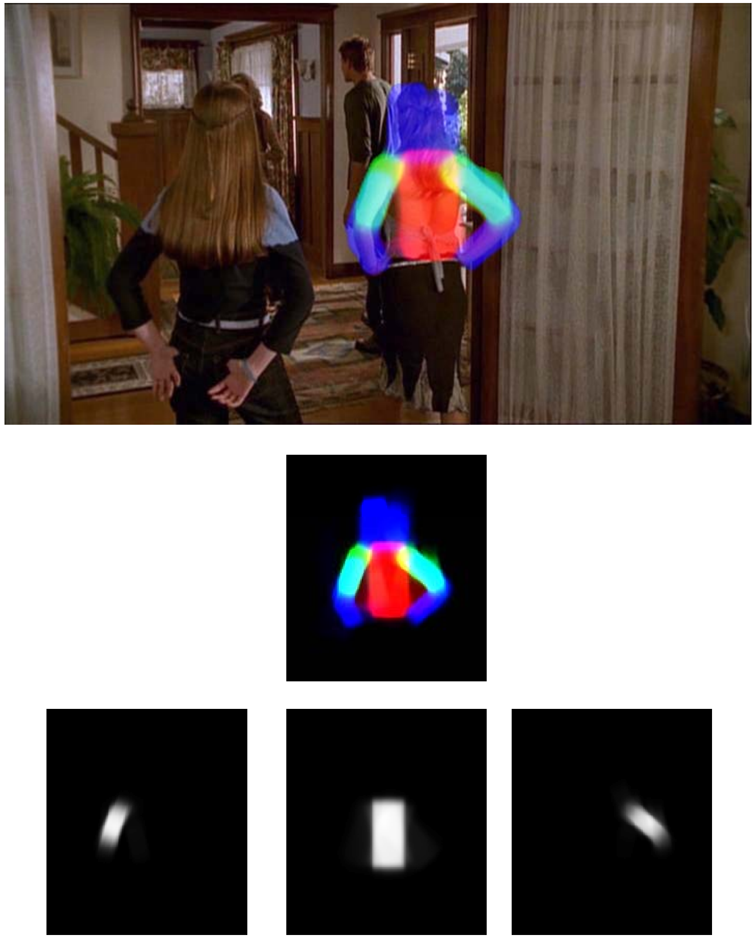
… retrieve shots with persons
in the same pose from video database



video database

CVPR 2009

Pose Search



Pose descriptors

- soft-segmentations of body parts

- distributions over orient+location
 for parts and pairs of parts

Similarity measures

- dot-product (= soft intersection)

- Batthacharrya / Chi-square

Processing

Off-line:

- Detect upper bodies in every frame
- Link (track) upper body detections
- Estimate upper body pose for each frame of track
- Compute descriptor (vector) for each upper body pose

Run-time:

- Rank each track by its similarity to the query pose

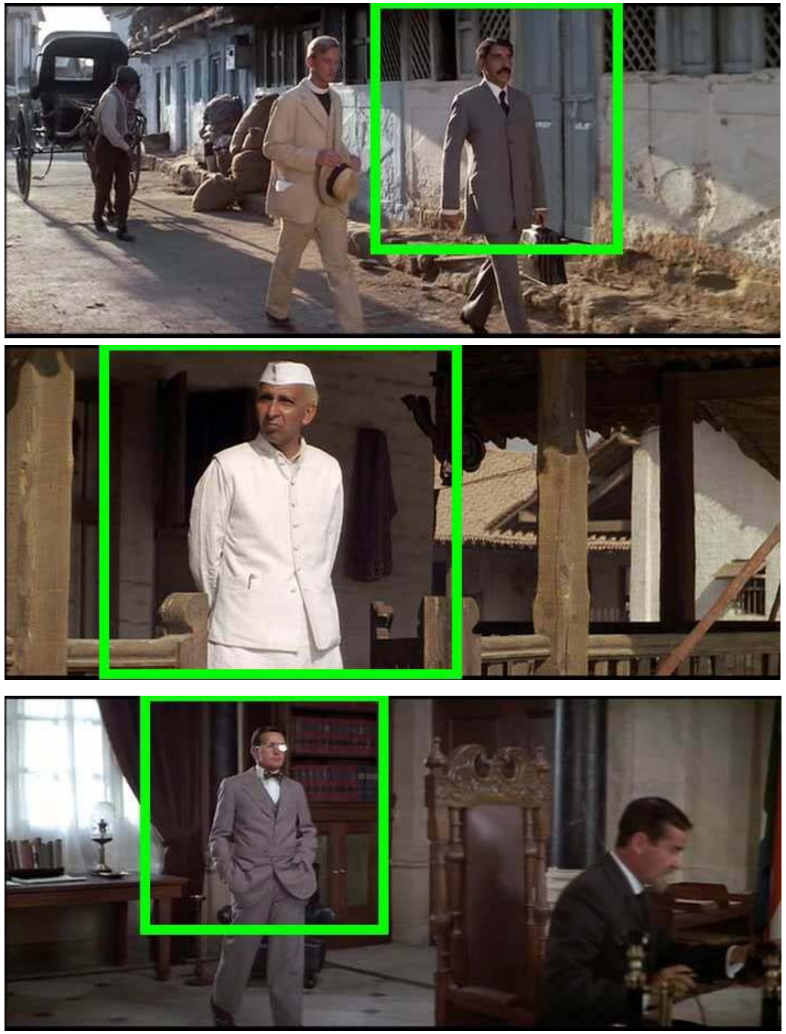
Pose Search

“hips pose”

Pose Search

"rest pose"

Pose Search



“rest pose”

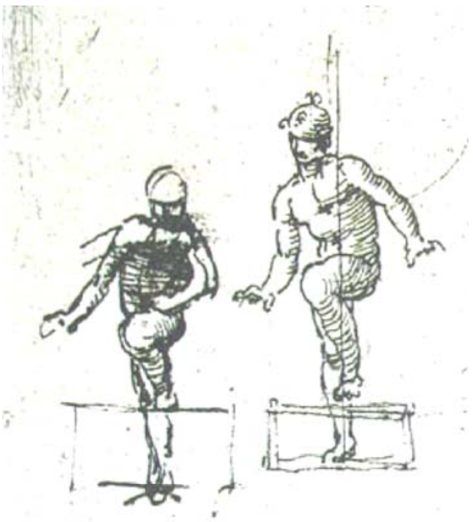
Other poses – query interesting pose

Hollywood movies – Query on Gandhi, Search Hugh Grant opus

Other poses – query interesting pose

Hollywood movies – Query on Gandhi, Search Hugh Grant opus

Class overview



Motivation

Historic review
Modern applications

Human Pose Estimation

Pictorial structures
Learning models from image data
Recent advances

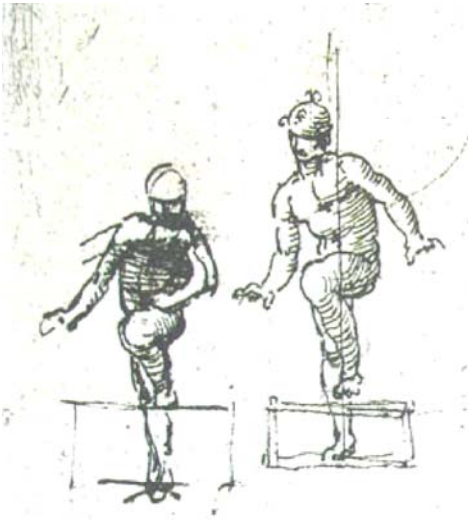
Appearance-based methods

Motion history images
Active shape models
Motion priors

Motion-based methods

Generic and parametric Optical Flow
Motion templates

Class overview



Motivation

Historic review
Modern applications

Human Pose Estimation

Pictorial structures
Learning models from image data
Recent advances

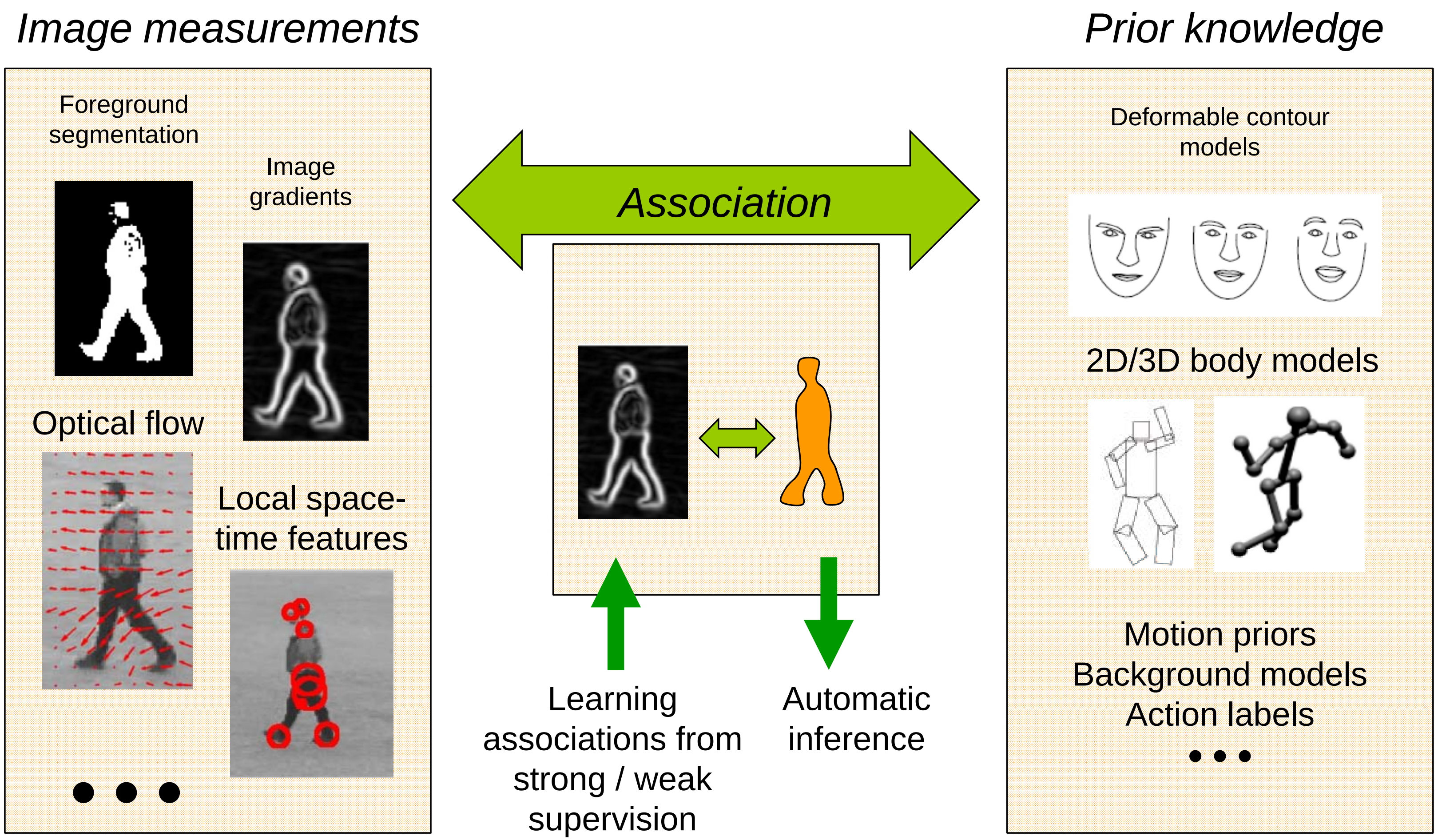
Appearance-based methods

Motion history images
Active shape models
Motion priors

Motion-based methods

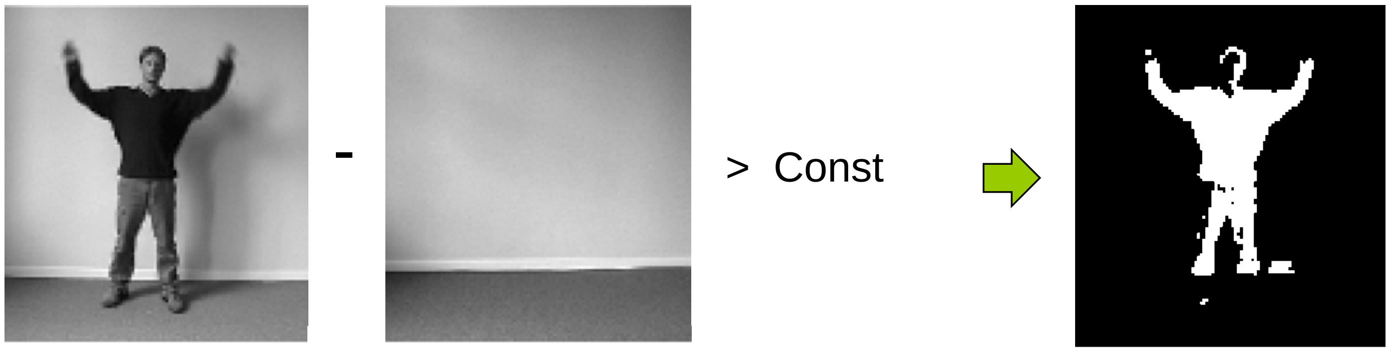
Generic and parametric Optical Flow
Motion templates

Action understanding: Key components



Foreground segmentation

Image differencing: a simple way to measure motion/change

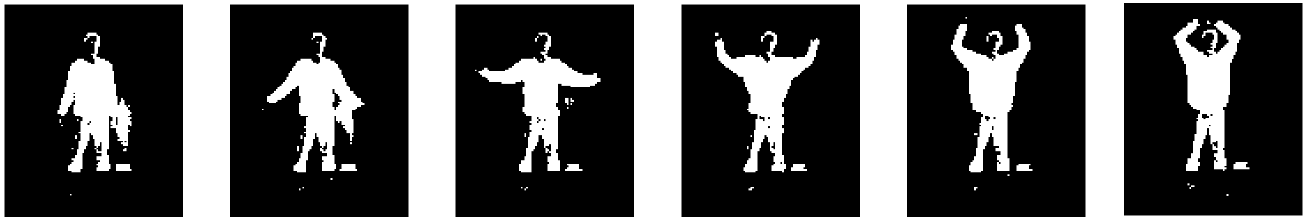


Better Background / Foreground separation methods exist:

- 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

Temporal Templates

$D(x, y, t) \quad t = 1, ..., T$



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

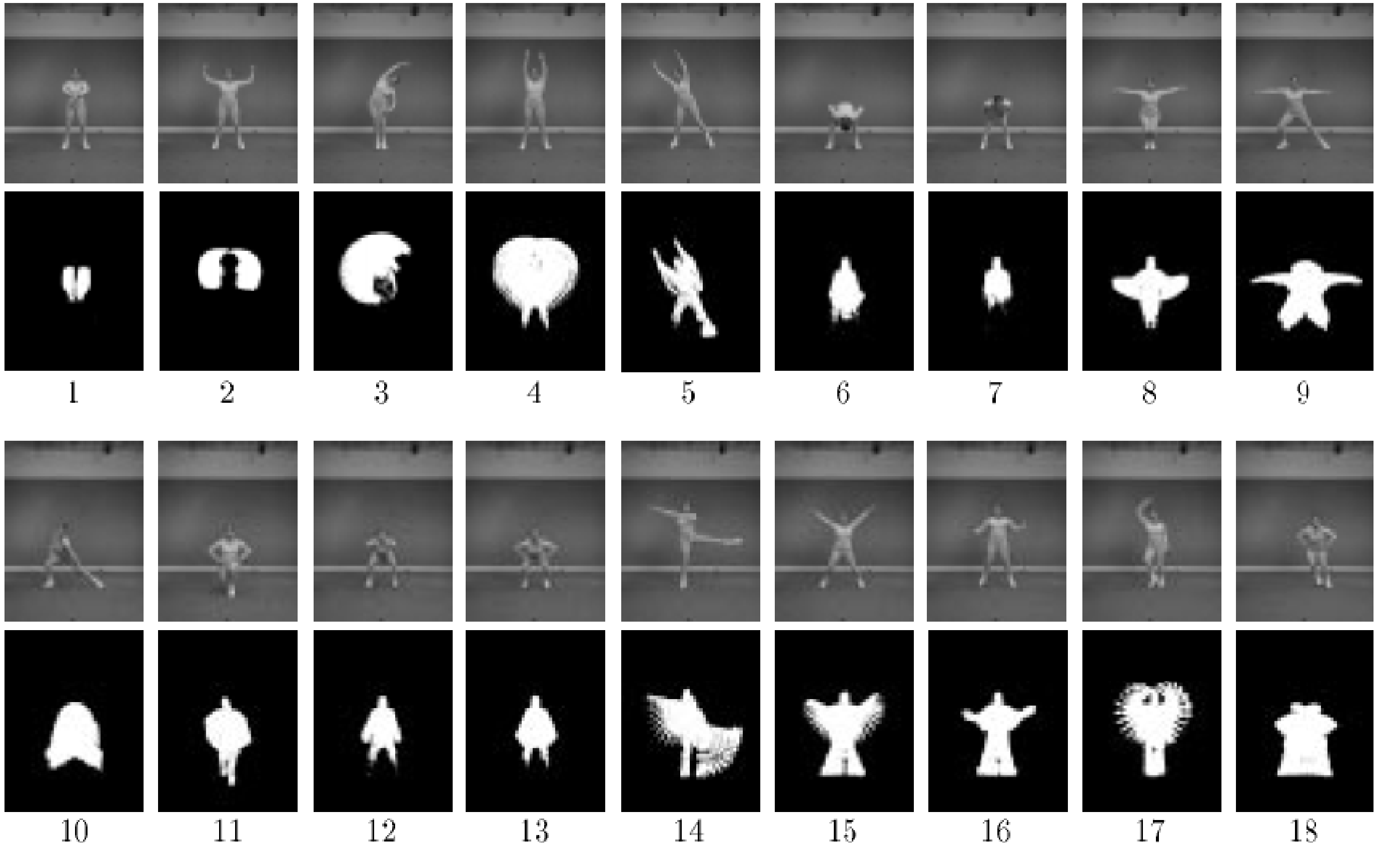
$$H_\tau(x, y, t) = \begin{cases} \tau & \text{if } D(x, y, t) = 1 \\ \max\ (0, H_\tau(x, y, t-1) - 1) & \text{otherwise} \end{cases}$$

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

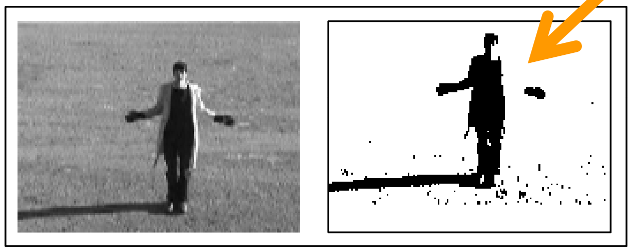
Pros:

+ Simple and fast
+ Works in controlled settings

Not all shapes are valid ⇒ Restrict the space of admissible silhouettes

Cons:

- Prone to errors of background subtraction



Variations in light, shadows, clothing…

What is the background here?

- Does not capture *interior* motion and shape

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*

$$\mathbf{x} = (x_1, \ldots, x_n, y_1, \ldots, y_n)^T$$

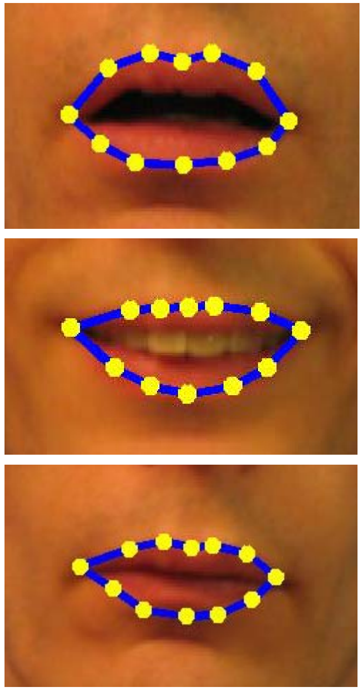
- Assume each shape can be represented by the linear combination of basis shapes

$$\mathbf{\Phi} = (\phi_1 | \phi_2 | \ldots | \phi_t)$$

such that $\mathbf{x} \approx \bar{\mathbf{x}} + \mathbf{\Phi}\mathbf{b}$

for mean shape $\bar{\mathbf{x}} = \frac{1}{s} \sum_{i=1}^{s} \mathbf{x}_i$

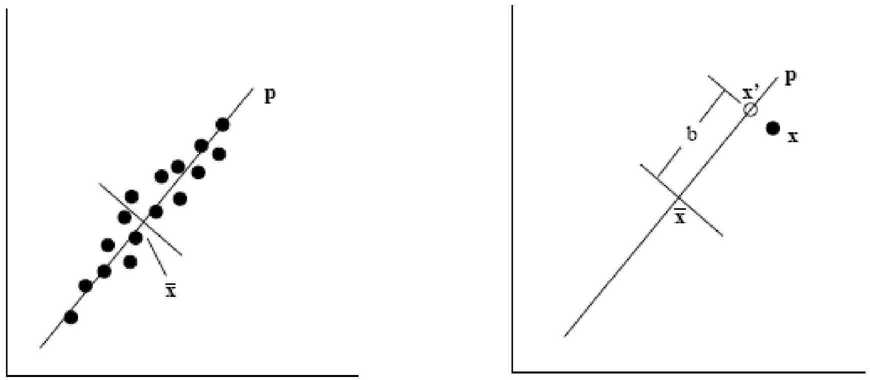
and some parameters $\mathbf{b}$



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 N-Dim space)



Principle Component Analysis (PCA):

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

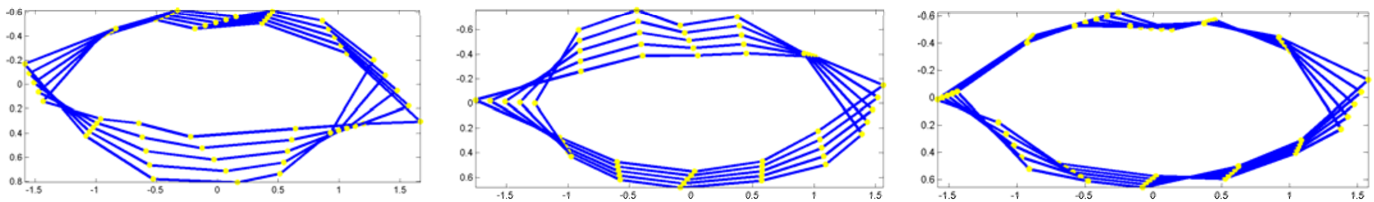
Eigenvectors $\mathbf{\Phi} = (\phi_1|\phi_2|\ldots|\phi_t)$ eigenvalues $\lambda_1, \ldots, \lambda_t$

Active Shape Models of Cootes et al.

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

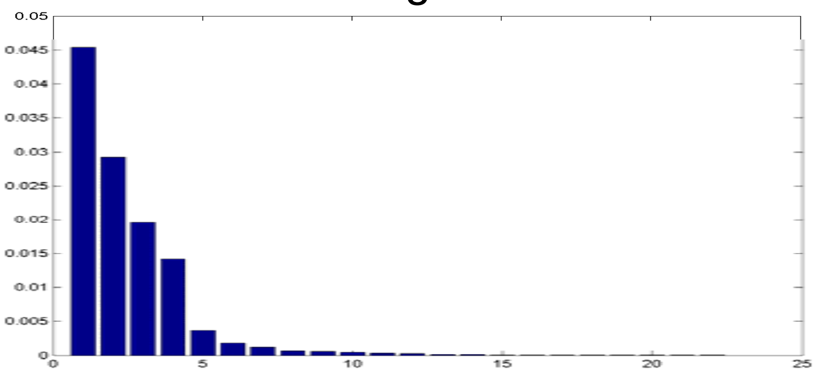
Three main modes of lips-shape variation:

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



$\mu = -3, 1.5, 0, 1.5, 3$

Distribution of eigenvalues: $\lambda_1, \lambda_2, \lambda_3, ...$



A small fraction of basis shapes (eigenvecors) accounts for the most of shape variation (=> landmarks are redundant)

Active Shape Models of Cootes et al.

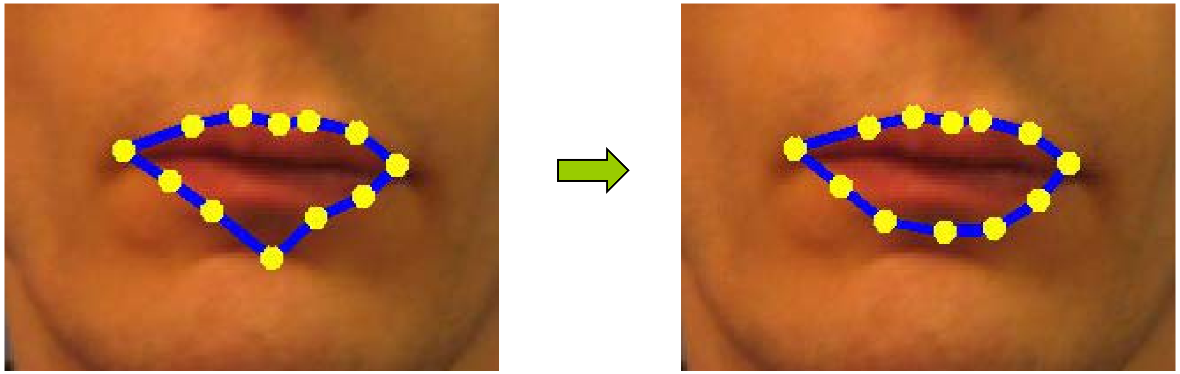
- $\mathbf{\Phi}$ is orthonormal basis, therefore $\mathbf{\Phi}^{-1} = \mathbf{\Phi}^\top$

$\Rightarrow$ Given estimate of $\mathbf{x}$ we can recover shape parameters $\mathbf{b}$

$$\mathbf{b} = \mathbf{\Phi}^\top(\mathbf{x} - \bar{\mathbf{x}})$$

- Projection onto the shape-space serves as a *regularization*

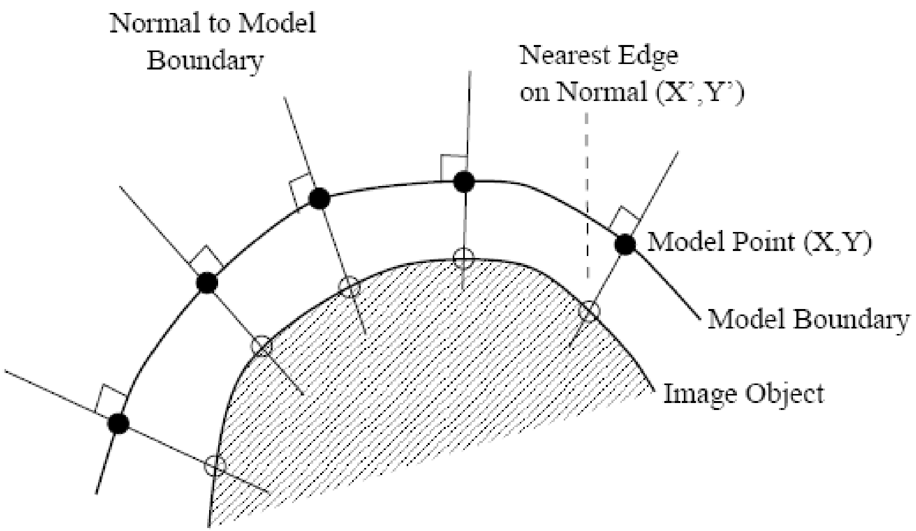
$$\mathbf{x} \quad \Rightarrow \quad \mathbf{b} = \mathbf{\Phi}^\top(\mathbf{x} - \bar{\mathbf{x}}) \quad \Rightarrow \quad \mathbf{x}_{\mathrm{reg}} = \bar{\mathbf{x}} + \mathbf{\Phi}\mathbf{b}$$



Active Shape Models of Cootes et al.

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

$$\mathbf{b}' = \mathbf{\Phi}^\top (\mathbf{x}' - \bar{\mathbf{x}})$$

Active Shape Models of Cootes et al.

- To handle translation, scale and rotation, it is useful to normalize $\mathbf{x}$ prior to shape estimation:

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

using similarity transformation

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

A simple way to estimate $\mathbf{T}$ is to assign (t_x, t_y) and a to the mean position and the standard deviation of points in $\mathbf{x}$ 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 $\bar{\mathbf{x}}, \mathbf{\Phi}$ have to be computed using *normalized* image point coordinates $\mathbf{x}_{\mathrm{norm}} = T^{-1}(\mathbf{x})$

Active Shape Models of Cootes et al.

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

Example: face alignment

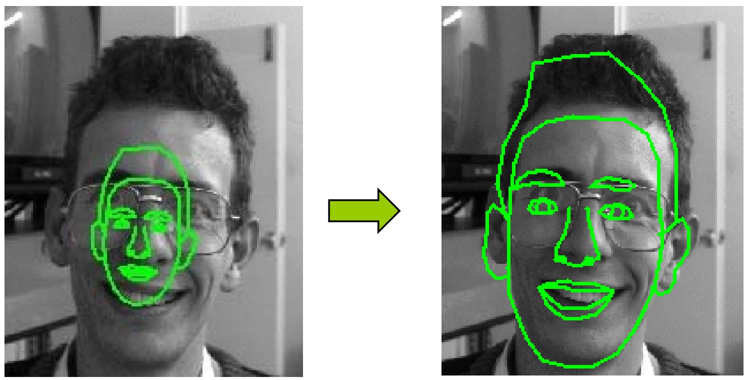
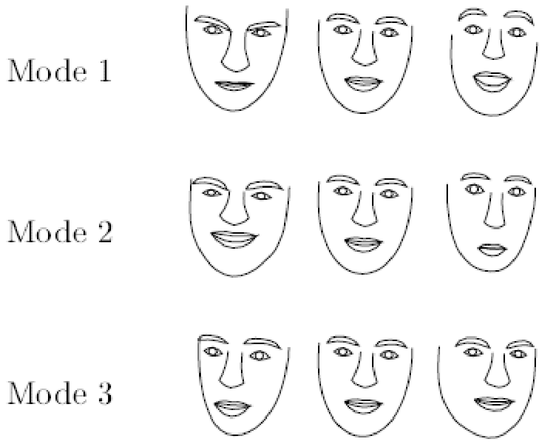


Illustration of face shape space



Active Shape Models: Their Training and Application
T.F. Cootes, C.J. Taylor, D.H. Cooper, and J. Graham, **CVIU** 1995

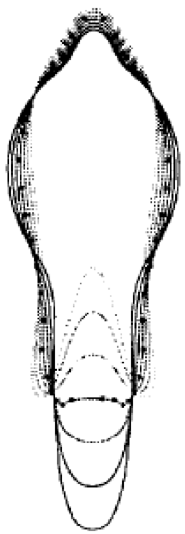
Active Shape Model tracking

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

- Impose time-continuity constraint on model parameters.
  For example, for shape parameters $\mathbf{b}$:

$$b_i^{(k)} = b_i(k-1) + w_i^{k-1}$$

$$w_i \sim \mathcal{N}(0, \mu\lambda_i) \quad \text{Gaussian noise}$$



For similarity transformation $\mathbf{T}$

$$a^{(k)} = a^{(k-1)} + w_a^{k-1}, \quad w_a = \mathcal{N}(0, \sigma_a)$$

$$t_{x|y}^{(k)} = t_{x|y}^{(k-1)} + v_{x|y}^{(k-1)} + w_{x|y}^{k-1}, \quad w_{x|y} = \mathcal{N}(0, \sigma_{x|y})$$

More complex dynamical models possible

- Update model parameters at each time frame using e.g. Kalman filter

Person Tracking

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

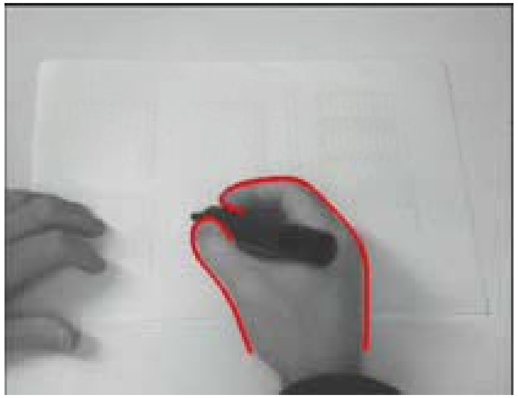
Example:

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

Bayesian Tracking

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

Notation:

$\mathbf{Z}_i$ image data at time i

$\mathbf{X}_i$ model parameters at time i (e.g. shape and its dynamics)

$p(\mathbf{X}_i)$ prior density for $\mathbf{X}_i$

$p(\mathbf{Z}_i|\mathbf{X}_i)$ likelihood of data for the given model configuration

We search posterior defined by the Bayes' rule

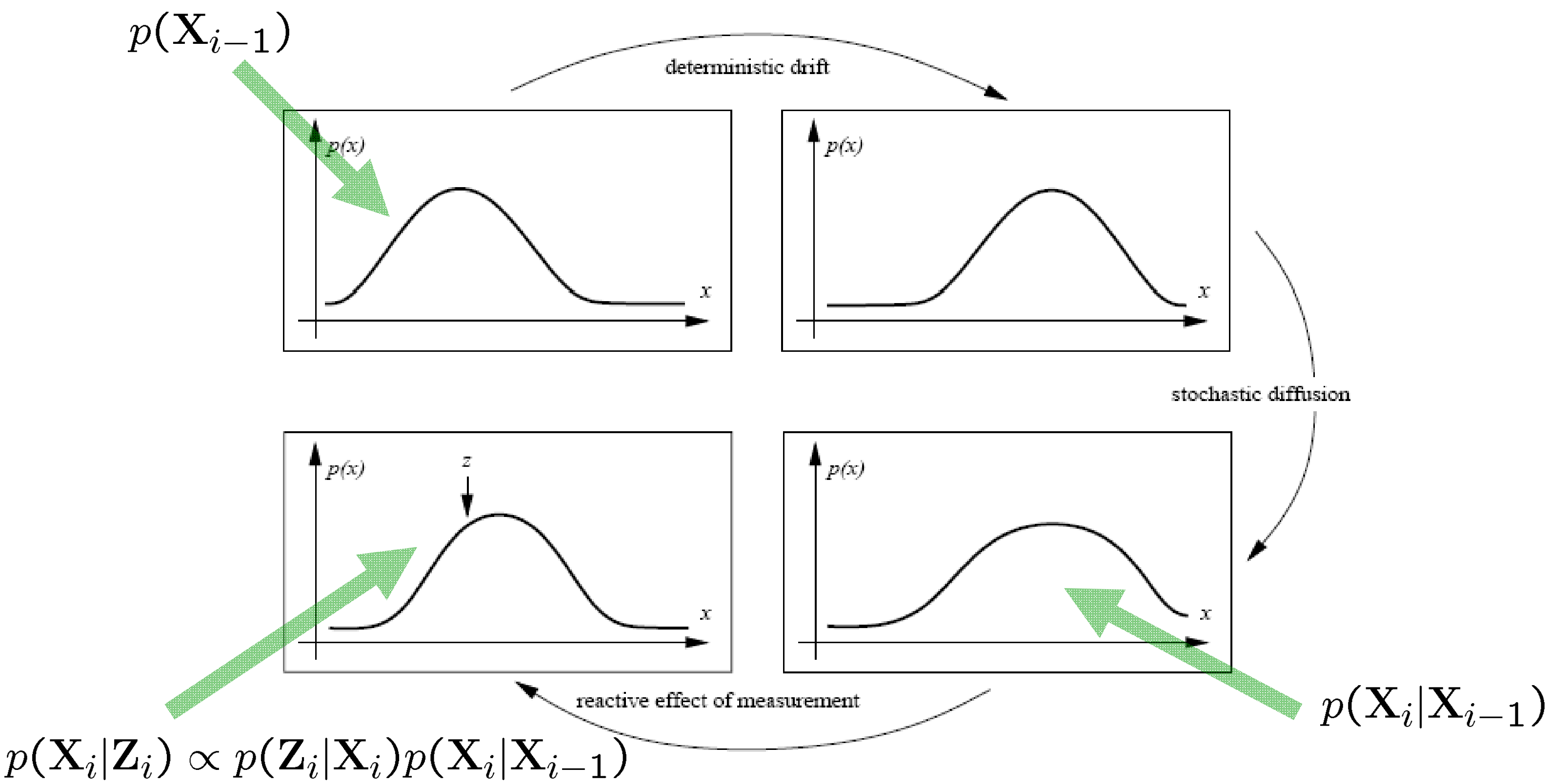
$$p(\mathbf{X}|\mathbf{Z}) \propto \mathbf{p}(\mathbf{Z}|\mathbf{X})\mathbf{p}(\mathbf{X})$$

For tracking the Markov assumption gives the prior $p(\mathbf{X}_i|\mathbf{X}_{i-1})$

Temporal update rule: $p(\mathbf{X}_i|\mathbf{Z}_i) \propto p(\mathbf{Z}_i|\mathbf{X}_i)p(\mathbf{X}_i|\mathbf{X}_{i-1})$

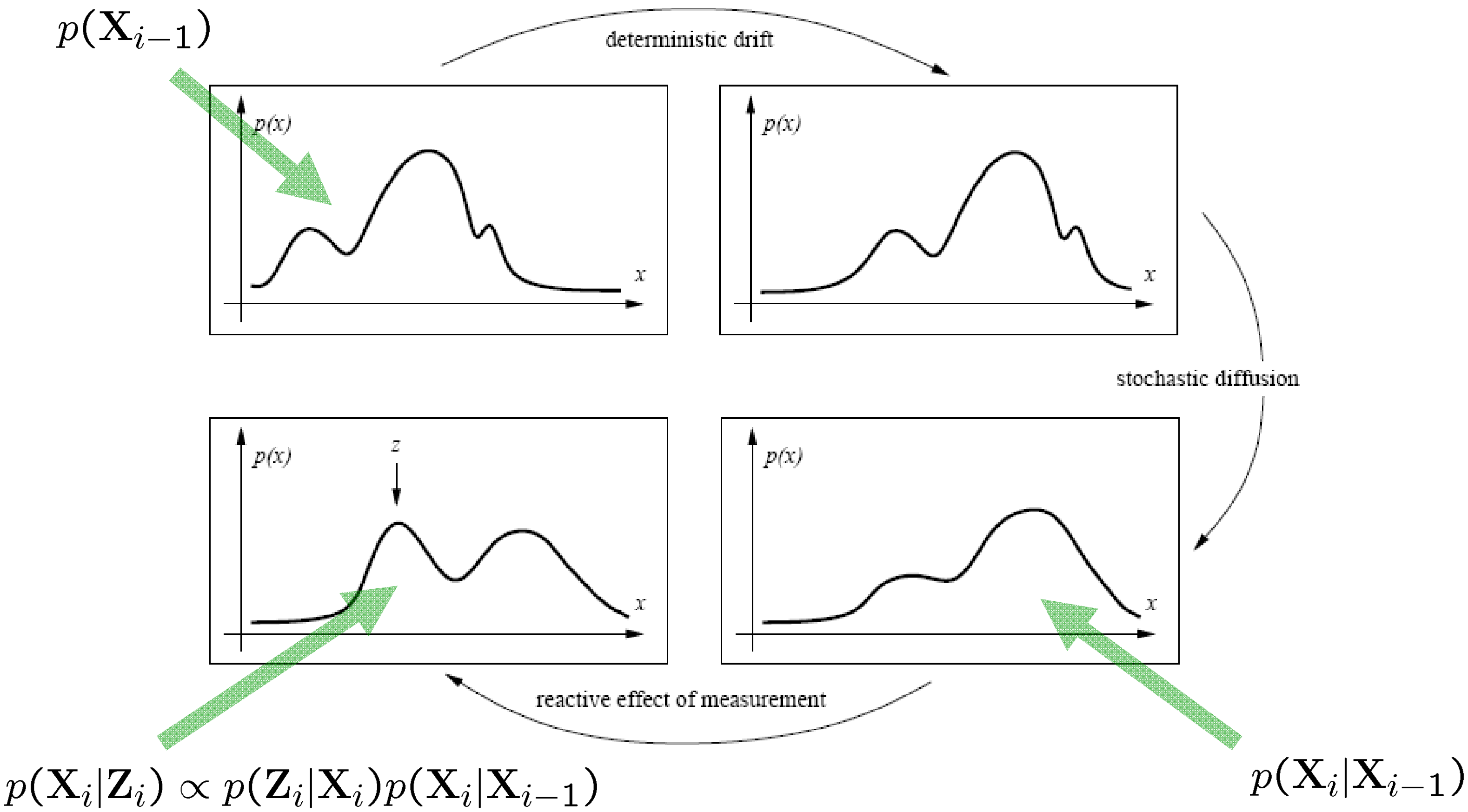
Kalman Filtering

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



Particle Filtering

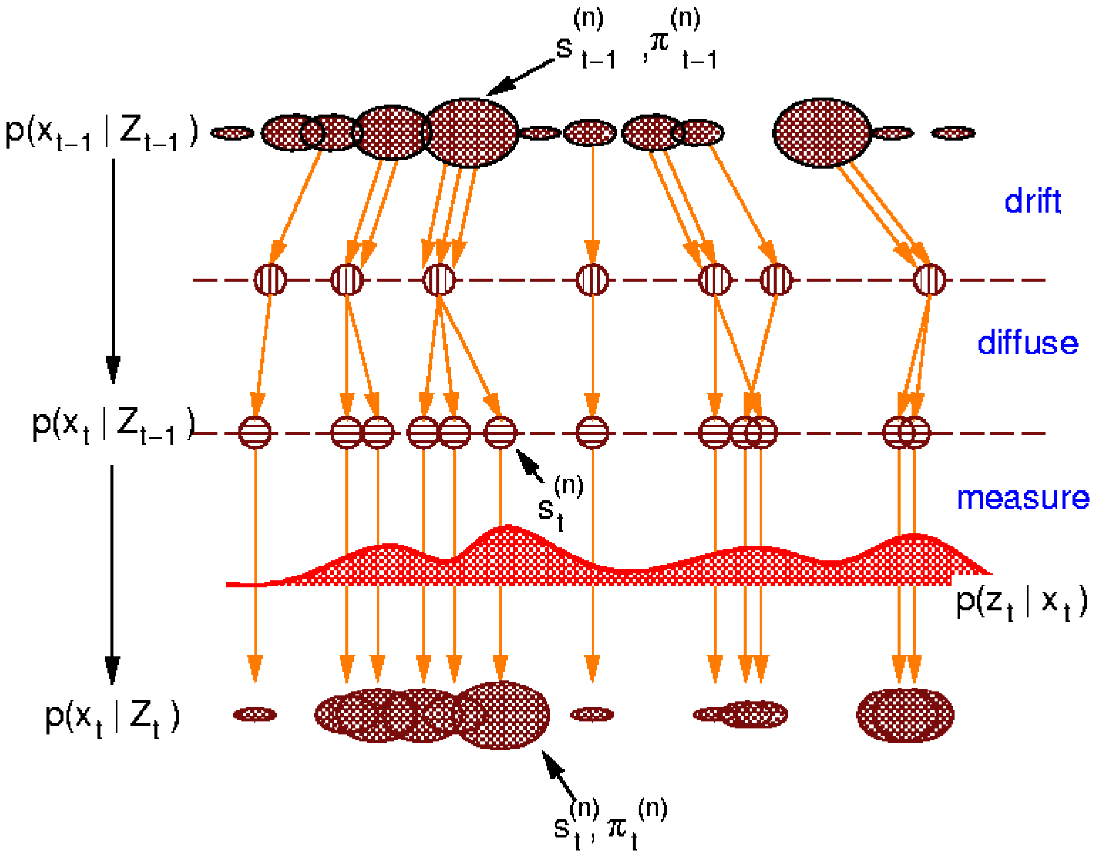
In reality probability densities are almost always *multi-modal*



Particle Filtering

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

➡ Approximate distributions with weighted particles

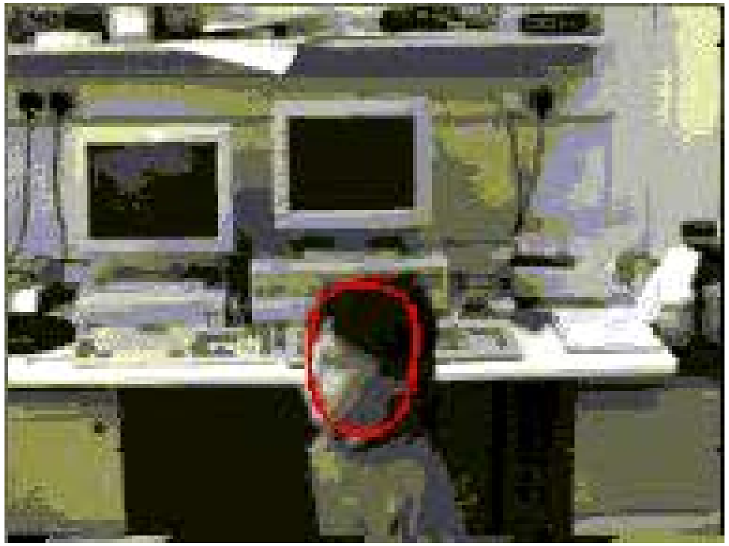


Particle Filtering

Tracking examples:

$\mathbf{x}$ describes leave shape

$\mathbf{x}$ describes head shape



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

Learning dynamic prior

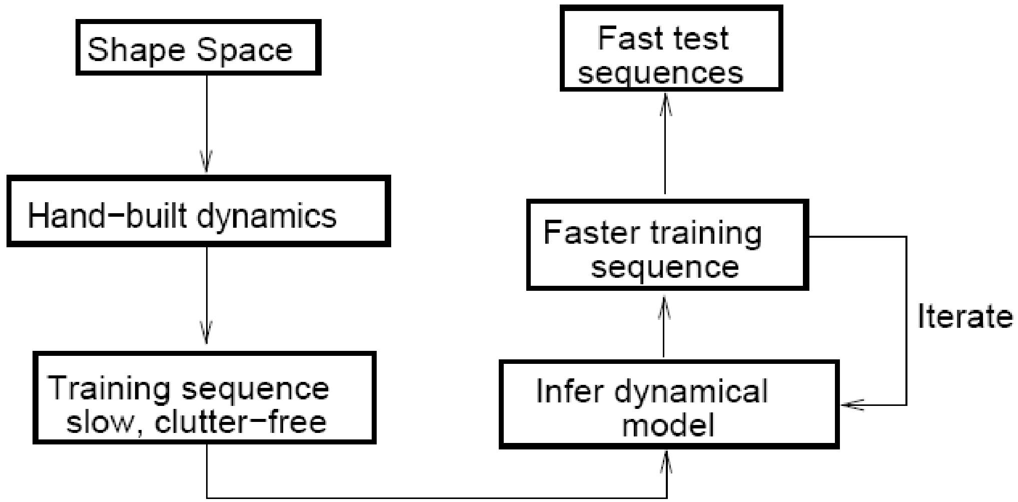
- Dynamic model: 2nd order Auto-Regressive Process

State $\mathcal{X}_k = \begin{pmatrix} \mathbf{X}_{k-1} \\ \mathbf{X}_k \end{pmatrix}$

Update rule: $\mathcal{X}_k - \overline{\mathcal{X}} = A(\mathcal{X}_{k-1} - \overline{\mathcal{X}}) + B\mathbf{w}_k$

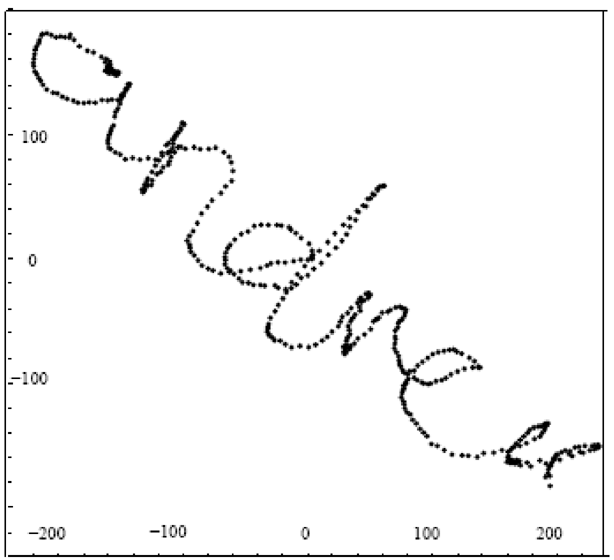
Model parameters: $A = \begin{pmatrix} 0 & I \\ A_2 & A_1 \end{pmatrix}, \quad \overline{\mathcal{X}} = \begin{pmatrix} \overline{\mathbf{X}} \\ \overline{\mathbf{X}} \end{pmatrix}$ and $B = \begin{pmatrix} 0 \\ B_0 \end{pmatrix}$

Learning scheme:

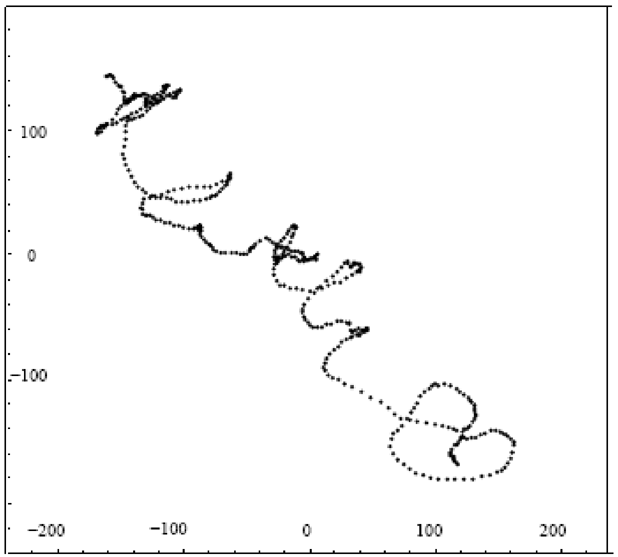


Learning dynamic prior

Learning point sequence



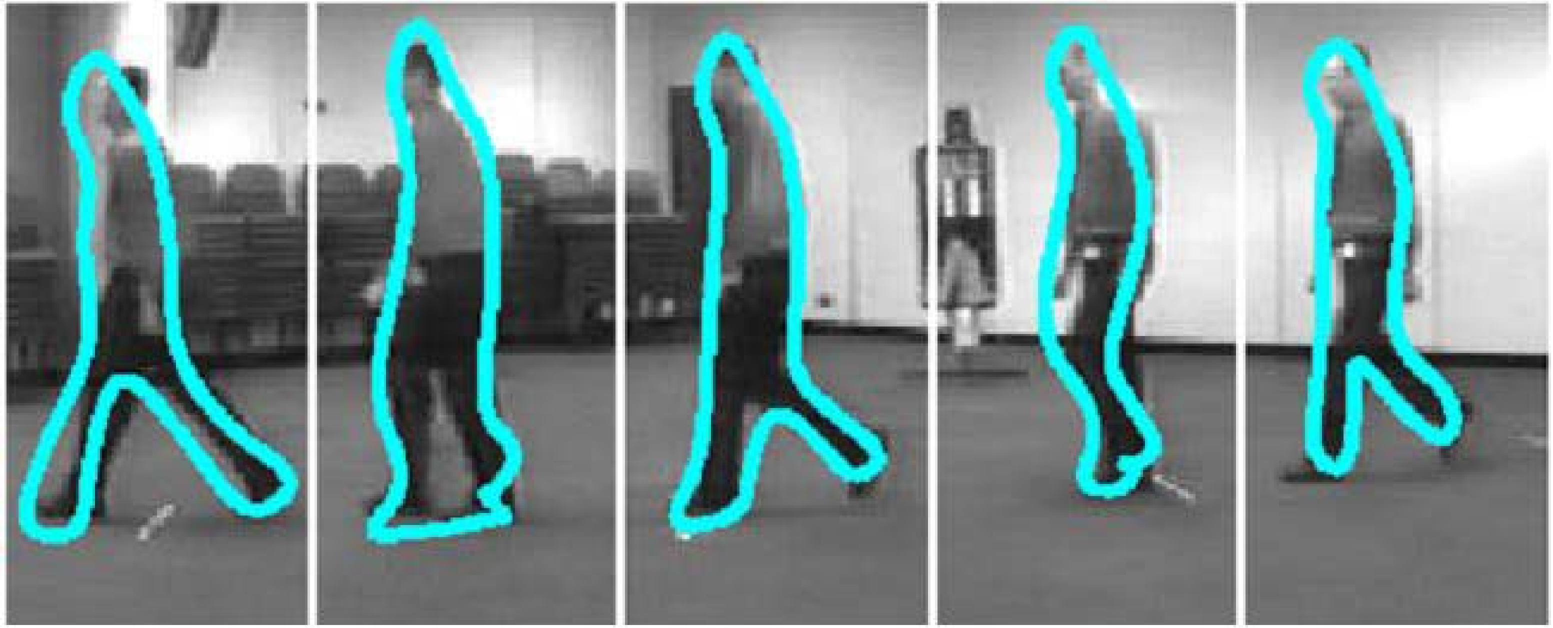
Random simulation of the
learned dynamical model



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 gate 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 $y_k = 1, 2, ..., n$

Transition probability matrix

$$P(y_k = j | y_{k-1} = i) = T_{i,j},$$

or more generally $P(y_k = j | y_{k-1} = i, \mathcal{X}_{k-1}) = T_{i,j}(\mathcal{X}_{k-1})$

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

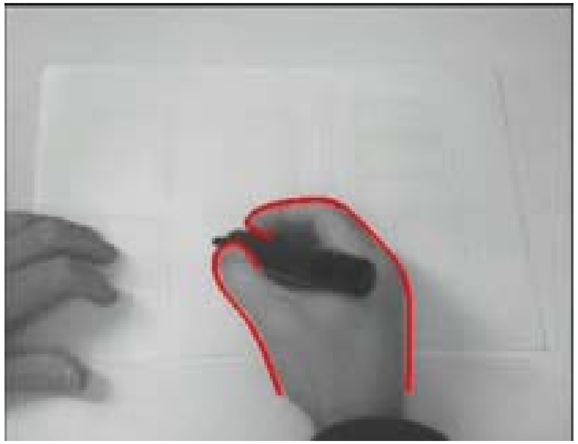
Dynamics with discrete states

Example: Drawing

Transition probability matrix

$$
\begin{array}{cccc}
 & \text{line} & \text{idle} & \text{scribbling} & \\
T = & \left(\begin{array}{ccc} 0.9800 & 0.0015 & 0.0185 \\ 0.0850 & 0.9000 & 0.0150 \\ 0.0050 & 0.0150 & 0.9800 \end{array}\right) & & & \begin{array}{l} \text{line} \\ \text{idle} \\ \text{scribbling} \end{array}
\end{array}
$$

Result: simultaneously improved tracking and gesture recognition



line drawing

scribbling

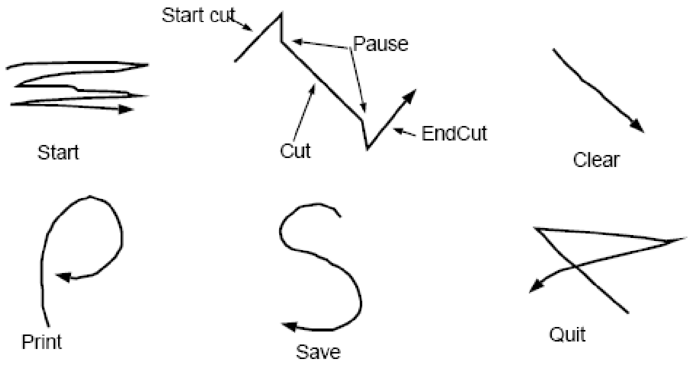
idle

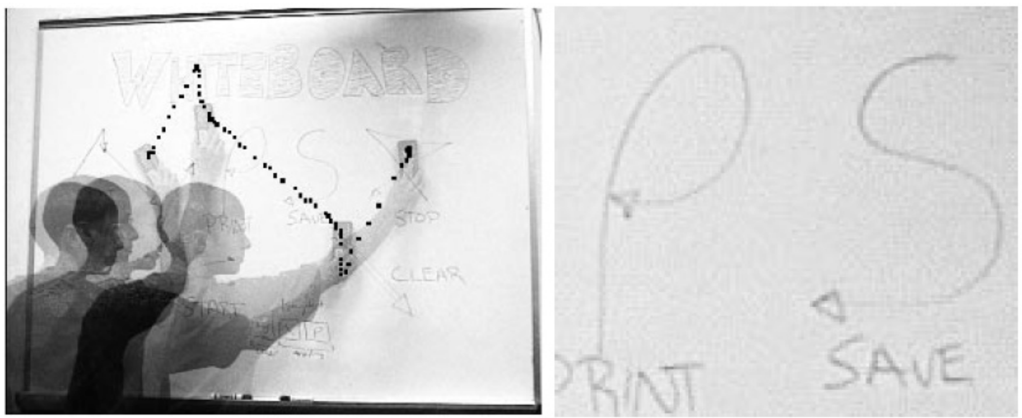
A mixed-state Condensation tracker with automatic model-switching

M. Isard and A. Blake, **ICCV** 1998

Dynamics with discrete states

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





[M.J. Black and A.D. Jepson, ECCV 1998]

Motion priors & Trackimg: Summary

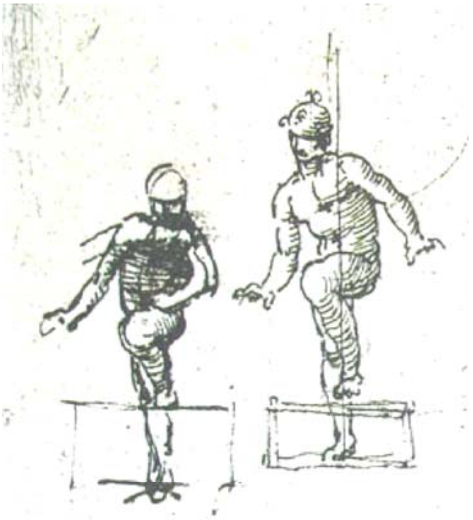
Pros:

- **+** 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

Human Pose Estimation

Pictorial structures
Learning models from image data
Recent advances
Datasets and challenges

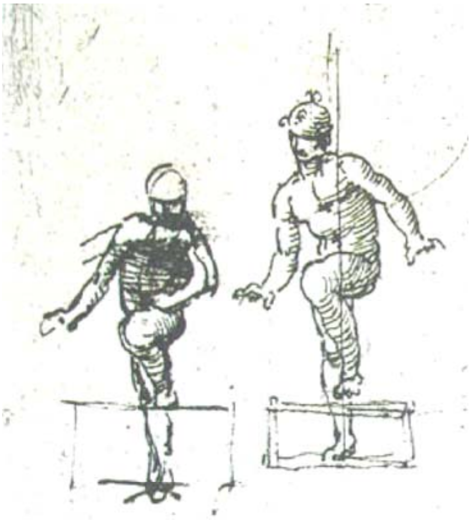
Appearance-based methods

Motion history images
Active shape models
Motion priors

Motion-based methods

Generic and parametric Optical Flow
Motion templates

Class overview



Motivation

Historic review
Modern applications

Human Pose Estimation

Pictorial structures
Learning models from image data
Recent advances
Datasets and challenges

Appearance-based methods

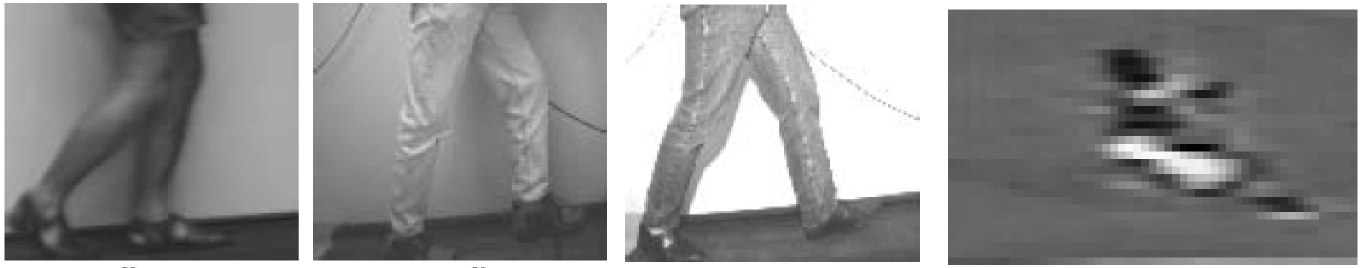
Motion history images
Active shape models
Motion priors

Motion-based methods

Generic and parametric Optical Flow
Motion templates

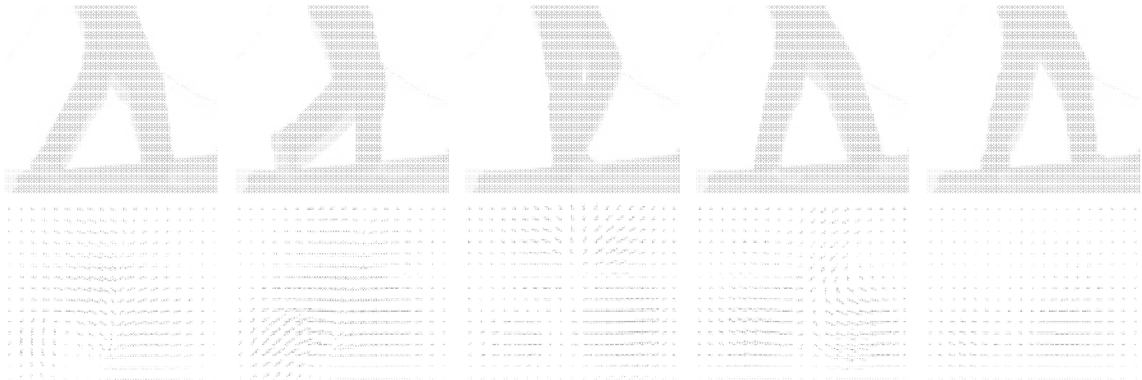
Shape and Appearance vs. Motion

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



[Efros et al. 2003]

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



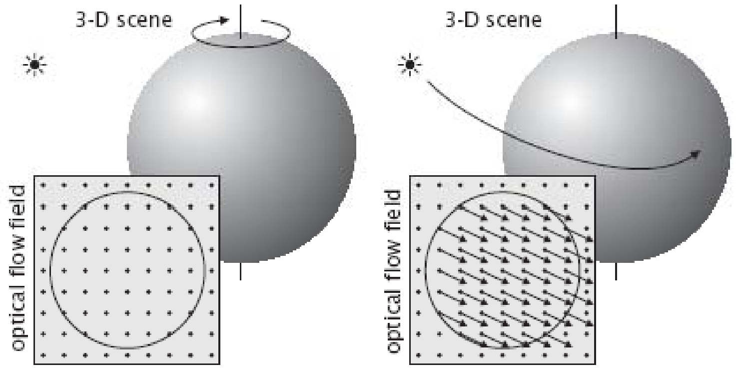
Motion estimation: Optical Flow

- 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 Change 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



Generic Optical Flow

- Brightness Change Constraint Equation (BCCE)

$$(\nabla I)^\top \mathbf{v} + I_t = 0$$

$\mathbf{v} = (v_x, v_y)^\top$ 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]

$$< \nabla I (\nabla I)^\top > \mathbf{v} = - < \nabla I I_t >$$

$$\begin{pmatrix} < I_x^2 > & < I_x I_y > \\ < I_x I_y > & < I_y^2 > \end{pmatrix} \mathbf{v} = - \begin{pmatrix} < I_x I_t > \\ < I_y I_t > \end{pmatrix}$$

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

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

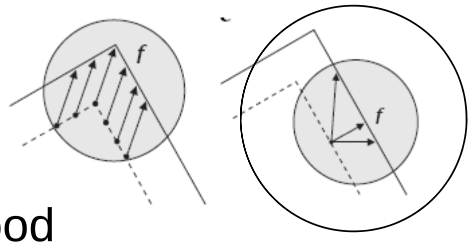
Generic Optical Flow

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

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

- Solutions:

  (2) Insufficient gradient variation known as *aperture problem*

  ➡ Increase integration neighborhood

  (3) Non-constant motion in $< \cdot >$

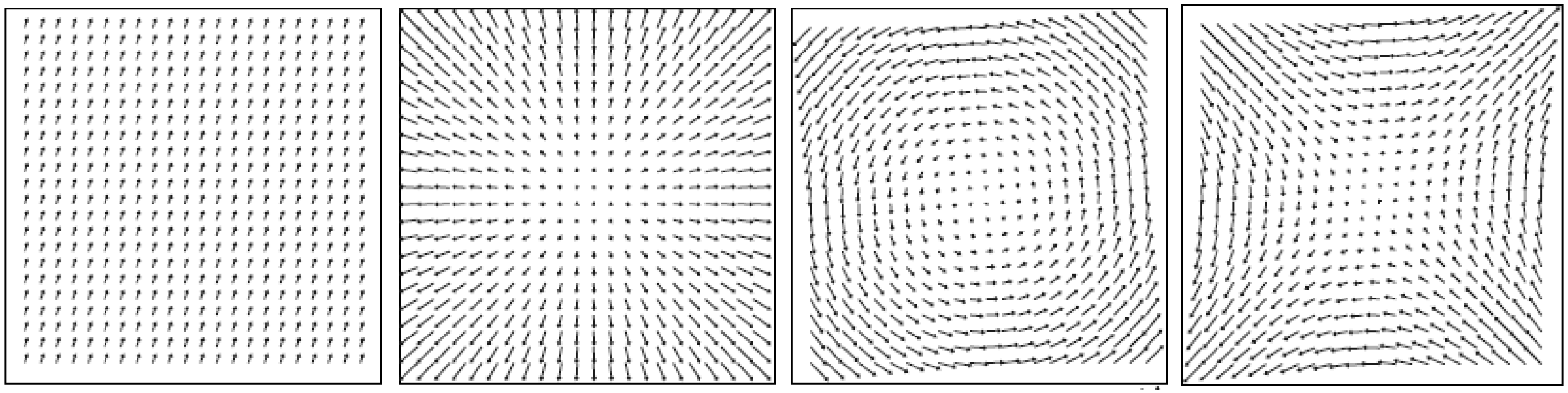
  ➡ Use more sophisticated motion model

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 $(x, y)^\top$ inside the neighborhood

Examples of Affine motion models for different parameters:



- Can be formulated as Least Squares approach to estimate $\mathbf{v}$ as before!

Parameterized Optical Flow

- Another extension of the constant motion model is to compute PCA basis flow fields from training examples

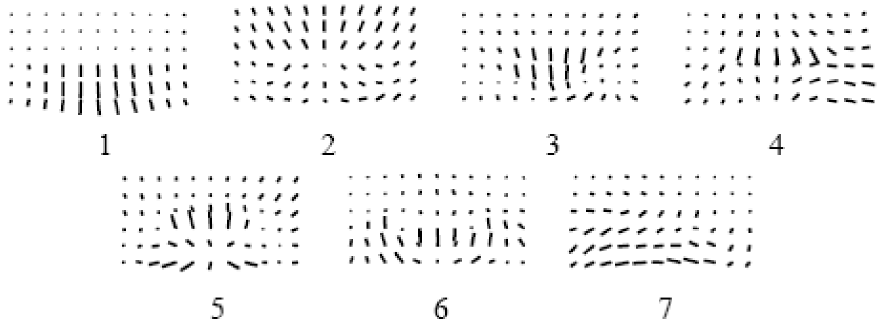
1. Compute standard Optical Flow for many examples
2. Put velocity components into one vector

$$\mathbf{w} = (v_x^1, v_y^1, v_x^2, v_y^2, ..., v_x^n, v_y^n)^\top$$

3. Do PCA on $\mathbf{w}$ and obtain most informative PCA flow basis vectors

Training samples

PCA flow bases



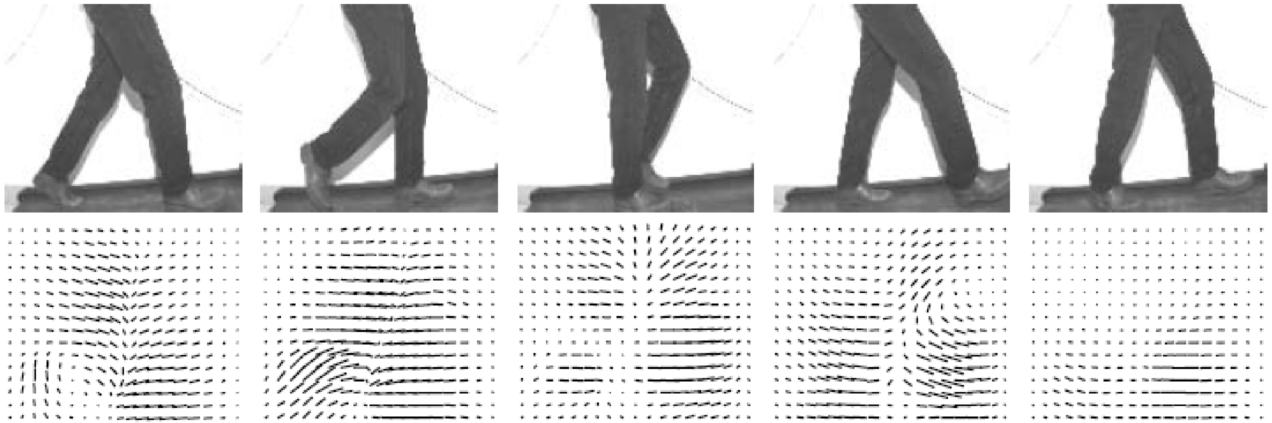
Learning Parameterized Models of Image Motion
M.J. Black, Y. Yacoob, A.D. Jepson and D.J. Fleet, **CVPR 1997**

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

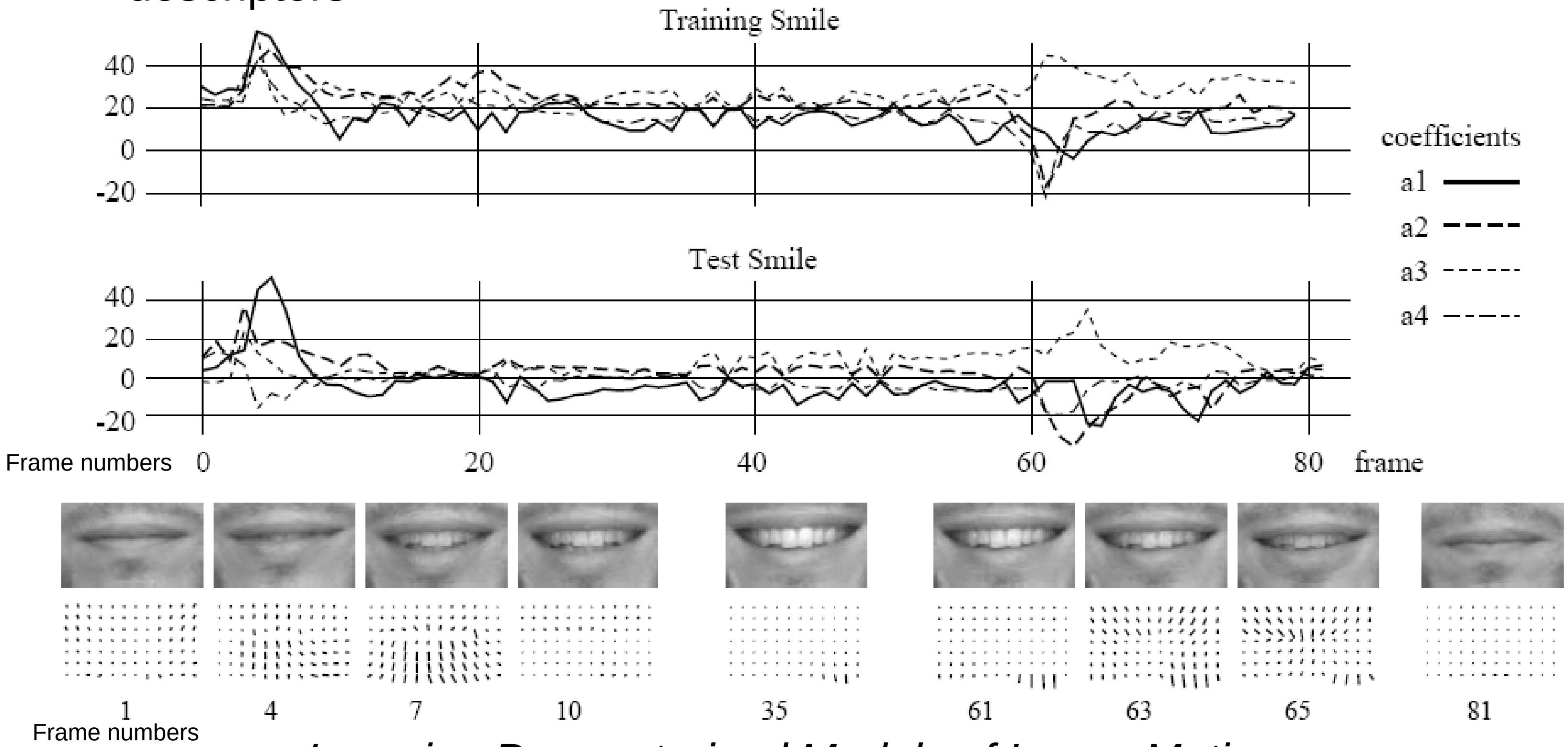
Direct flow recovery:



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

Parameterized Optical Flow

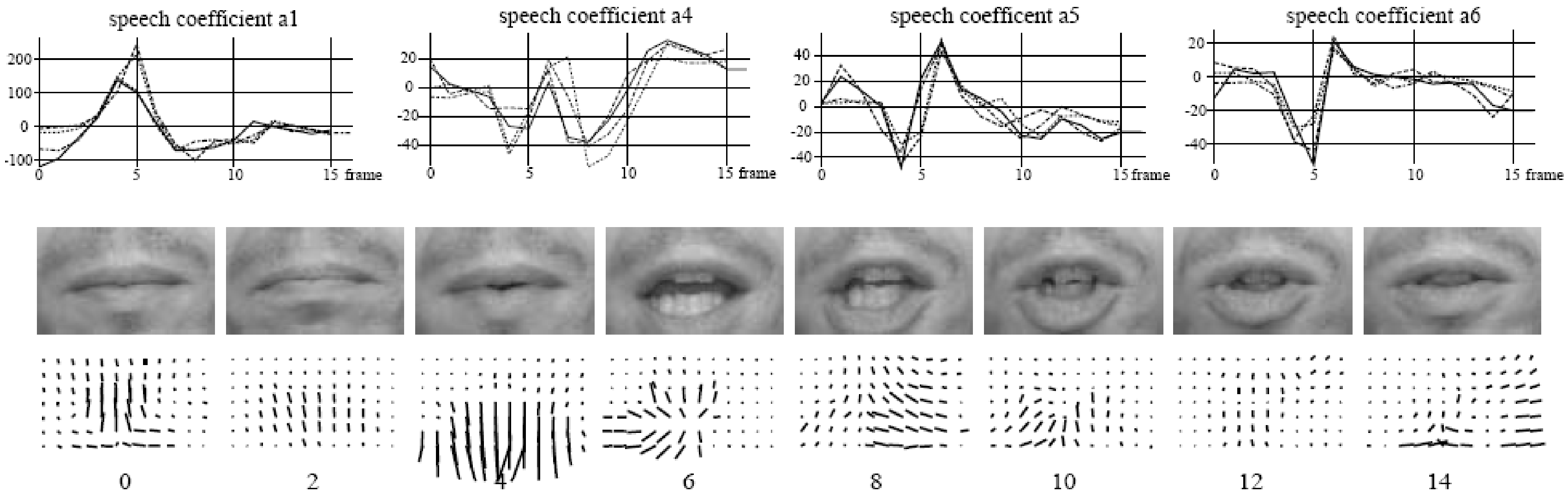
- Estimated coefficients of PCA flow bases can be used as action descriptors



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



Frame numbers

➡ Optical flow seems to be an interesting descriptor for motion/action recognition

Spatial Motion Descriptor

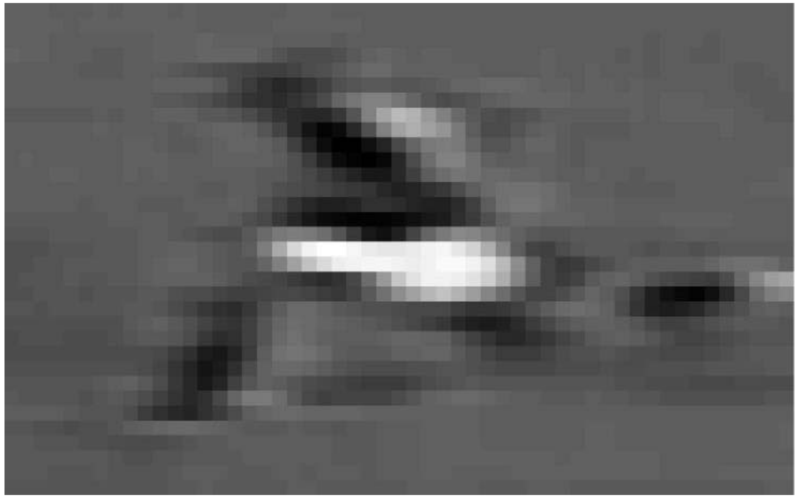
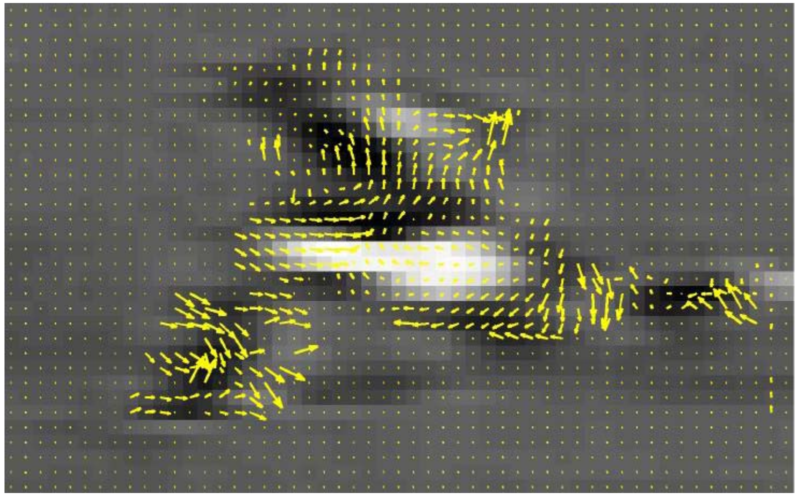
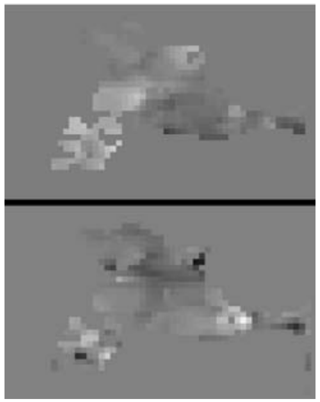


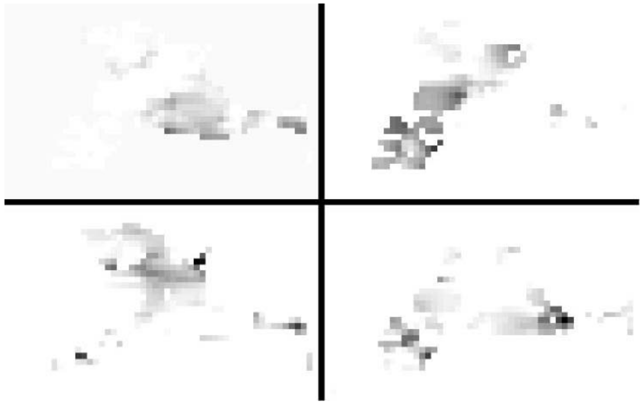
Image frame



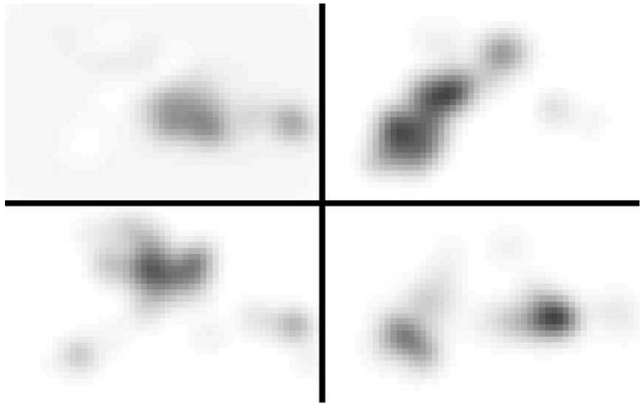
Optical flow $F_{x,y}$



F_x, F_y

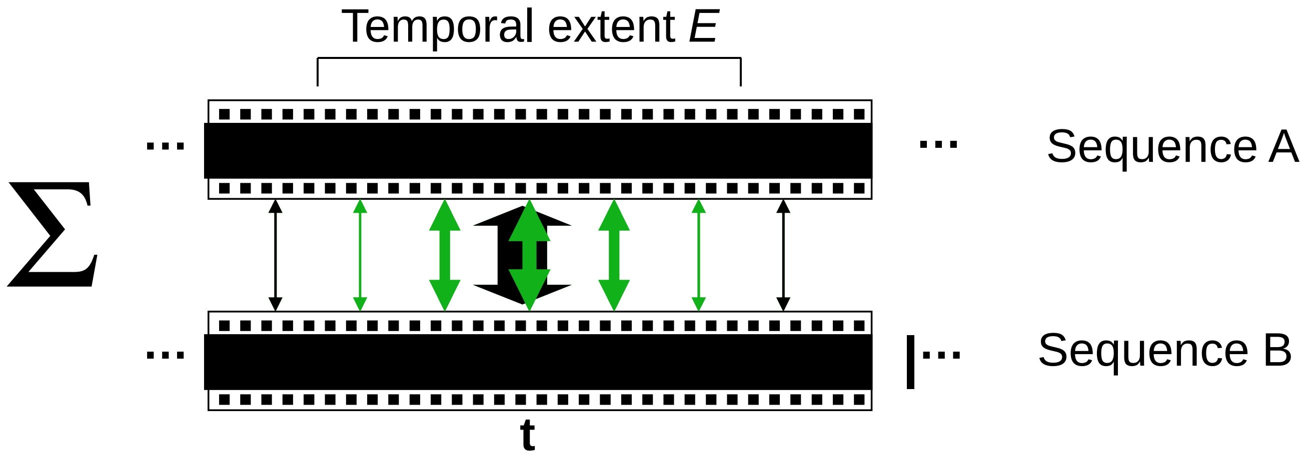


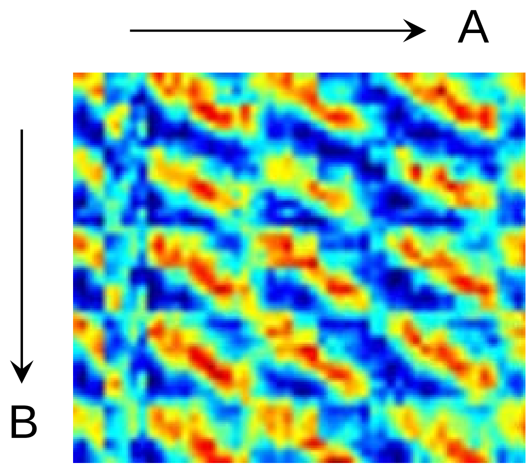
$F_x^-, F_x^+, F_y^-, F_y^+$



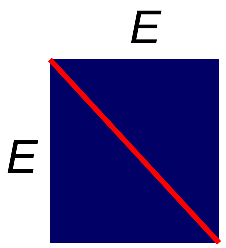
blurred $F_x^-, F_x^+, F_y^-, F_y^+$

Spatio-Temporal Motion Descriptor

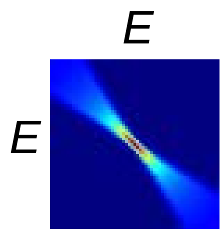




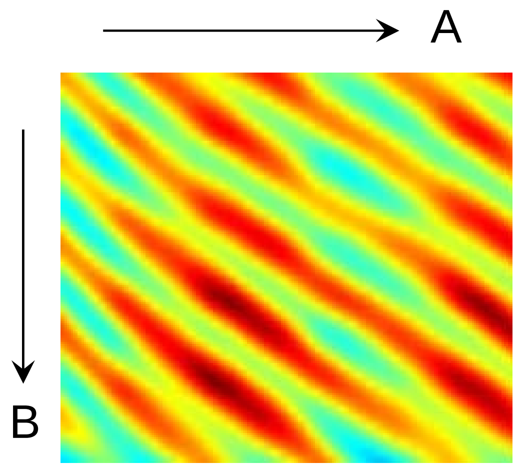
frame-to-frame
similarity matrix



I matrix



blurry I

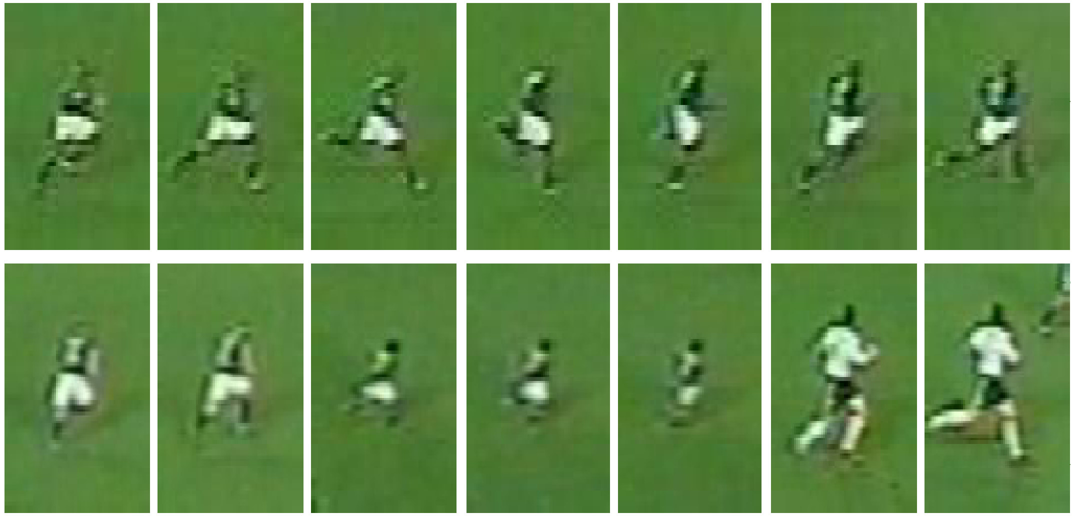


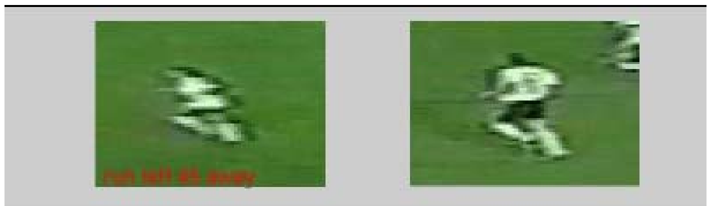
motion-to-motion
similarity matrix

Football Actions: matching

Input
Sequence

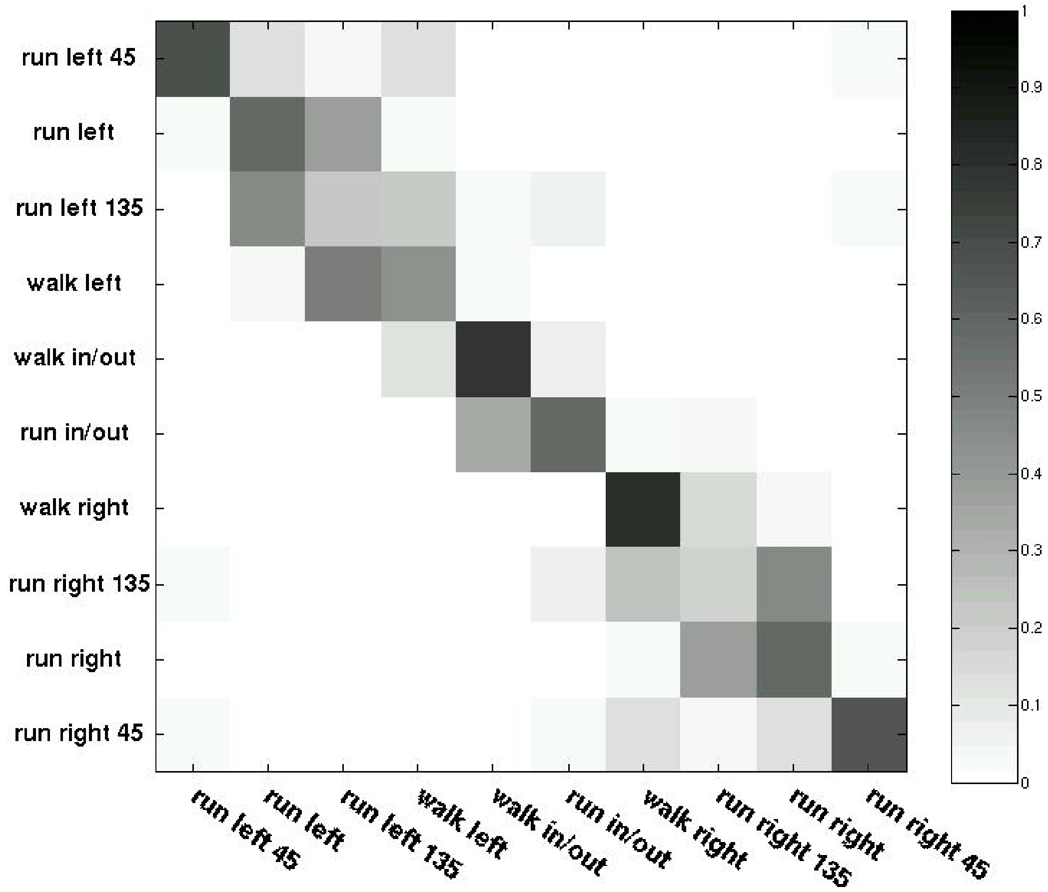
Matched
Frames





input matched

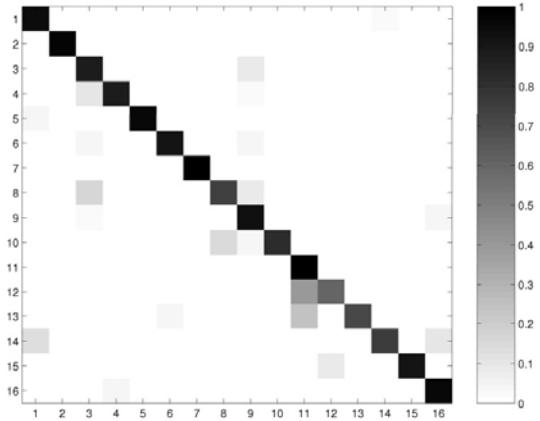
Football Actions: classification



10 actions; 4500 total frames; 13-frame motion descriptor

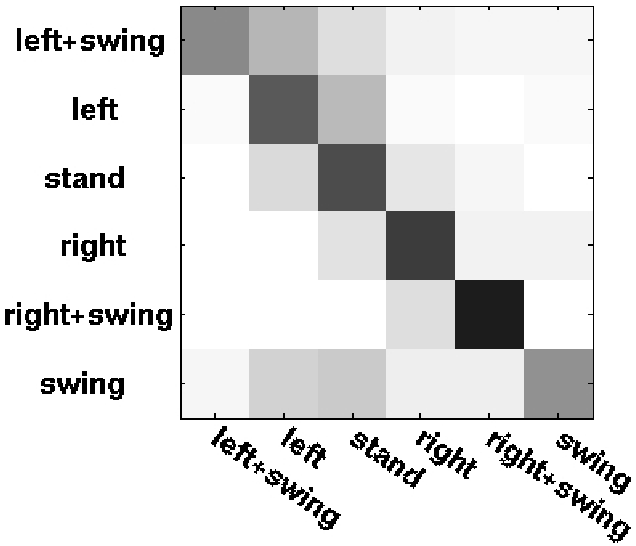
Classifying Ballet Actions

16 Actions; 24800 total frames; 51-frame motion descriptor. Men used to classify women and vice versa.



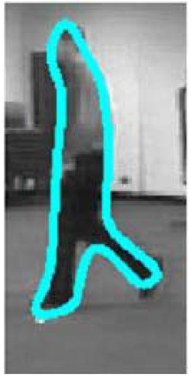
Classifying Tennis Actions

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

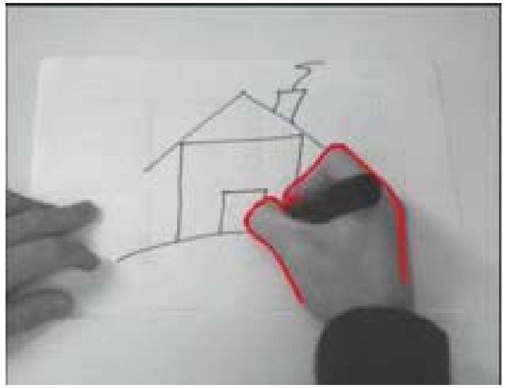


Where are we so far ?

Temporal templates:
+ simple, fast
- sensitive to segmentation errors

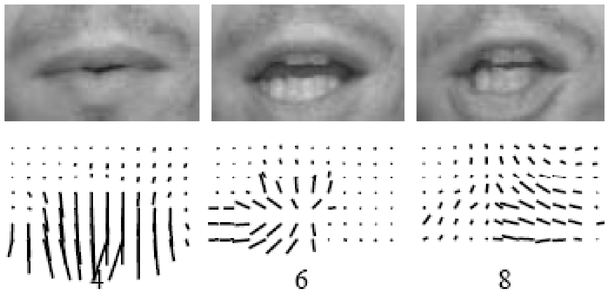


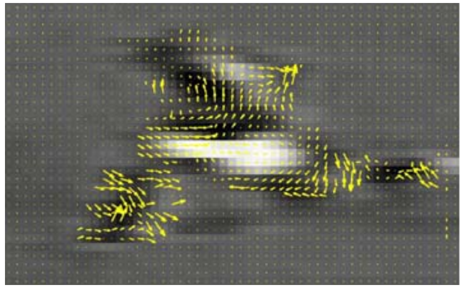
Active shape models:
+ shape regularization
- sensitive to initialization and tracking failures



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





Motivation

Goal:
Interpreting complex dynamic scenes

Common methods:

- Segmentation **?**
- Tracking **?**

Common problems:

- Complex & changing BG
- Changing appearance

⇒ *No global assumptions about the scene*

Space-time

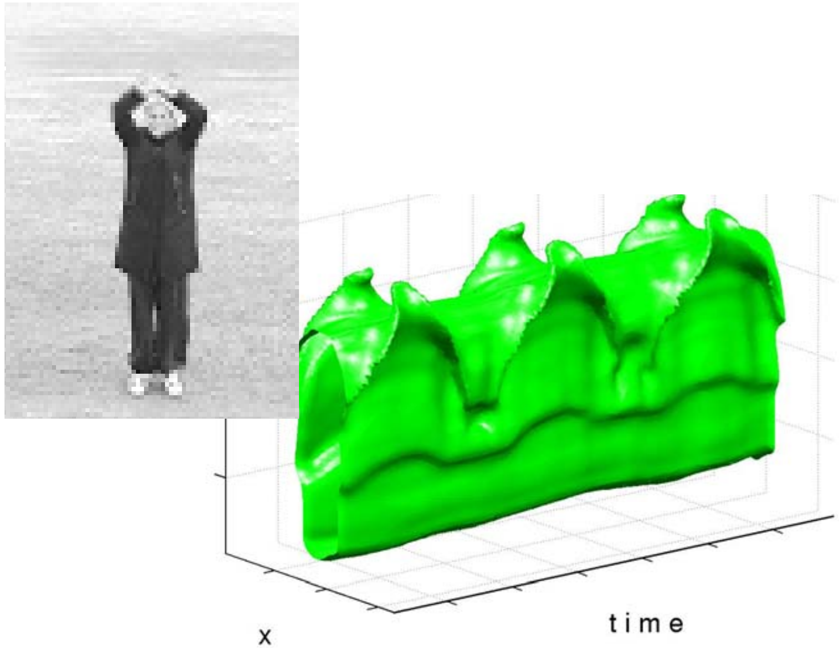
No global assumptions $\Rightarrow$

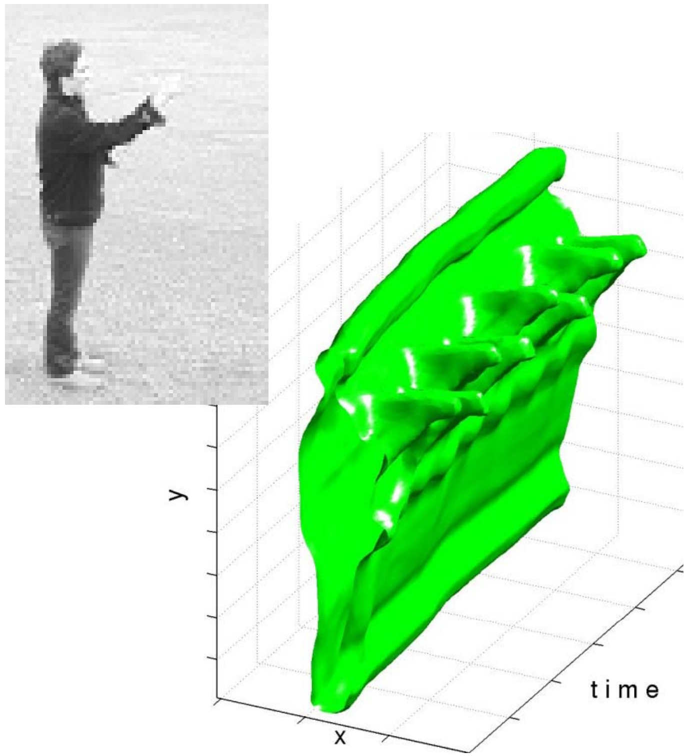
Consider local spatio-temporal neighborhoods

hand waving

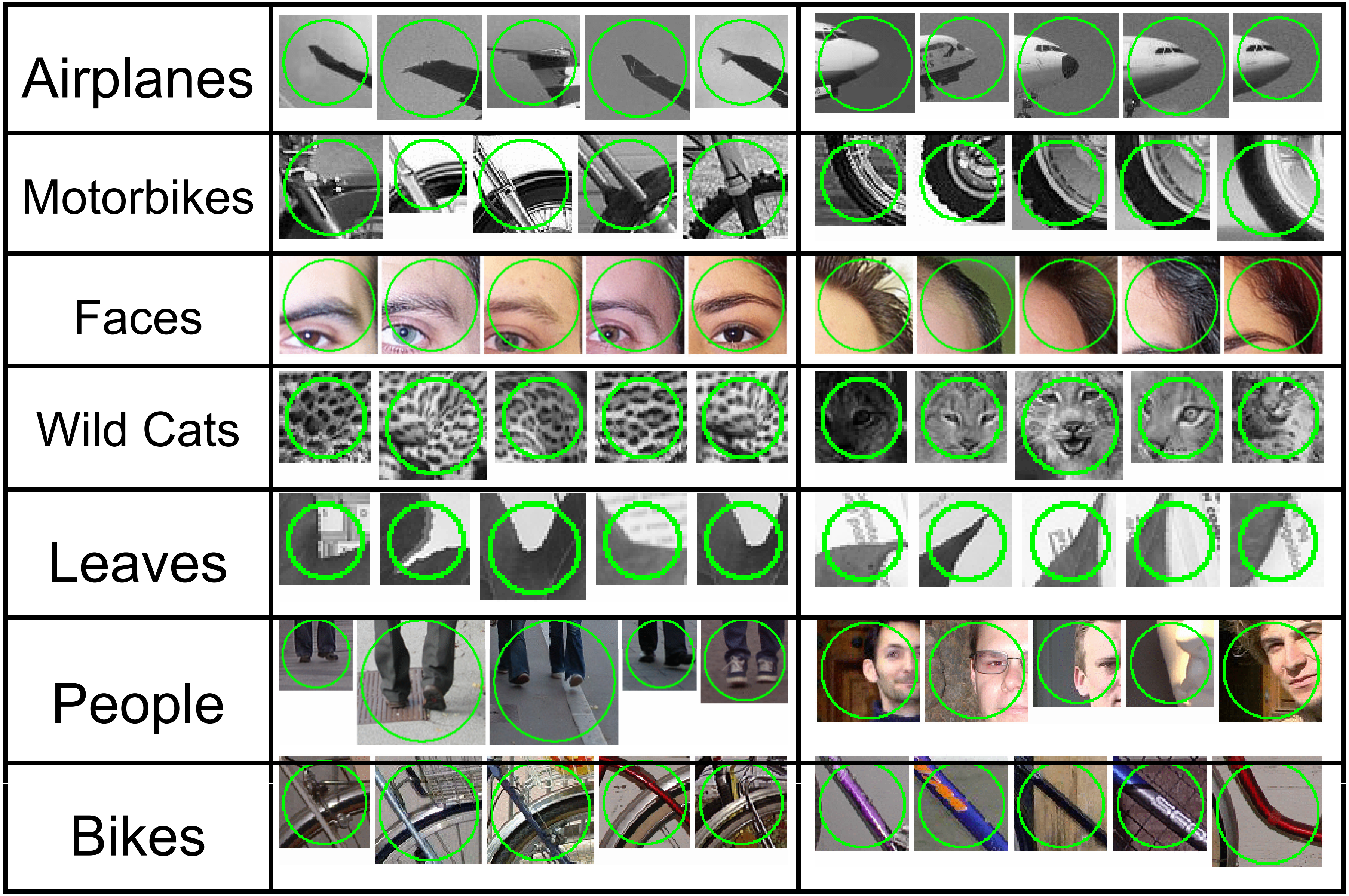
boxing

Actions == Space-time objects?

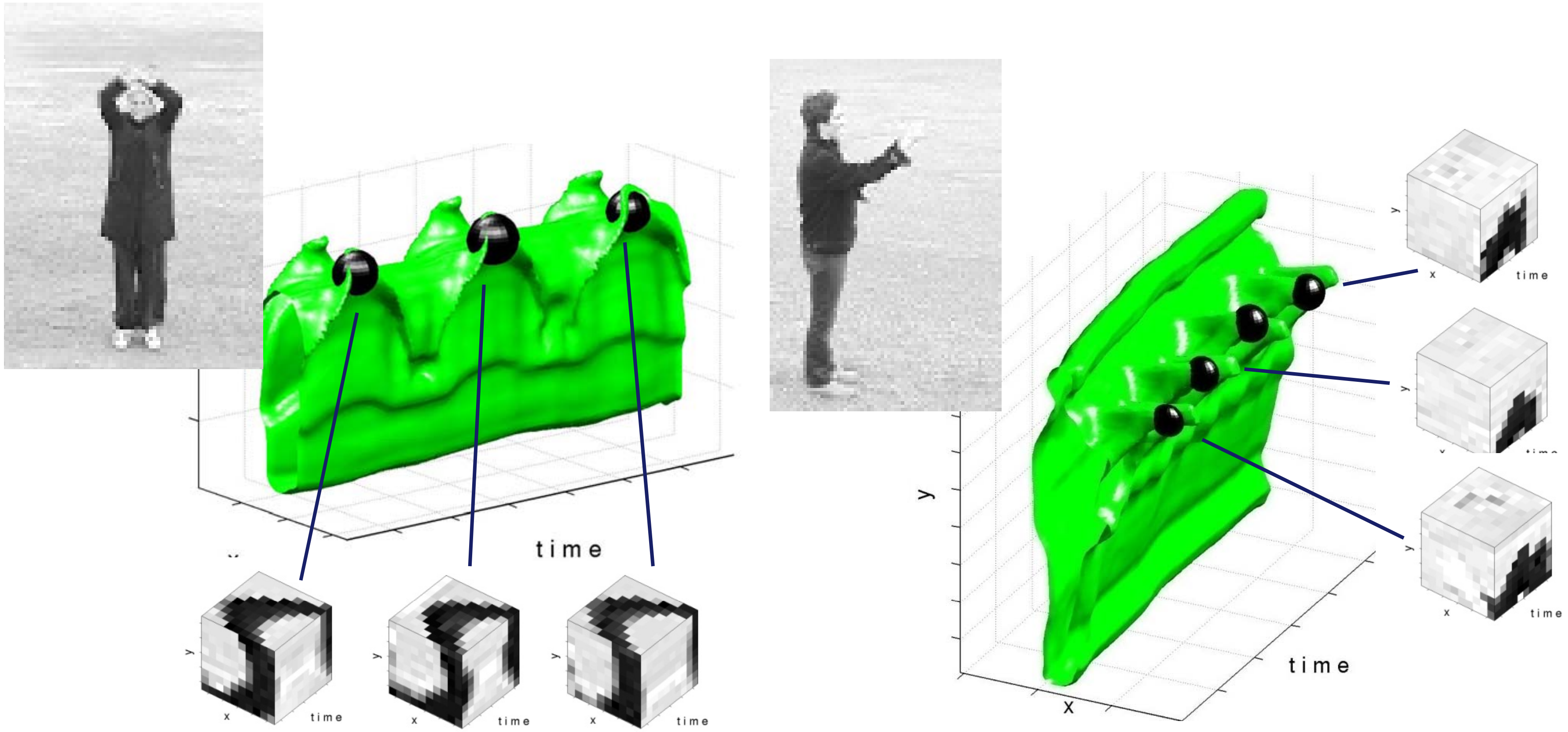




Local approach: Bag of Visual Words



Space-time local features



Space-Time Interest Points: Detection

What neighborhoods to consider?

Distinctive neighborhoods $\Rightarrow$ High image variation in space and time $\Rightarrow$ Look at the distribution of the gradient

Definitions:

$f : \mathbb{R}^2 \times \mathbb{R} \rightarrow \mathbb{R}$ Original image sequence

$g(x, y, t; \Sigma)$ Space-time Gaussian with covariance $\Sigma \in \mathrm{SPSD}(3)$

$L_\xi(\cdot; \Sigma) = f(\cdot) * g_\xi(\cdot; \Sigma)$ Gaussian derivative of f

$\nabla L = (L_x, L_y, L_t)^T$ Space-time gradient

$$\mu(\cdot; \Sigma) = \nabla L(\cdot; \Sigma)(\nabla L(\cdot; \Sigma))^T * g(\cdot; s\Sigma) = \begin{pmatrix} \mu_{xx} & \mu_{xy} & \mu_{xt} \\ \mu_{xy} & \mu_{yy} & \mu_{yt} \\ \mu_{xt} & \mu_{yt} & \mu_{tt} \end{pmatrix}$$

Second-moment matrix

Space-Time Interest Points: Detection

Properties of $\mu(\cdot;\,\Sigma)$

$\mu(\cdot;\,\Sigma)$ defines second order approximation for the local distribution of ∇L within neighborhood Σ

$\mathrm{rank}(\mu) = 1 \quad \Rightarrow$ 1D space-time variation of f e.g. moving bar

$\mathrm{rank}(\mu) = 2 \quad \Rightarrow$ 2D space-time variation of f e.g. moving ball

$\mathrm{rank}(\mu) = 3 \quad \Rightarrow$ 3D space-time variation of f e.g. jumping ball

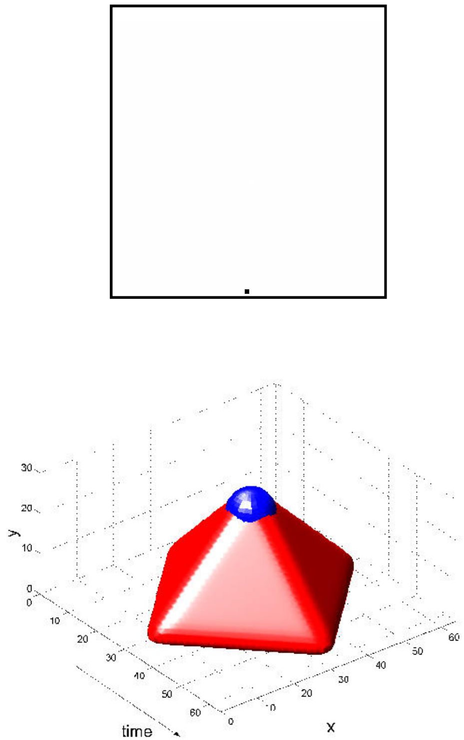
Large eigenvalues of μ can be detected by the local maxima of H over (x,y,t):

$$\begin{aligned} H(p;\,\Sigma) &= \det(\mu(p;\,\Sigma)) + k\,\mathrm{trace}^3(\mu(p;\,\Sigma)) \\ &= \lambda_1\lambda_2\lambda_3 - k(\lambda_1+\lambda_2+\lambda_3)^3 \end{aligned}$$

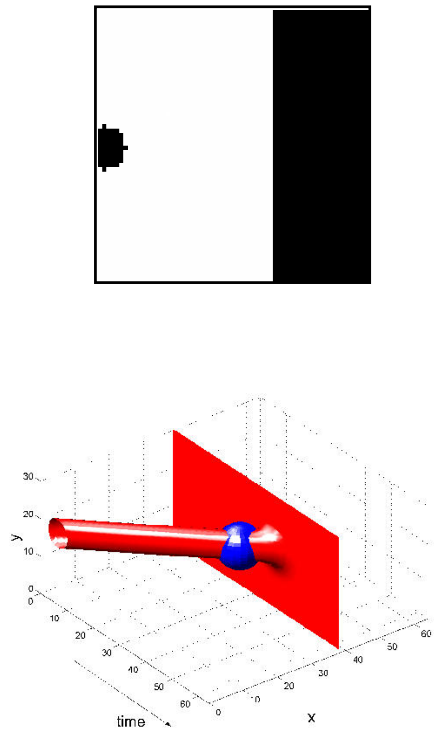
(similar to Harris operator [Harris and Stephens, 1988])

Space-Time interest points

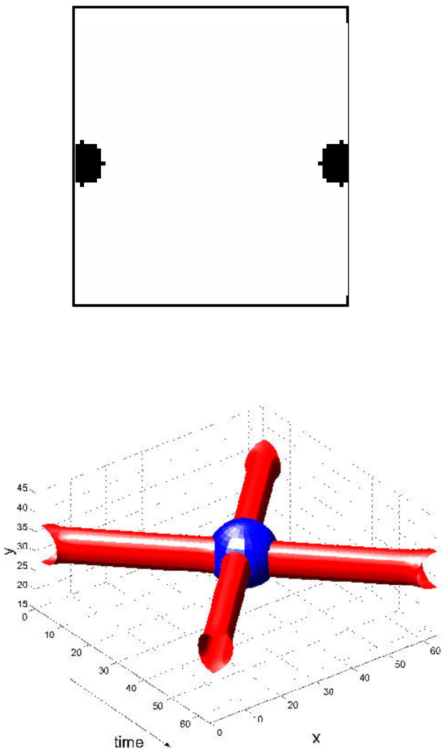
Velocity changes



appearance/ disappearance

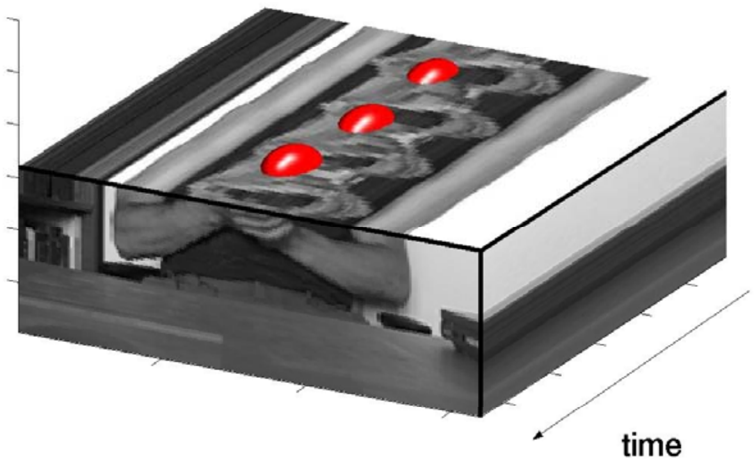


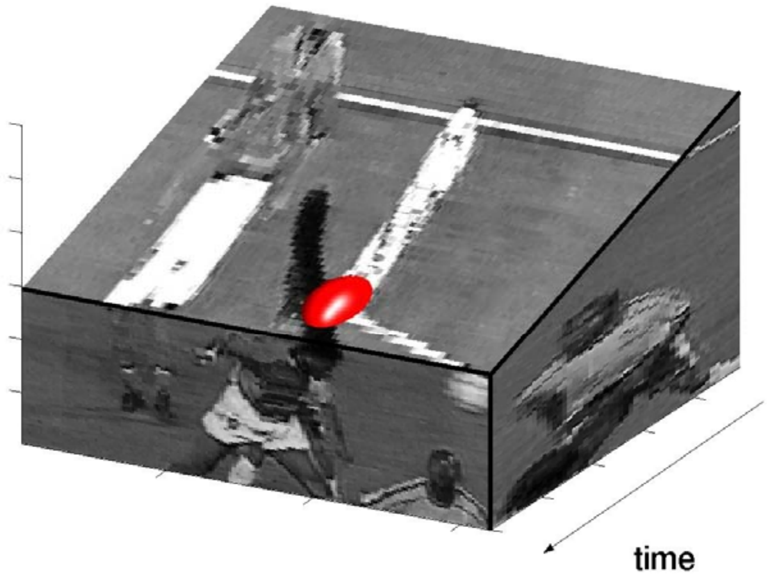
split/merge



Space-Time Interest Points: Examples

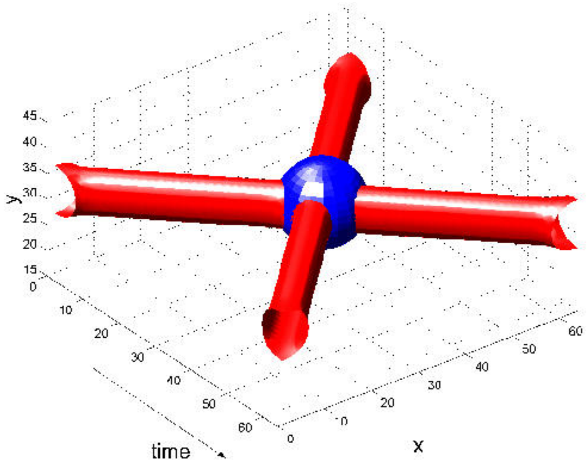
Motion event detection

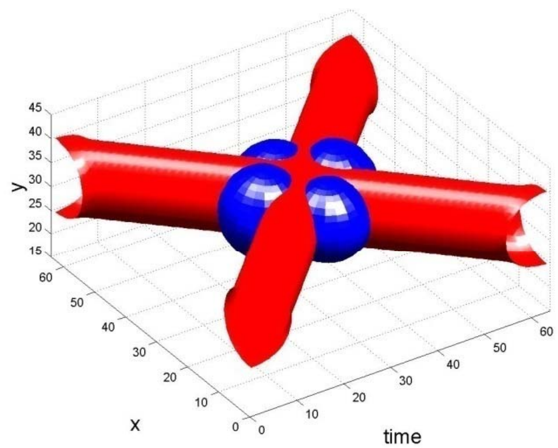




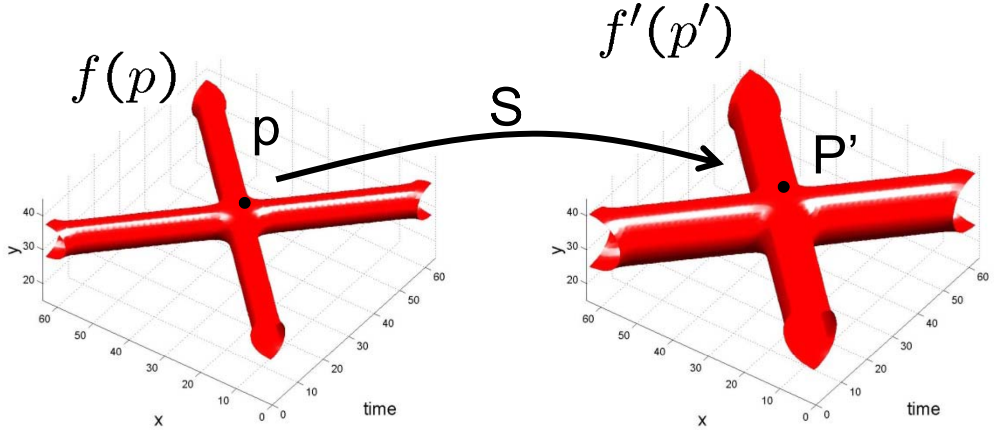
Spatio-temporal scale

What if the spatial and/or temporal resolution changes?





Spatio-temporal scale selection



point transformation

$$p = S^{-1}p', \quad S = \begin{pmatrix} s_\sigma & 0 & \\ 0 & s_\sigma & 0 \\ 0 & 0 & s_\tau \end{pmatrix}, \quad p = \begin{pmatrix} x \\ y \\ t \end{pmatrix}$$

covariance transformation

$$\Sigma = pp^T = S^{-2}\Sigma' = \begin{pmatrix} \sigma^2 & 0 & 0 \\ 0 & \sigma^2 & 0 \\ 0 & 0 & \tau^2 \end{pmatrix}$$

Spatio-temporal scale selection

point transformation

$$p = S^{-1}p', \quad S = \begin{pmatrix} s_\sigma & 0 & \\ 0 & s_\sigma & 0 \\ 0 & 0 & s_\tau \end{pmatrix}, \quad p = \begin{pmatrix} x \\ y \\ t \end{pmatrix}$$

covariance transformation

$$\Sigma = pp^T = S^{-2}\Sigma' = \begin{pmatrix} \sigma^2 & 0 & 0 \\ 0 & \sigma^2 & 0 \\ 0 & 0 & \tau^2 \end{pmatrix}$$

To be invariant to scale transformations we need to change filter covariance:

$$\begin{aligned} L_\xi(\cdot;\, \Sigma) &= f(\cdot) * g_\xi(\cdot;\, \Sigma) \\ &= f'(\cdot) * g_\xi(\cdot;\, \Sigma') \end{aligned}$$

Q: how to estimate the right filer size Σ ?
=>
Scale selection problem

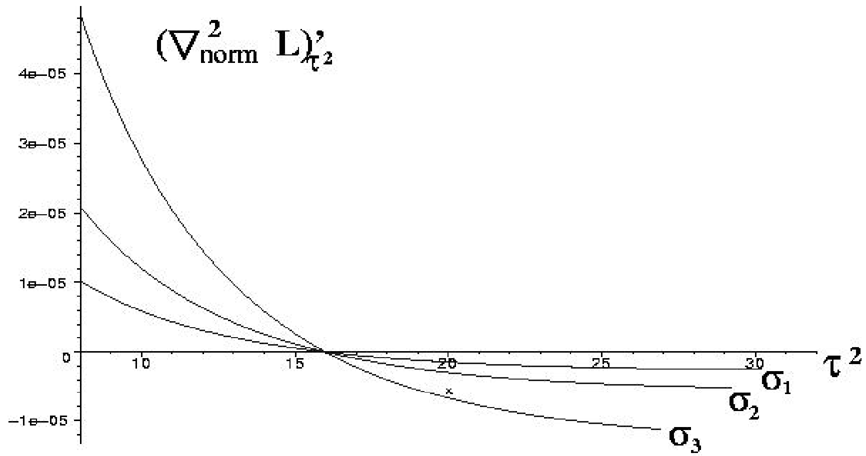
Spatio-temporal scale selection

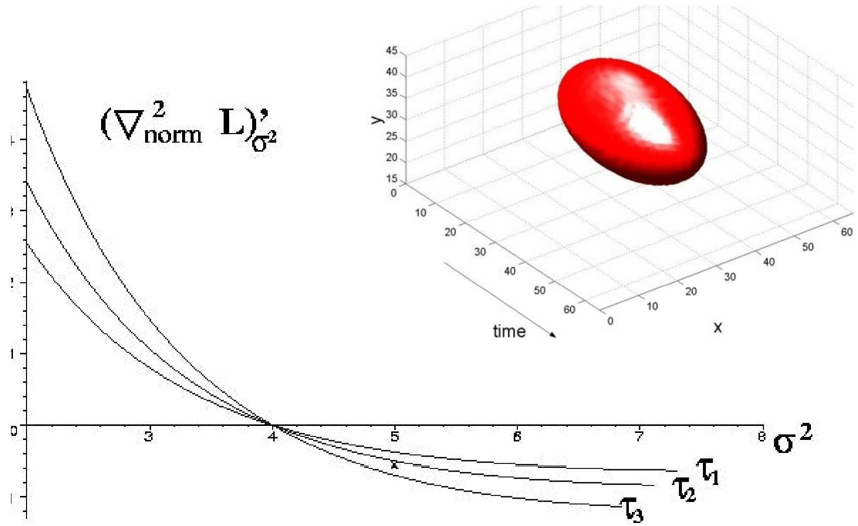
The normalized spatio-temporal Laplacian operator

$$\nabla^2_{norm} L = \sigma^2 \tau^{1/2} (L_{xx} + L_{yy}) + \sigma \tau^{3/2} L_{tt}$$

assumes scale-extrema values at the scale parameters of a spatio-temporal of a Gaussian blob

$\Rightarrow$ Estimate scale by maximizing $\quad (\nabla^2_{norm} L)^2 \quad \sigma, \tau$





(similar to scale selection is space [Lindeberg, 1998])

Space-Time interest points

H depends on μ and, hence, on Σ and scale transformation S

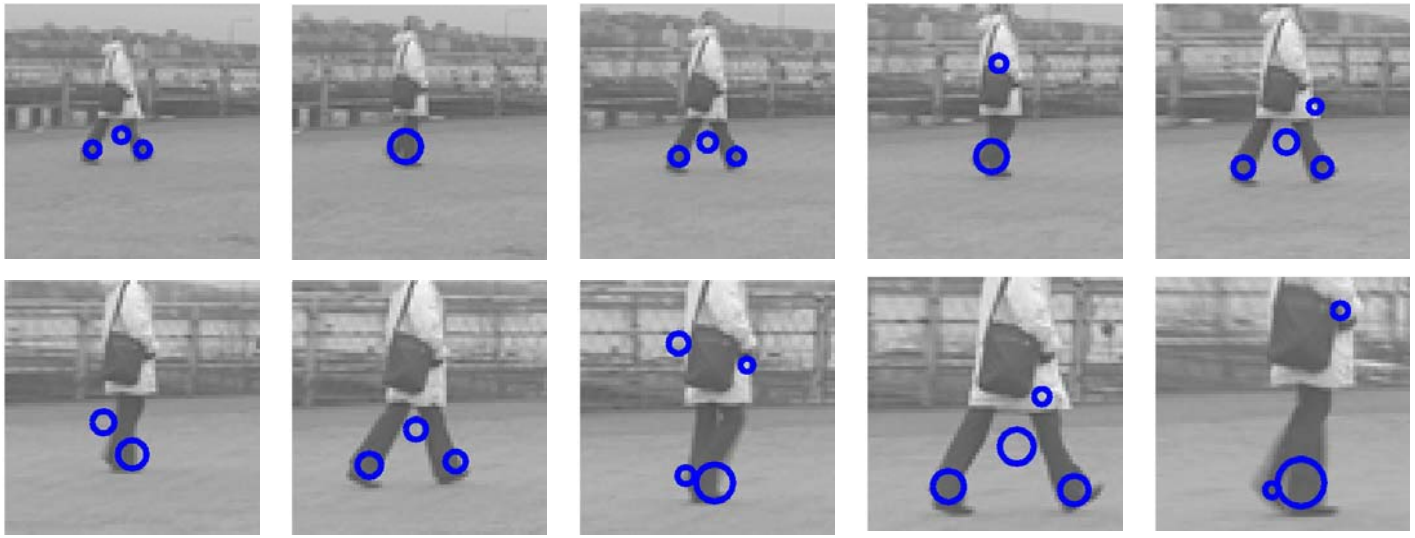
$\Rightarrow$ Adapt interest points by iteratively computing:

- Interest point detection $\quad H(p;\,\Sigma) = \det(\mu(p;\,\Sigma)) + k\,\mathrm{trace}^3(\mu(p;\,\Sigma))$ (*)
- Scale estimation $\quad (\sigma_0, \tau_0) = \mathrm{argmax}_{\sigma,\tau}(\nabla^2_{norm} L(p;\,\Sigma))^2$ (**)

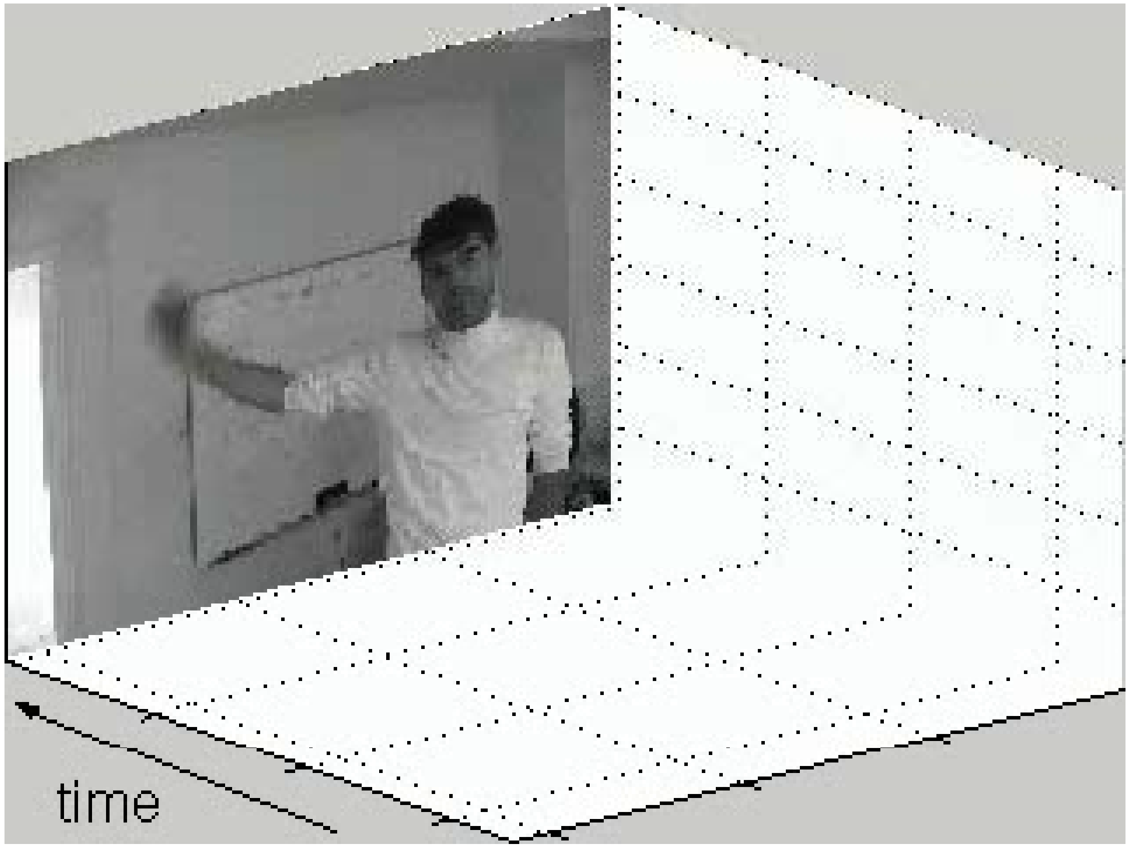
1. Fix Σ
2. For each detected interest point p_i (*)
3. Estimate scale $S(\sigma, \tau)$ (**)
4. Update covariance $\Sigma' = S^2$
5. Re-detect p_i using Σ'
6. Iterate 3-6 until convergence of σ, τ and p_i

Spatio-temporal scale selection

Stability to size changes,
e.g. camera zoom



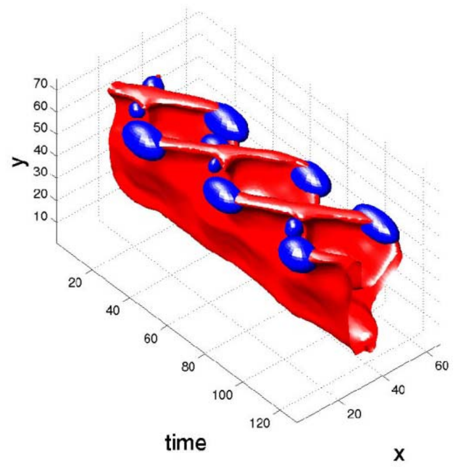
Spatio-temporal scale selection

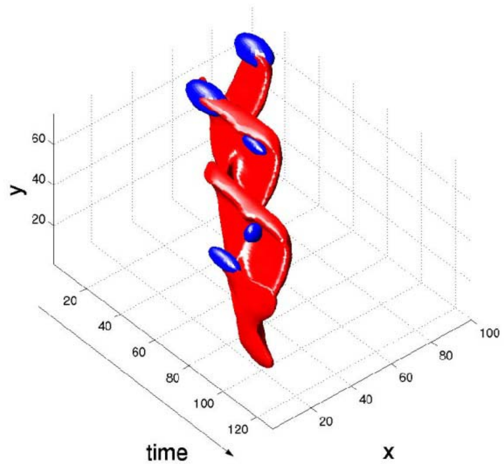


Selection of temporal scales captures the frequency of events

Relative camera motion

Space-time signal and its derivatives will change when if camera moves



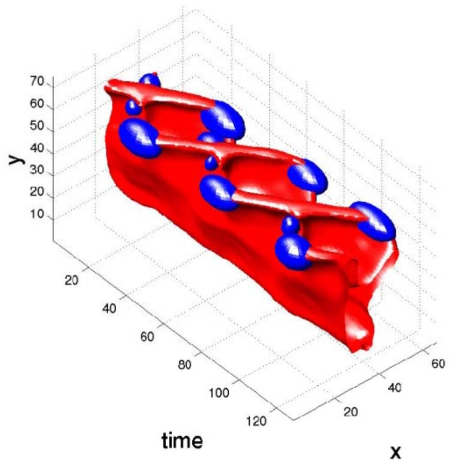


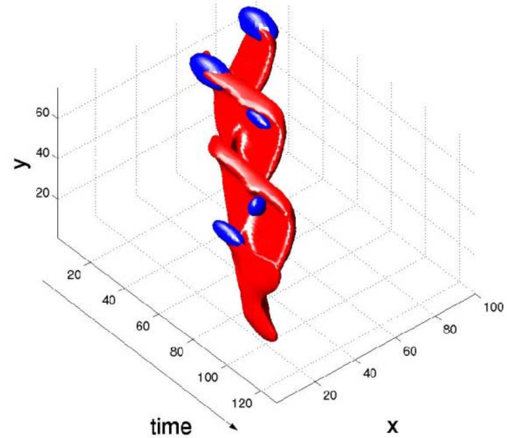
Adapted interest points

Stabilized camera

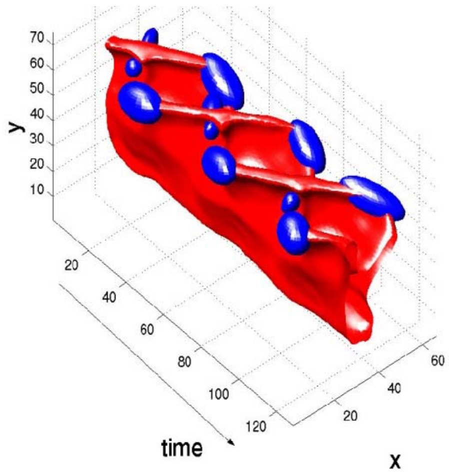
Stationary camera

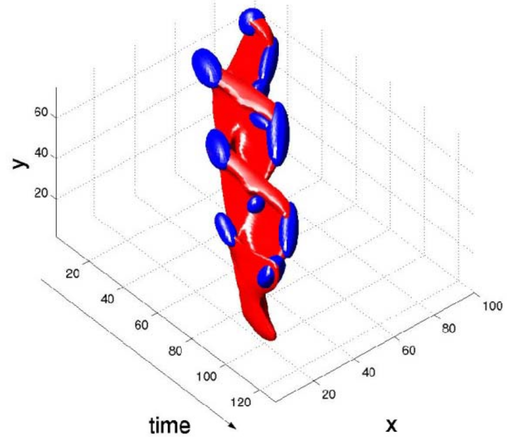
Interest points



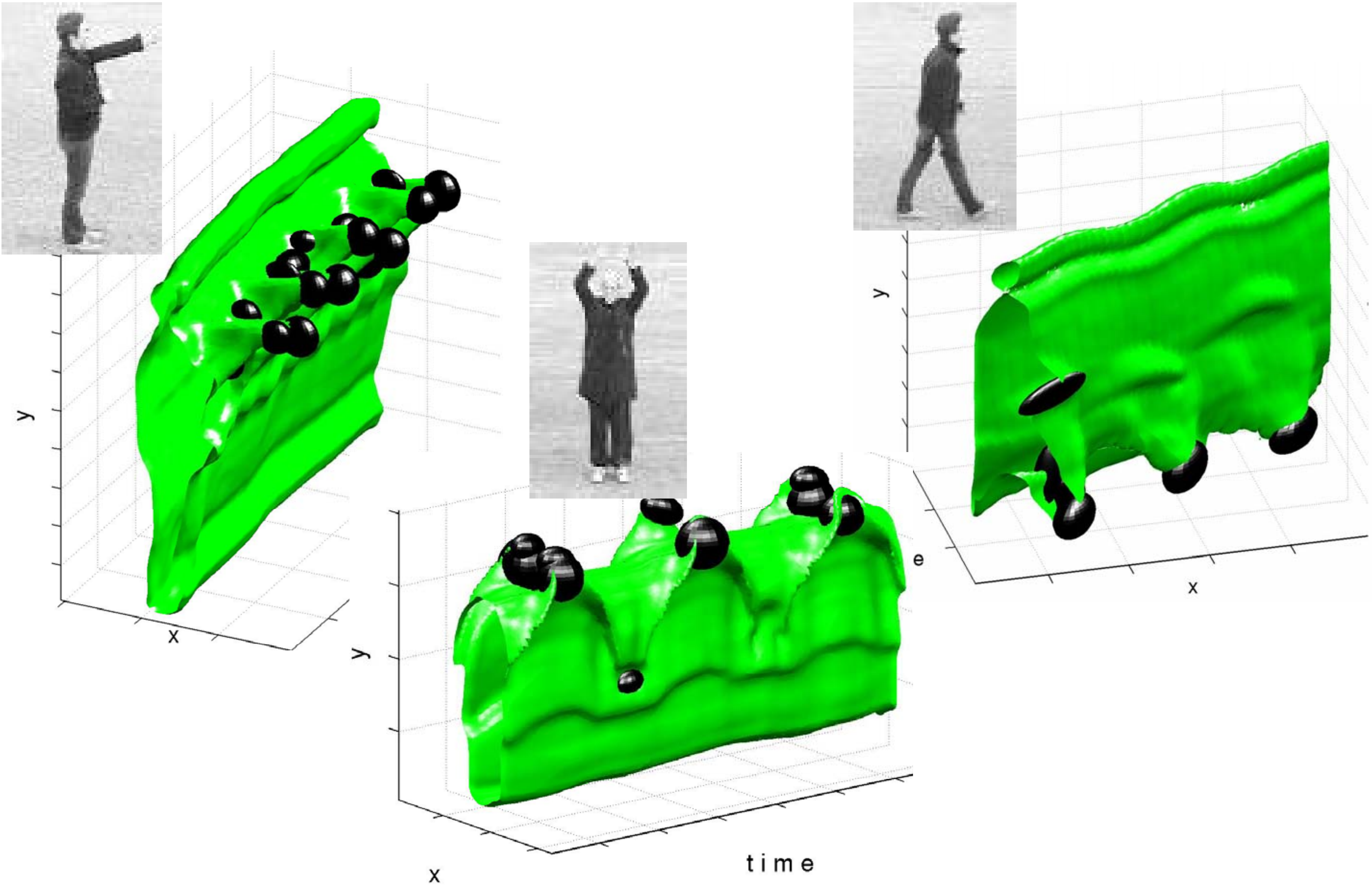


Velocity-adapted interest points

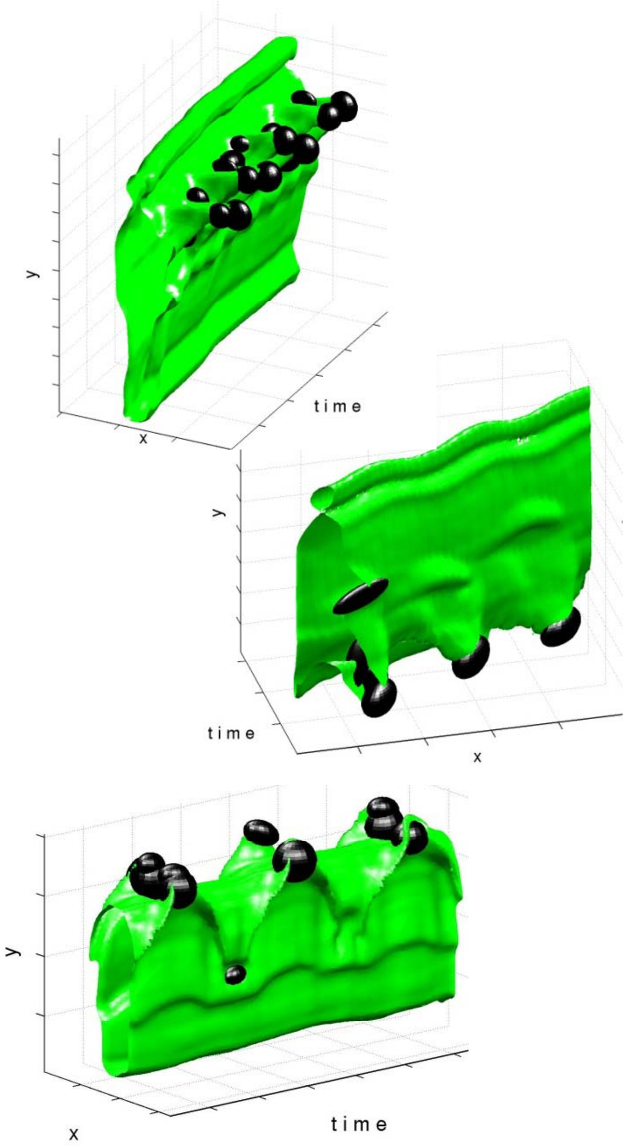


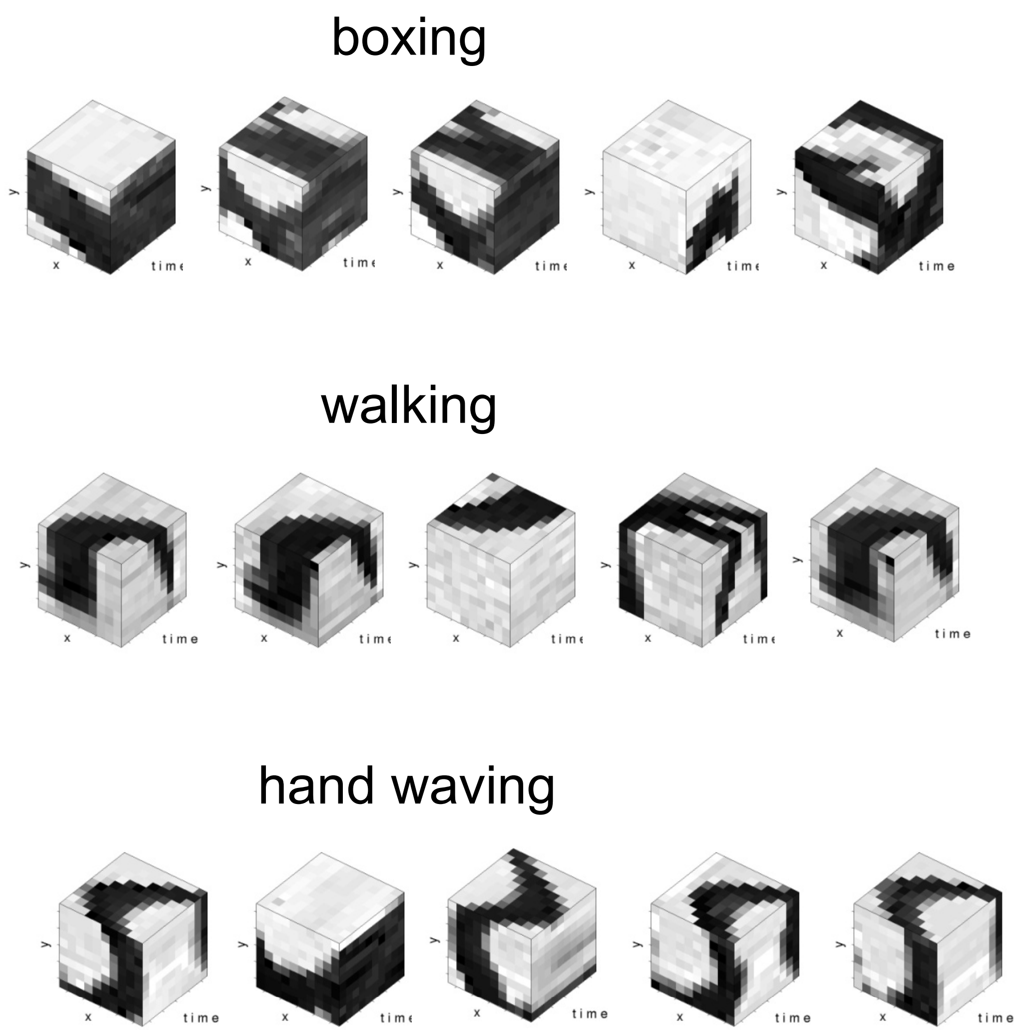


Local features for human actions



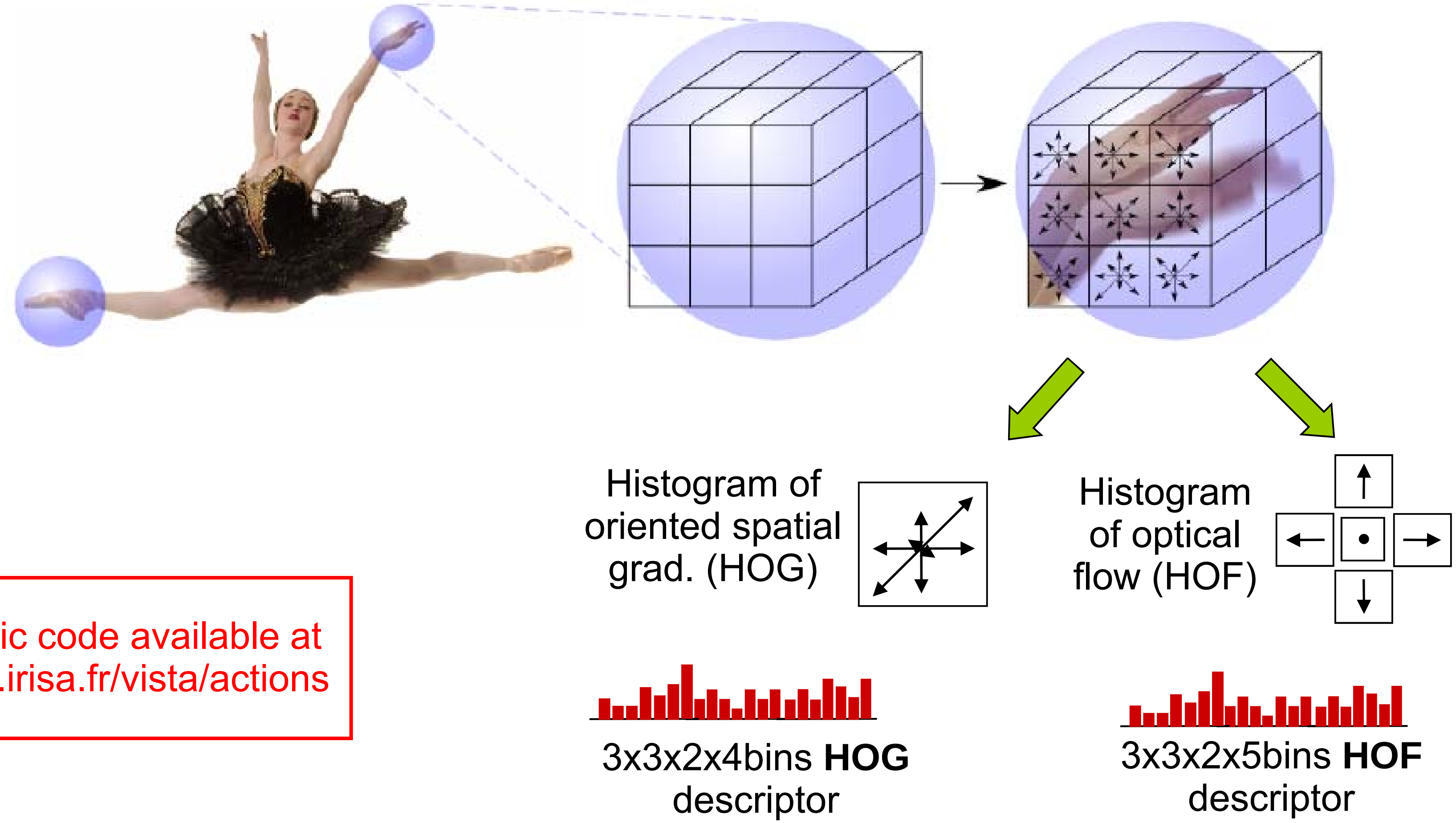
Local features for human actions





Local space-time descriptor: HOG/HOF

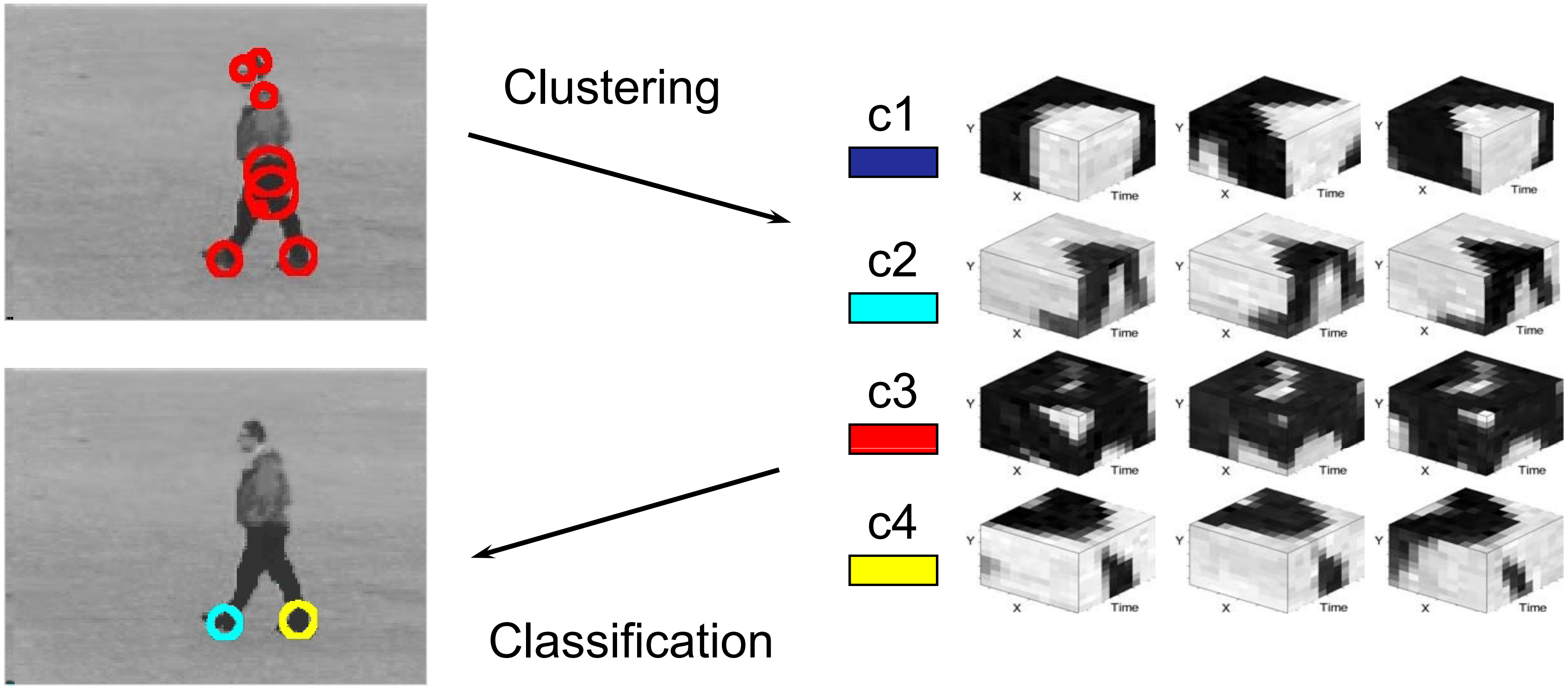
Multi-scale space-time patches



Public code available at www.irisa.fr/vista/actions

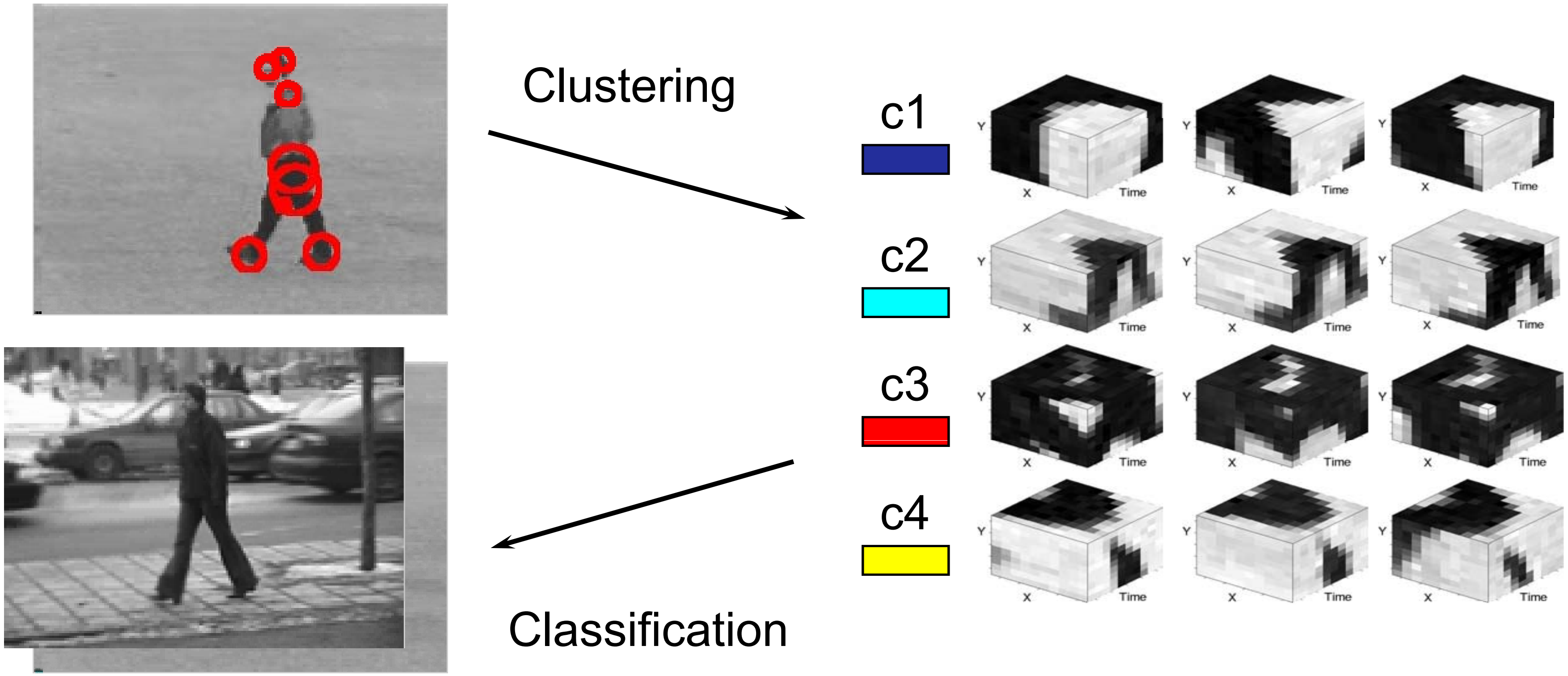
Visual Vocabulary: K-means clustering

- Group similar points in the space of image descriptors using K-means clustering
- Select significant clusters



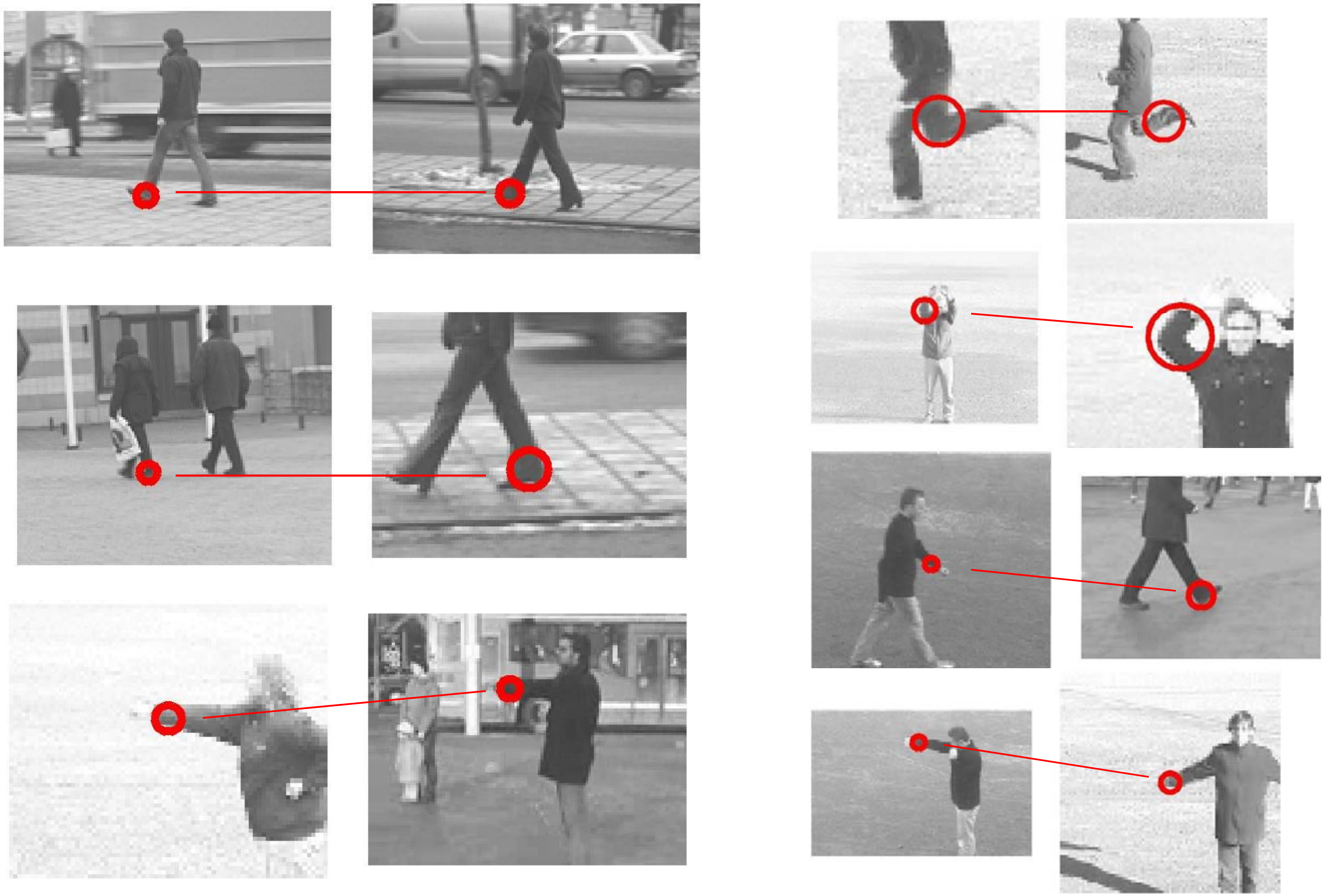
Visual Vocabulary: K-means clustering

- Group similar points in the space of image descriptors using K-means clustering
- Select significant clusters



Local Space-time features: Matching

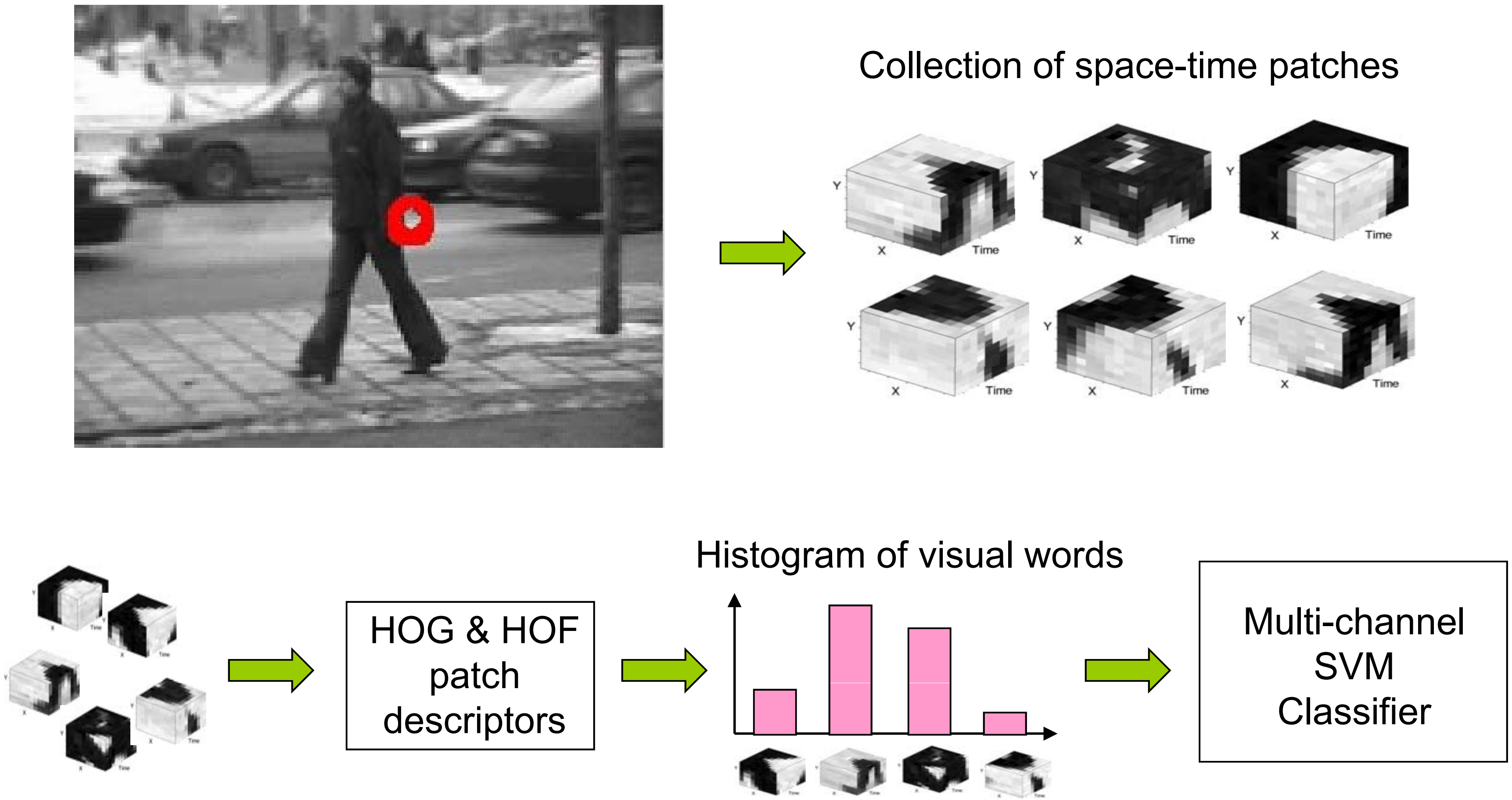
- Find similar events in pairs of video sequences



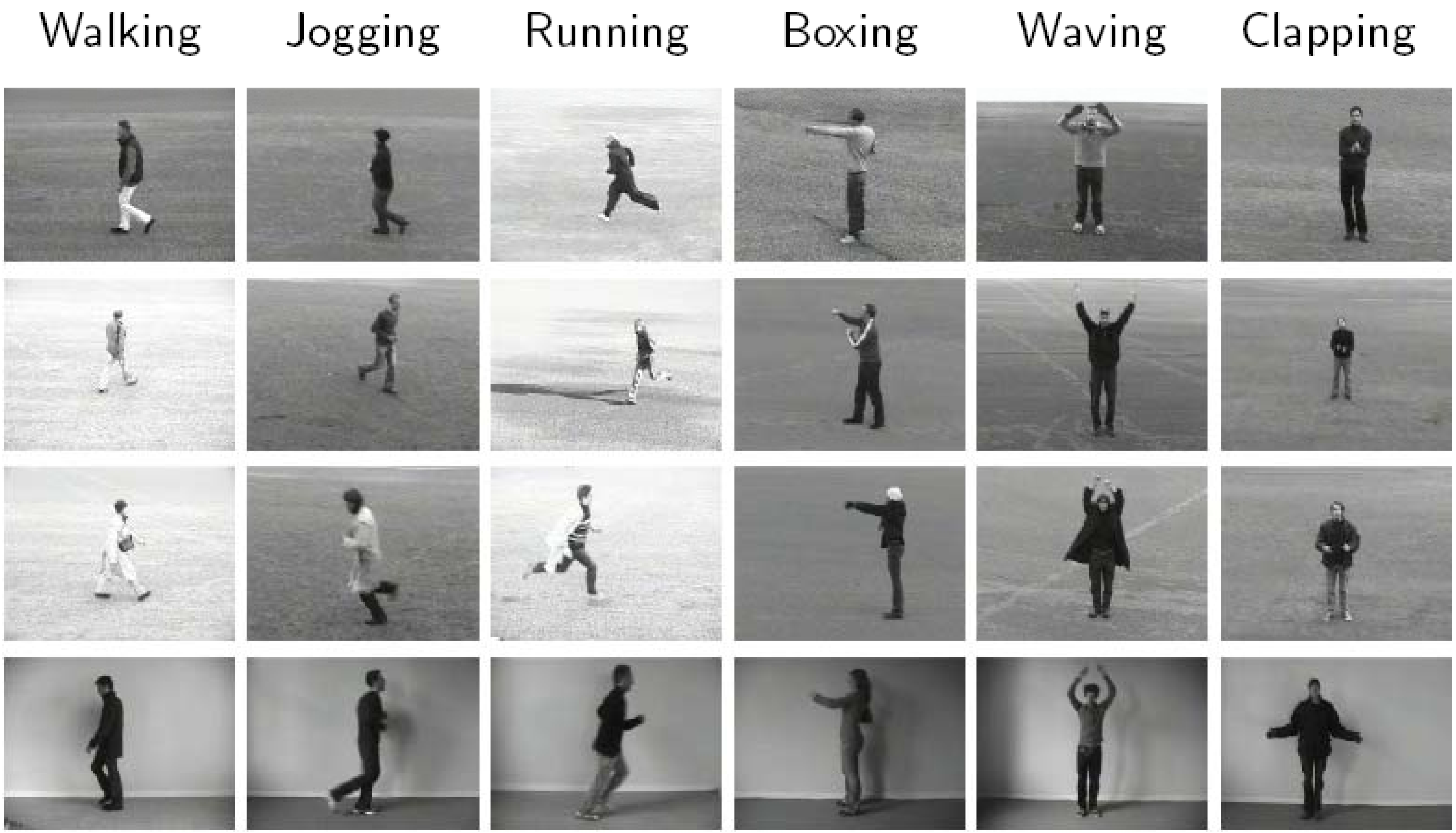
Action Classification: Overview

Bag of space-time features + multi-channel SVM

[Laptev’03, Schuldt’04, Niebles’06, Zhang’07]



Action recognition in KTH dataset



Sample frames from the KTH actions sequences, all six classes (columns) and scenarios (rows) are presented

Classification results on KTH dataset

	Walking	Jogging	Running	Boxing	Waving	Clapping
Walking	.99	.01	.00	.00	.00	.00
Jogging	.04	.89	.07	.00	.00	.00
Running	.01	.19	.80	.00	.00	.00
Boxing	.00	.00	.00	.97	.00	.03
Waving	.00	.00	.00	.00	.91	.09
Clapping	.00	.00	.00	.05	.00	.95

Confusion matrix for KTH actions

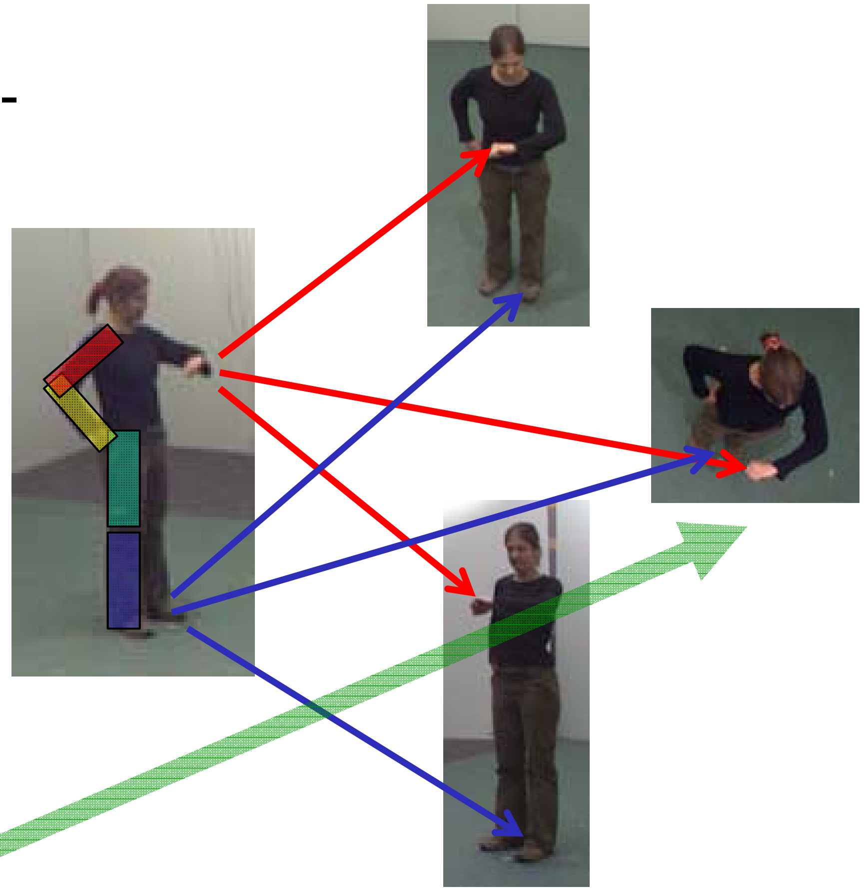
What about 3D?

Local motion and appearance features are not invariant to view changes

Multi-view action recognition

Difficult to apply standard multi-view methods:

- Do not want to search for multi-view point correspondence --- Non-rigid motion, clothing changes, … --> It's Hard!



- Do not want to identify body parts. Current methods are not reliable enough.

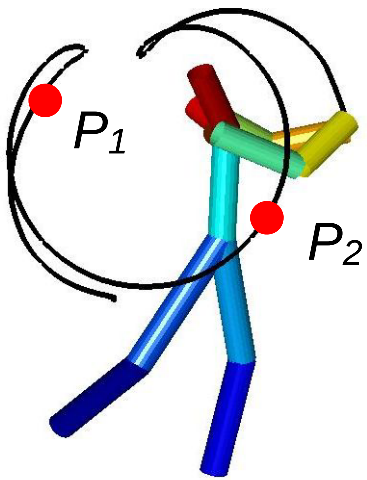
- Yet, want to learn actions from one view and recognize actions in very different views

Temporal self-similarities

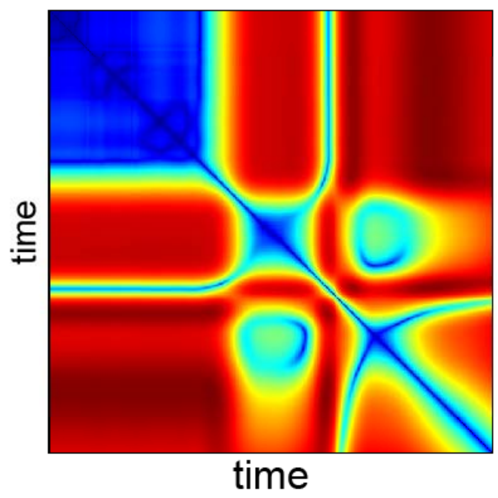
Idea:

- *Cross-view* matching is hard but *cross-time* matching (tracking) is relatively easy.
- Measure self-(dis)similarities across time: $\mathcal{D}(t_1, t_2),\ t_1, t_2 \in (1, ..., T)$

Example: $\mathcal{D}(t_1, t_2) = ||P_1 - P_2||_2$

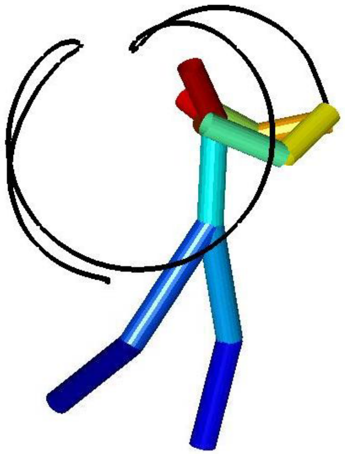


Distance matrix / self-similarity matrix (SSM):

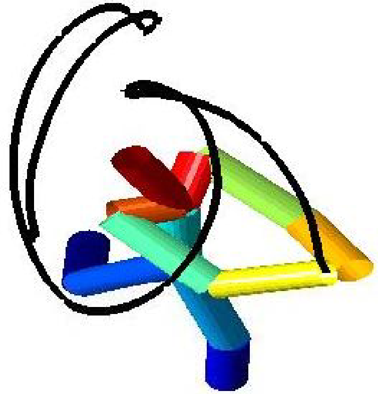


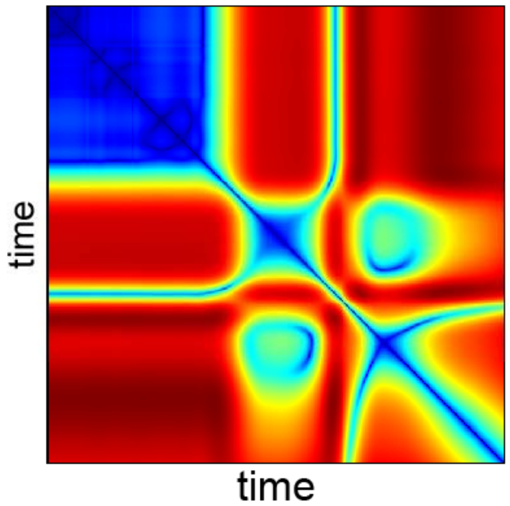
Temporal self-similarities: Multi-views

Side view

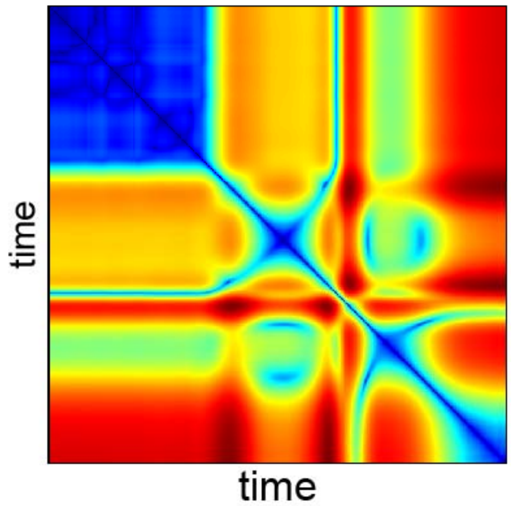


Top view





Appear very similar despite the view change!

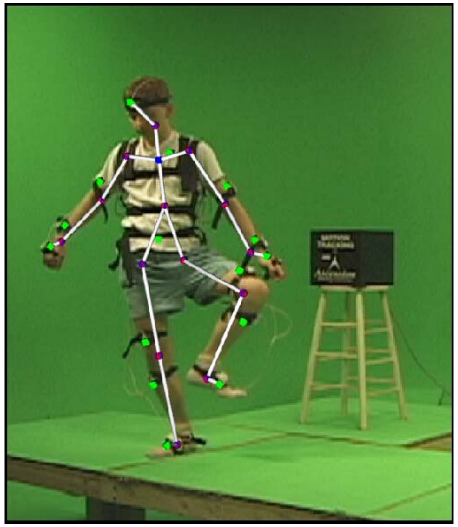


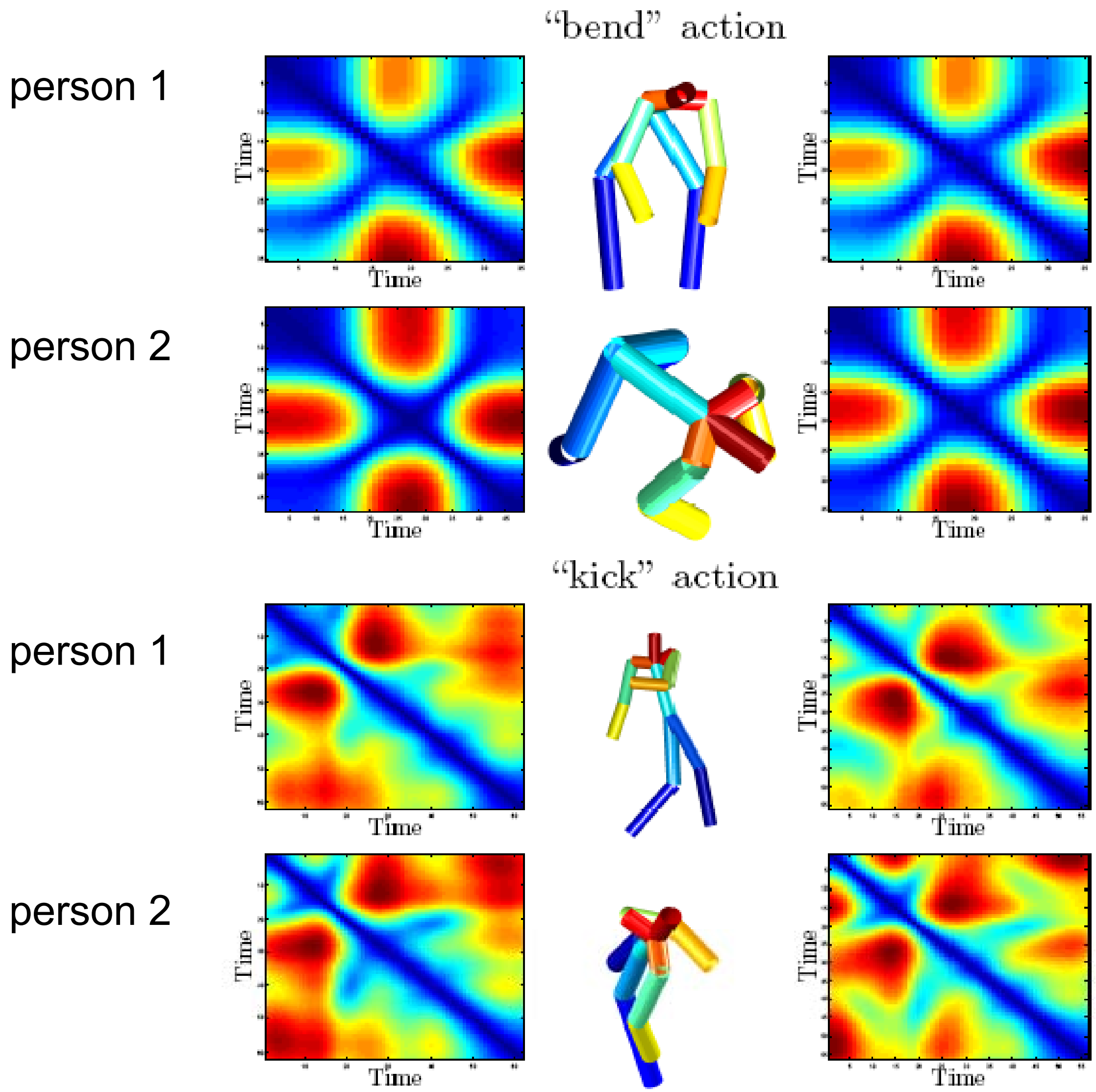
Intuition: 1. Distance between similar poses is low in any view

2. Distance among different poses is likely to be large in most views

Temporal self-similarities: MoCap

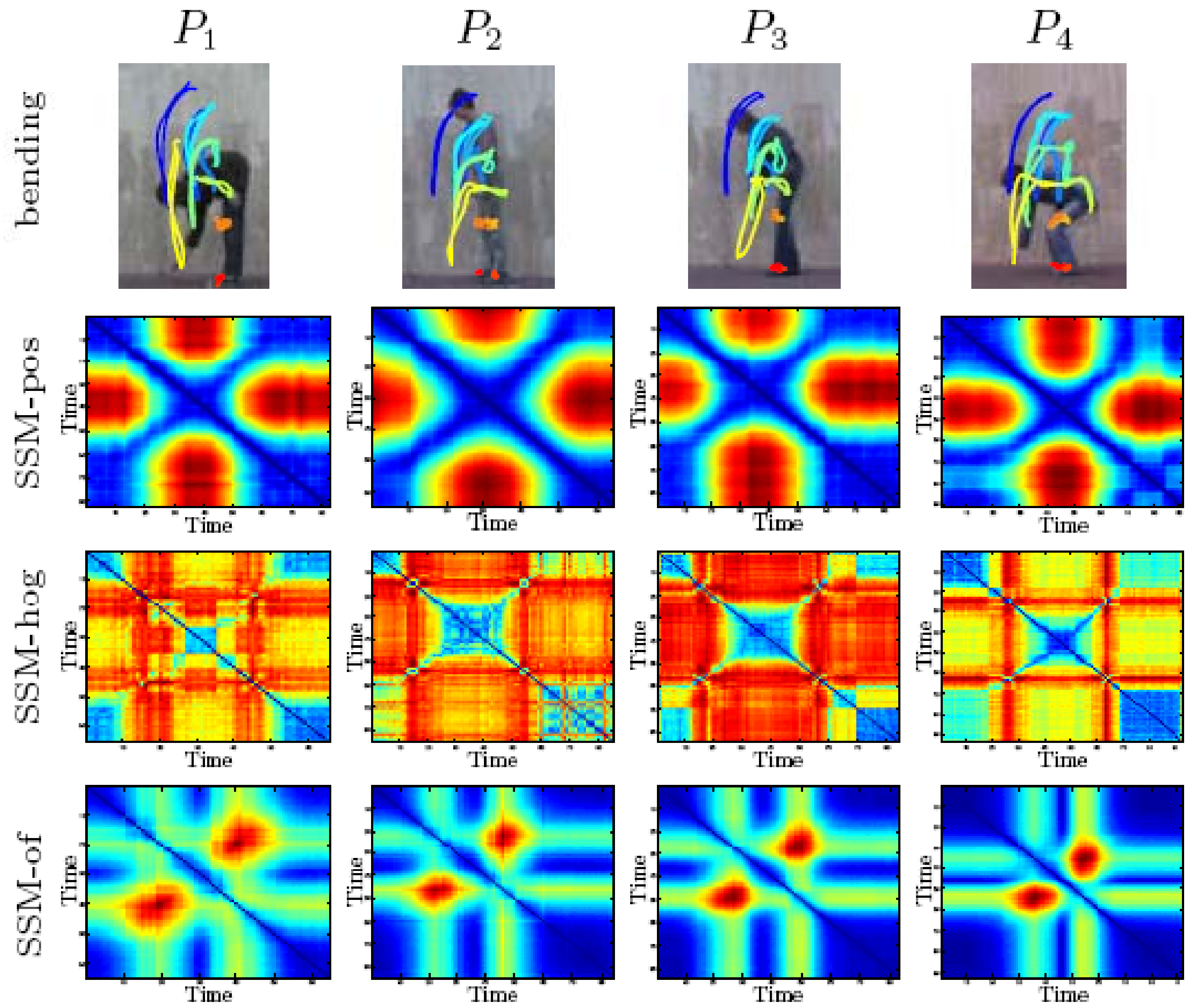
Self-similarities can be measured from Motion Capture (MoCap) data





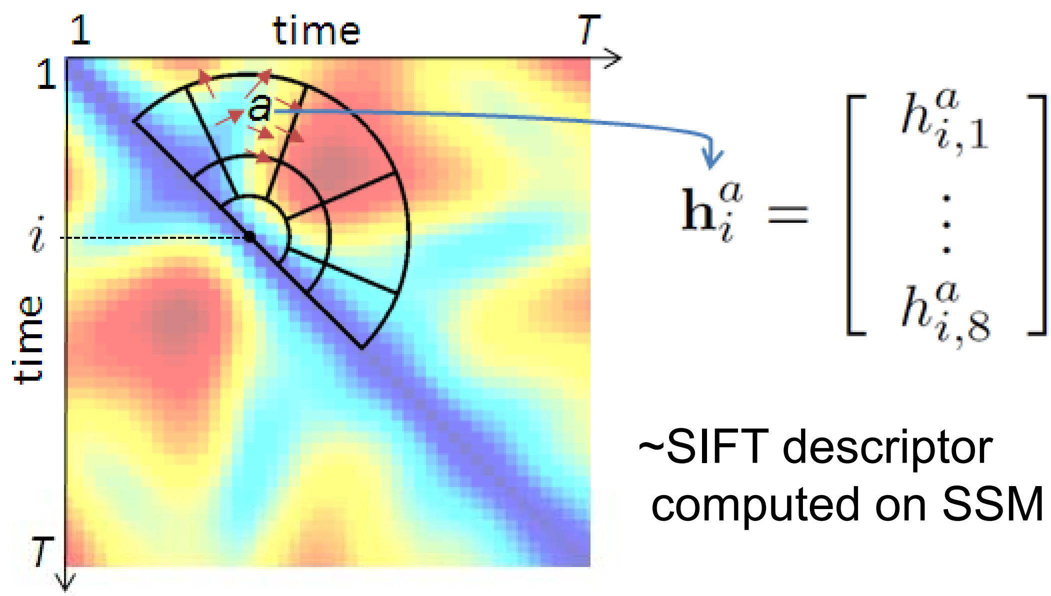
Temporal self-similarities: Video

Self-similarities can be measured directly from video:
HOG or Optical Flow descriptors in image frames



Self-similarity descriptor

Goal:
define a quantitative measure to compare self-similarity matrices



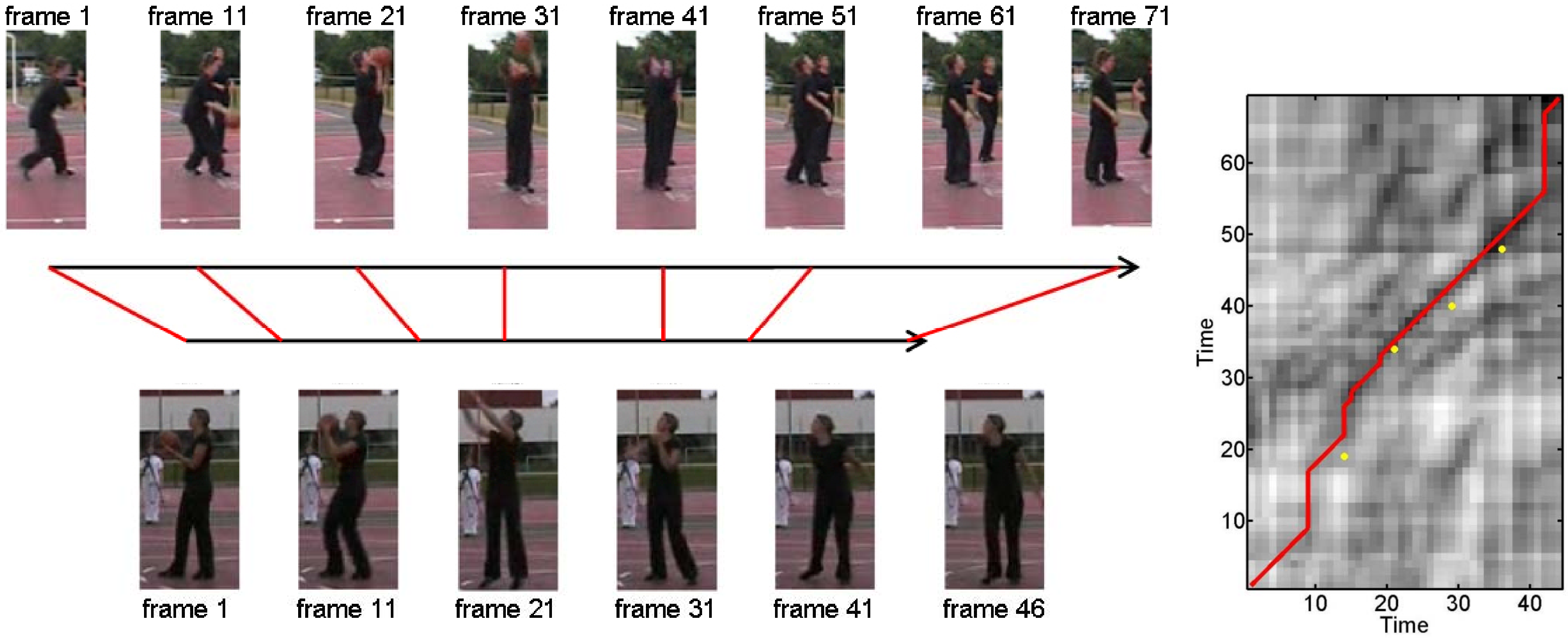
~SIFT descriptor computed on SSM

- Define a local histogram descriptor h_i for each point i on the diagonal.

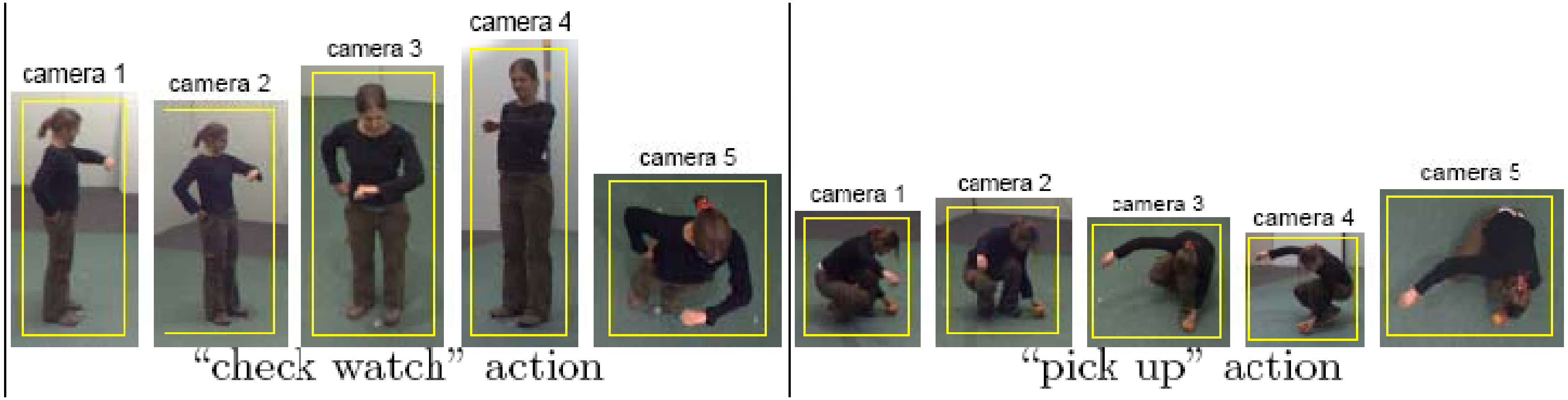
- **Sequence alignment:** Dynamic Programming for two sequences of descriptors $\{h_i\}$, $\{h_j\}$

- **Action recognition:**
  - Visual vocabulary for h
  - BoF representation of $\{h_i\}$
  - SVM

Multi-view alignment



Multi-view action recognition: Video



	Test Cam0	Test Cam1	Test Cam2	Test Cam3	Test Cam4	Test All
Train Cam0	77.0	75.2	69.7	71.8	49.4	68.6
Train Cam1	78.5	77.3	67.9	71.5	48.0	68.6
Train Cam2	70.0	73.0	75.8	68.5	55.2	68.5
Train Cam3	73.6	72.4	67.3	71.2	45.9	66.1
Train Cam4	44.5	41.5	55.2	37.9	68.8	49.6
Train All	77.0	78.8	80.0	73.9	63.3	74.6

cross−camera training/testing same camera training/testing

SSM-based recognition

	Test Cam0	Test Cam1	Test Cam2	Test Cam3	Test Cam4	Test All
Train Cam0	80.0	75.9	42.3	55.6	21.8	55.6
Train Cam1	74.8	83.9	36.5	58.3	23.6	56.0
Train Cam2	43.6	46.1	80.5	64.7	34.2	53.7
Train Cam3	47.0	50.0	45.8	85.5	18.8	49.5
Train Cam4	19.7	19.4	43.5	26.1	73.3	36.0
Train All	80.3	84.5	79.4	84.8	68.5	79.6

cross−camera training/testing same camera training/testing

Alternative view-dependent method (STIP)

What are Human Actions?

Actions in recent datasets:

Is it just about kinematics?

Should actions be defined by the *purpose?*

Kinematics + Objects

What are Human Actions?

Actions in recent datasets:


Is it just about kinematics?

Should actions be defined by the *purpose?*


Kinematics + Objects + Scenes

Action recognition in realistic settings

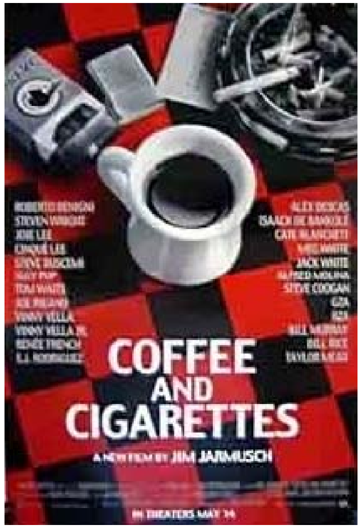

Standard
action
datasets

Actions “In the Wild”:






Action Dataset and Annotation

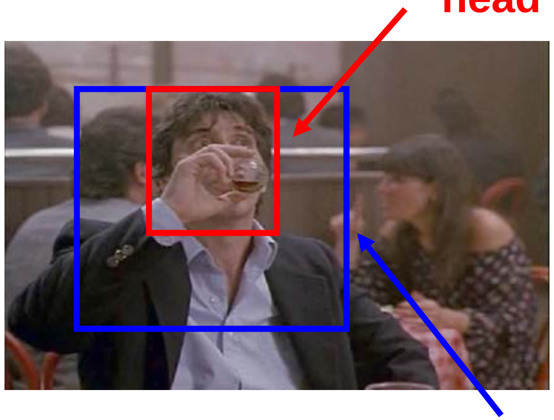


Manual annotation of drinking actions in movies:
“Coffee and Cigarettes”; “Sea of Love”

“*Drinking*”: 159 annotated samples
“*Smoking*”: 149 annotated samples

Spatial annotation

head rectangle



torso rectangle

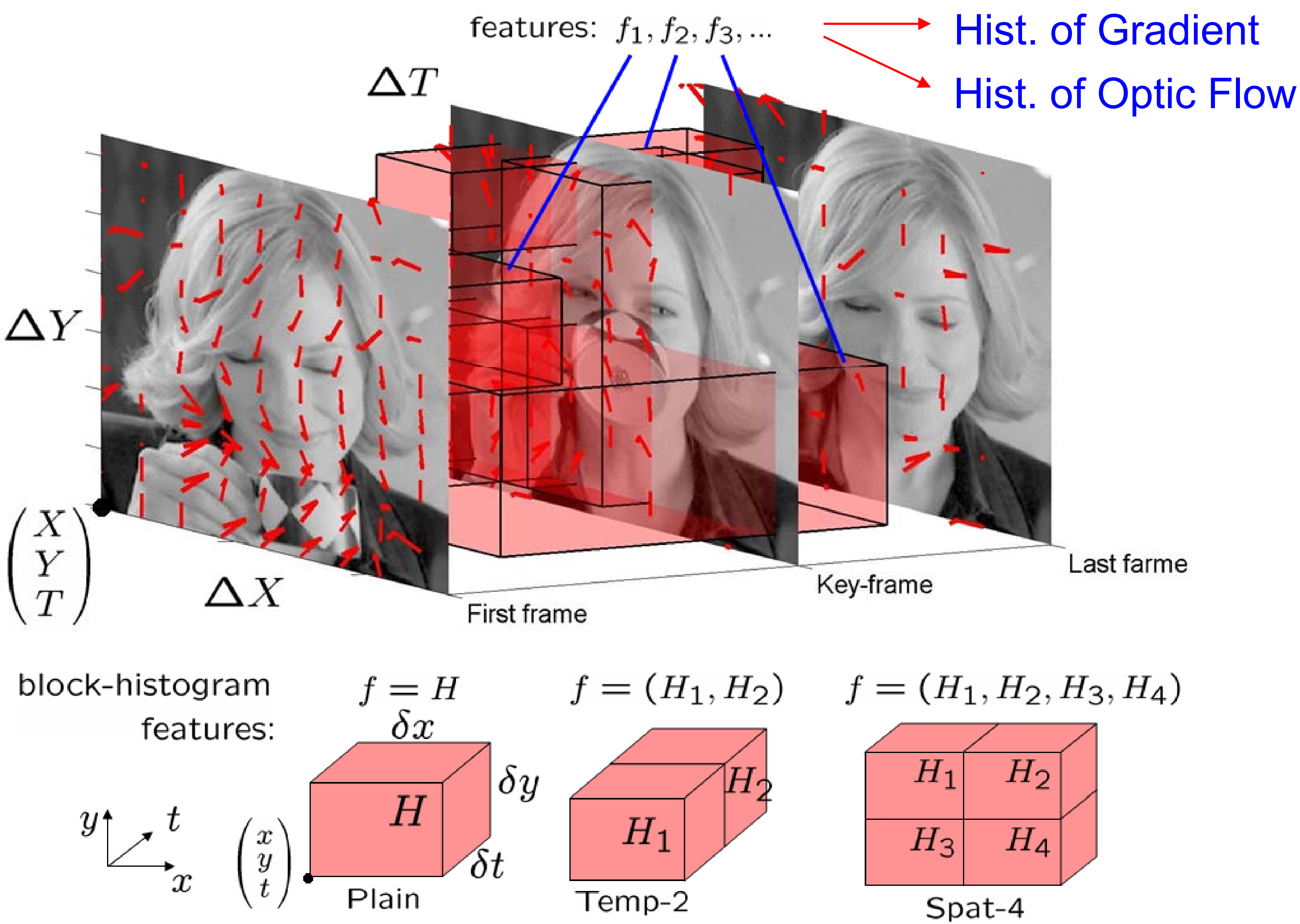
Temporal annotation

First frame **Keyframe** **Last frame**

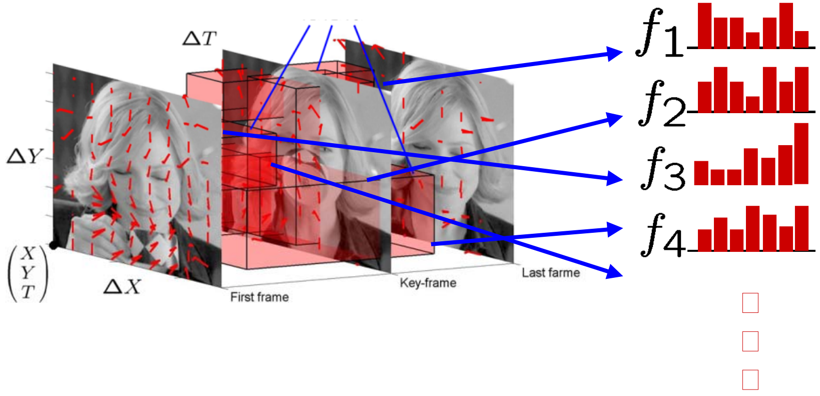
“Drinking” action samples

training samples　　　　test samples

Action representation



Action learning



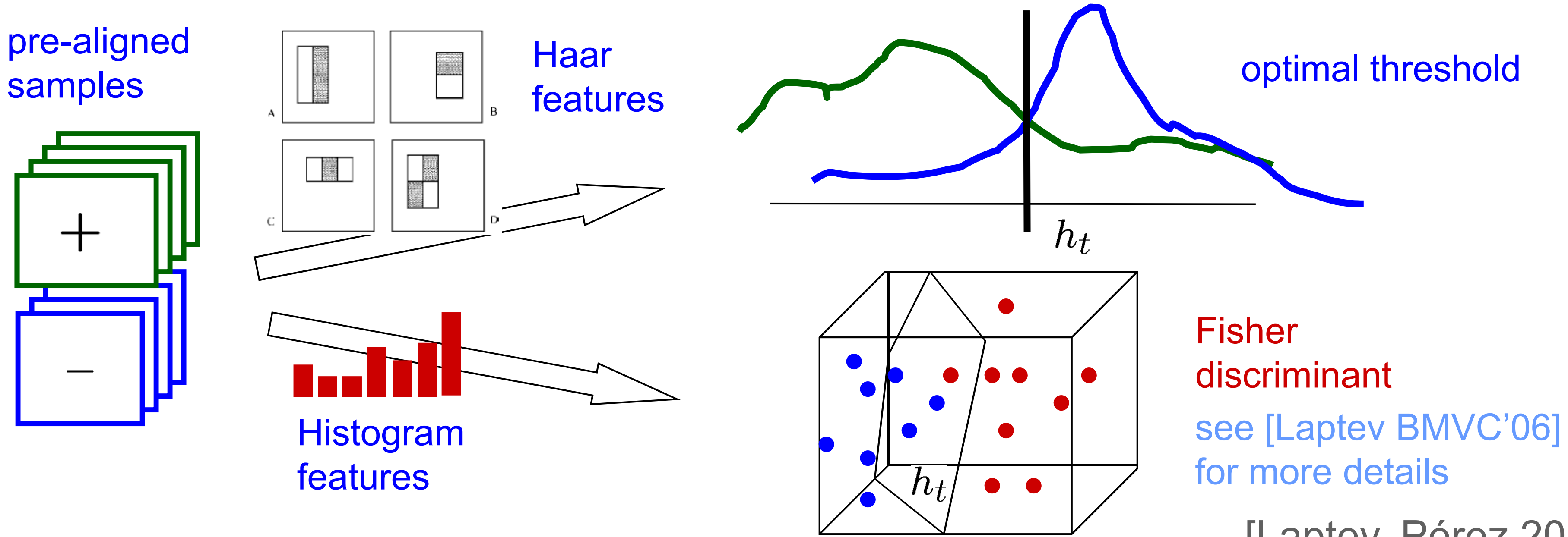
boosting

selected features

$$H(z) = \mathrm{sgn}\left(\sum_{t=1}^{T} \alpha_t h_t(f_t)\right)$$

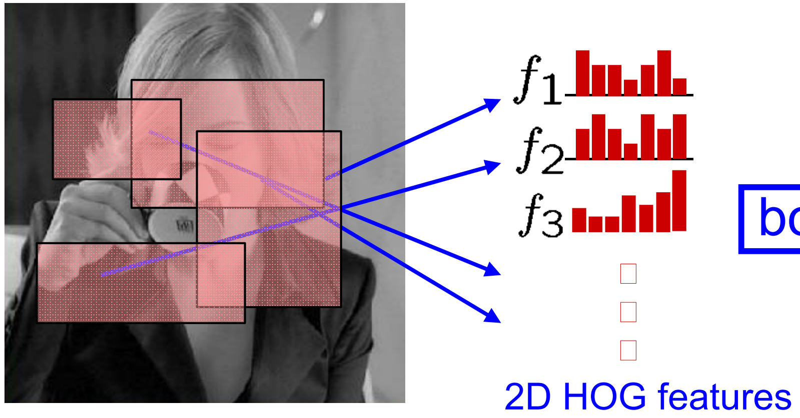
weak classifier

AdaBoost:
- Efficient discriminative classifier [Freund&Schapire'97]
- Good performance for face detection [Viola&Jones'01]



[Laptev, Pérez 2007]

Key-frame action classifier



boosting

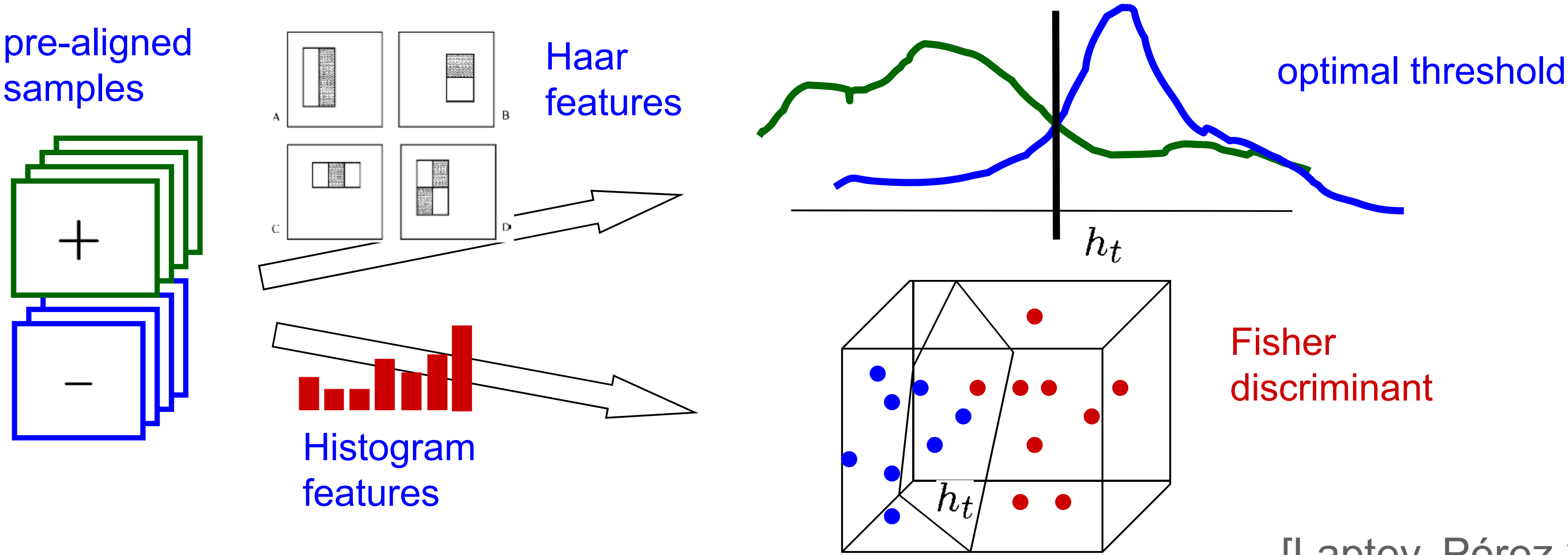
$$H(z) = \mathrm{sgn}\left(\sum_{t=1}^{T} \alpha_t h_t(f_t)\right)$$

selected features

weak classifier

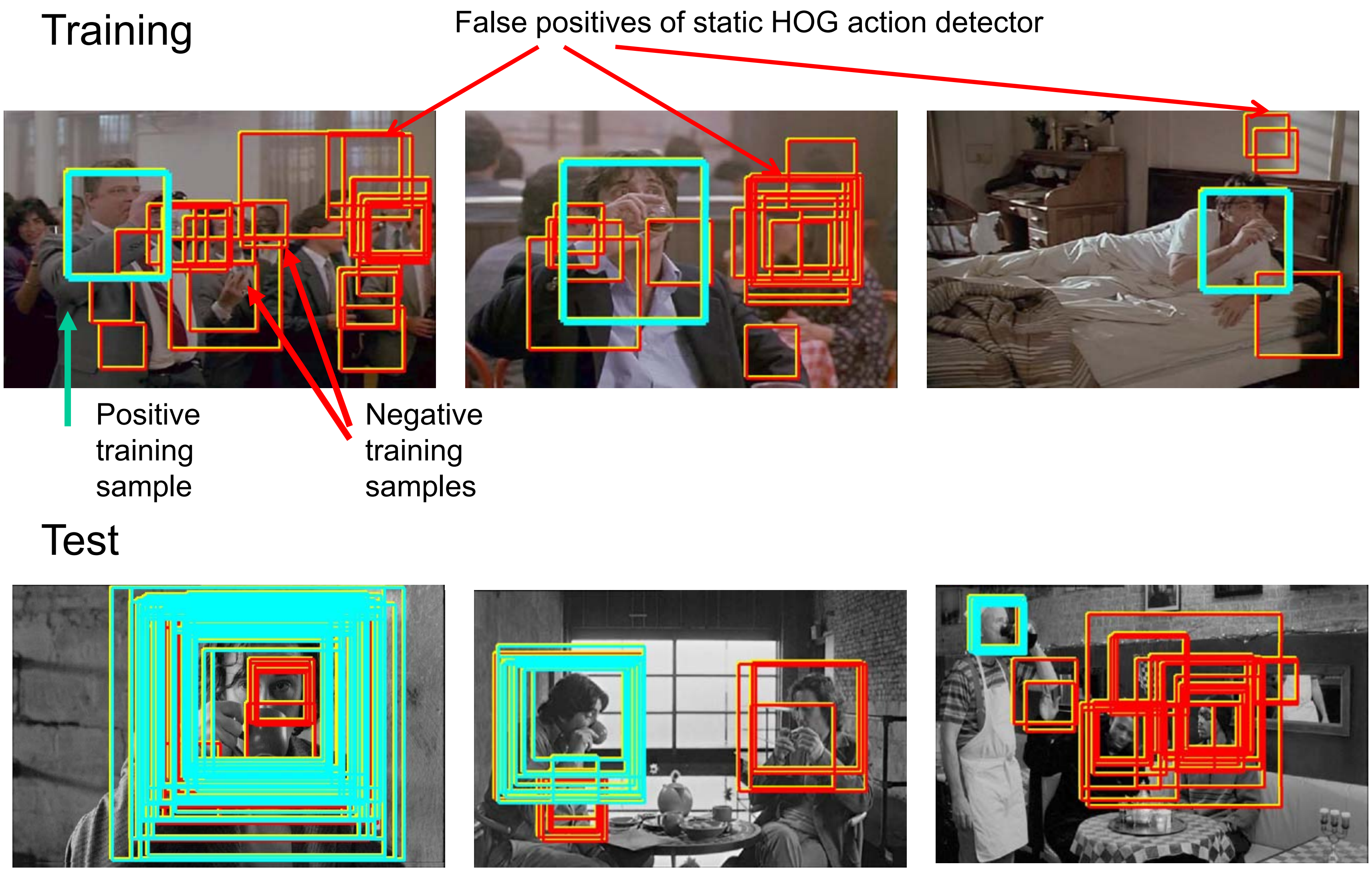
AdaBoost:

- Efficient discriminative classifier [Freund&Schapire'97]
- Good performance for face detection [Viola&Jones'01]



[Laptev, Pérez 2007]

Keyframe priming



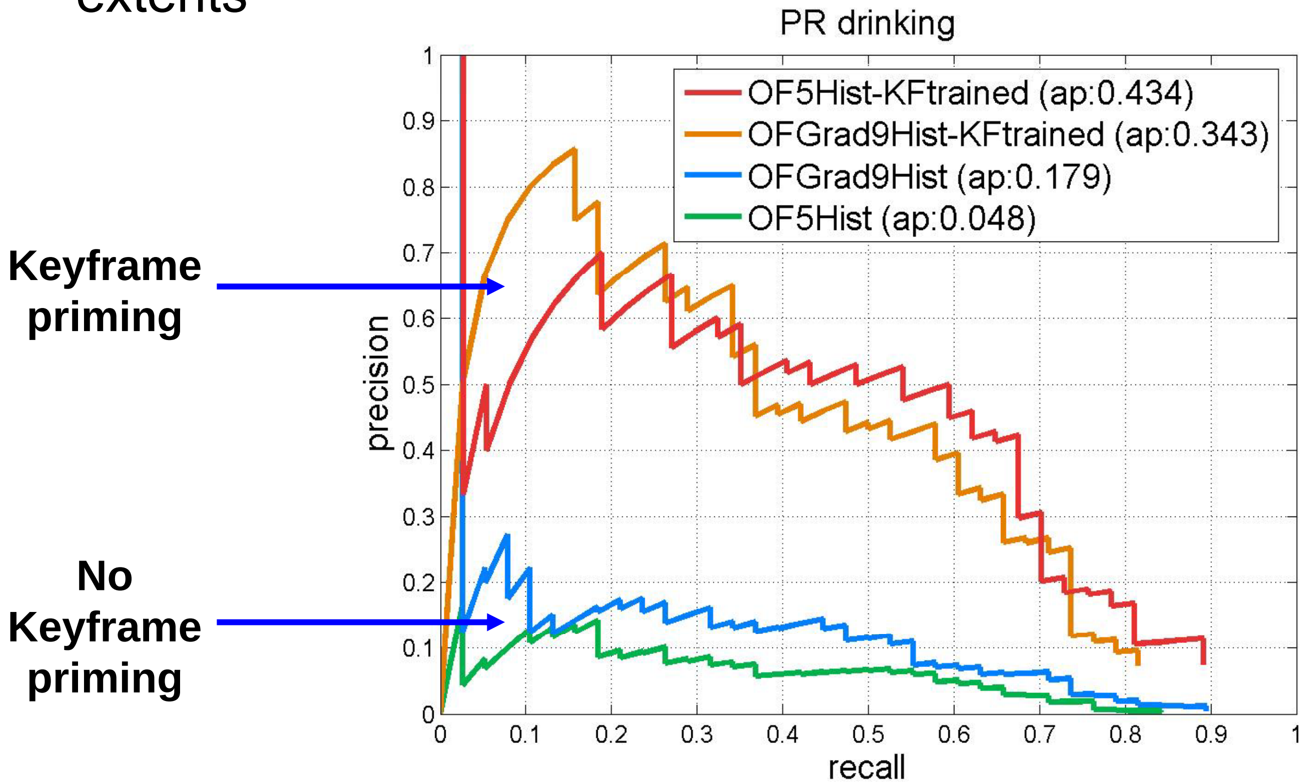
Action detection

Test set:

- 25min from "Coffee and Cigarettes" with GT 38 drinking actions
- No overlap with the training set in subjects or scenes

Detection:

- search over all space-time locations and spatio-temporal extents



Action Detection (ICCV 2007)

Test episodes from the movie “Coffee and cigarettes”

Video available at http://www.irisa.fr/vista/Equipe/People/Laptev/actiondetection.html

20 most confident detections

Learning Actions from Movies

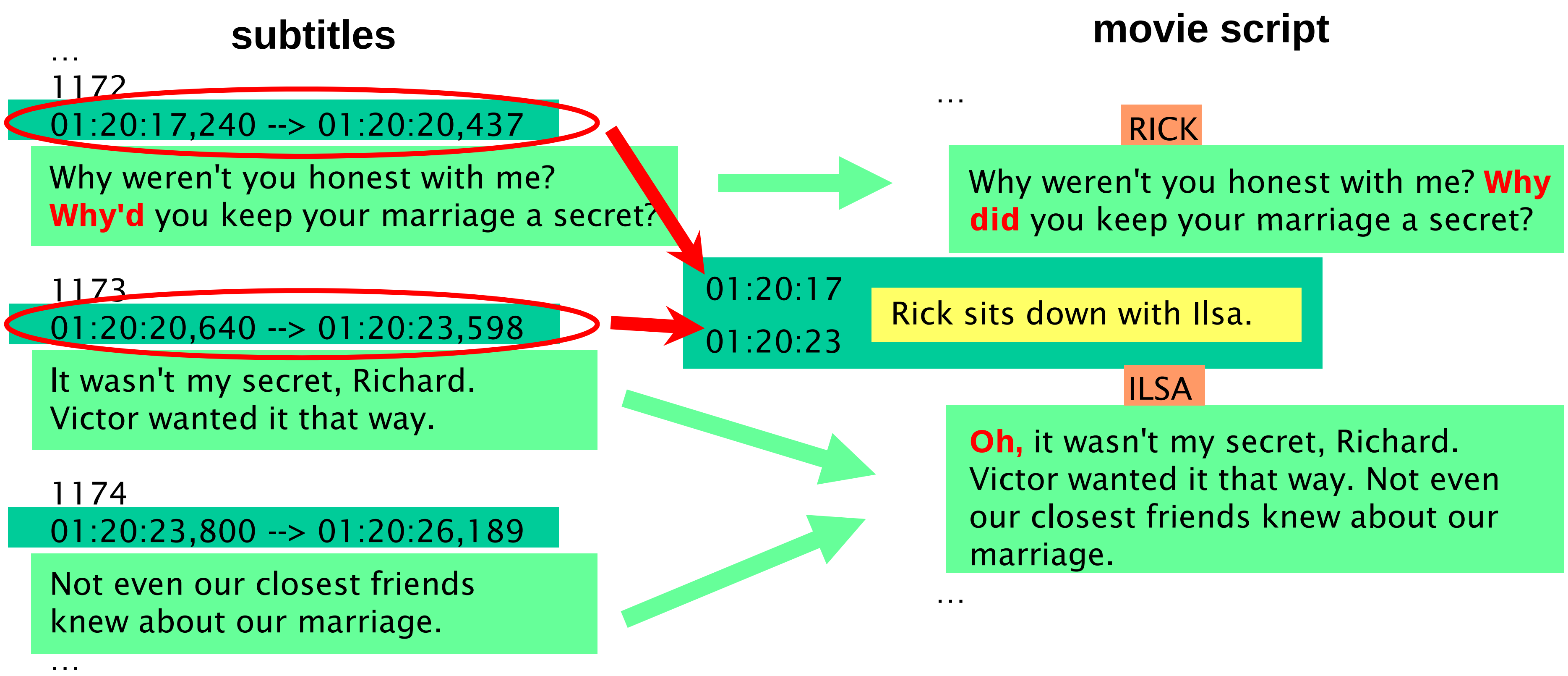
- Realistic variation of human actions
- Many classes and many examples per class

Problems:

- Typically only a few class-samples per movie
- Manual annotation is very time consuming

Automatic video annotation with scripts

- Scripts available for >500 movies (no time synchronization)
  www.dailyscript.com, www.movie-page.com, www.weeklyscript.com …
- Subtitles (with time info.) are available for the most of movies
- Can transfer time to scripts by text alignment



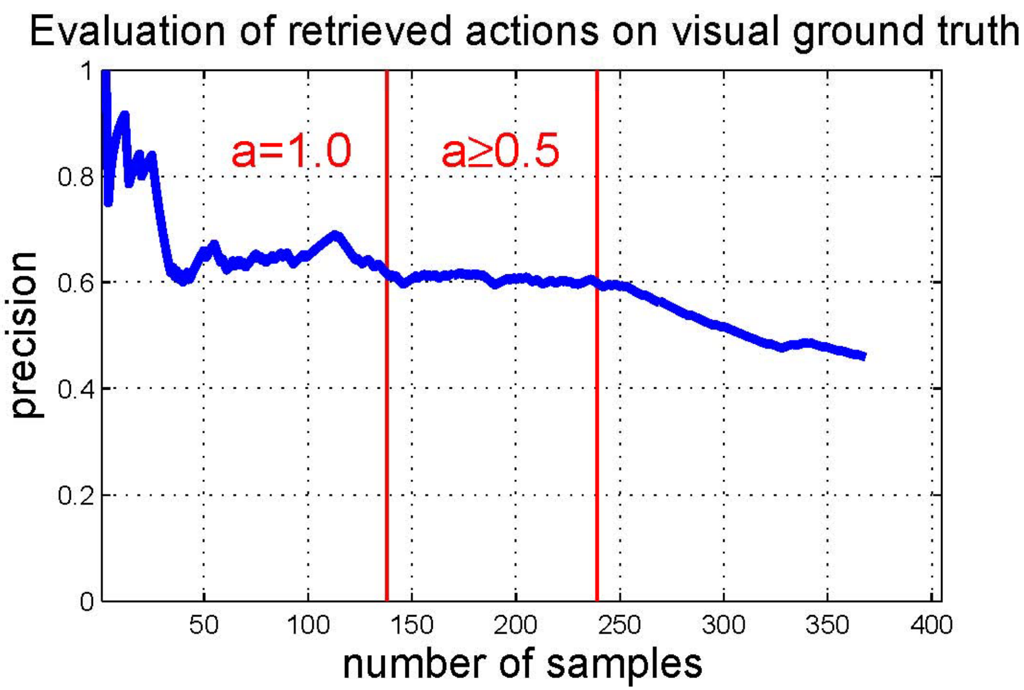
Script-based action annotation

- **On the good side:**
  - Realistic variation of actions: subjects, views, etc…
  - Many examples per class, many classes
  - No extra overhead for new classes
  - Actions, objects, scenes and their combinations
  - Character names may be used to resolve “who is doing what?”

- **Problems:**
  - No spatial localization
  - Temporal localization may be poor
  - Missing actions: e.g. scripts do not always follow the movie
  - Annotation is incomplete, not suitable as ground truth for testing action detection
  - Large within-class variability of action classes *in text*

Script alignment: Evaluation

- Annotate action samples *in text*
- Do automatic script-to-video alignment
- Check the correspondence of actions in scripts and movies



a: quality of subtitle-script matching

Example of a “visual false positive”

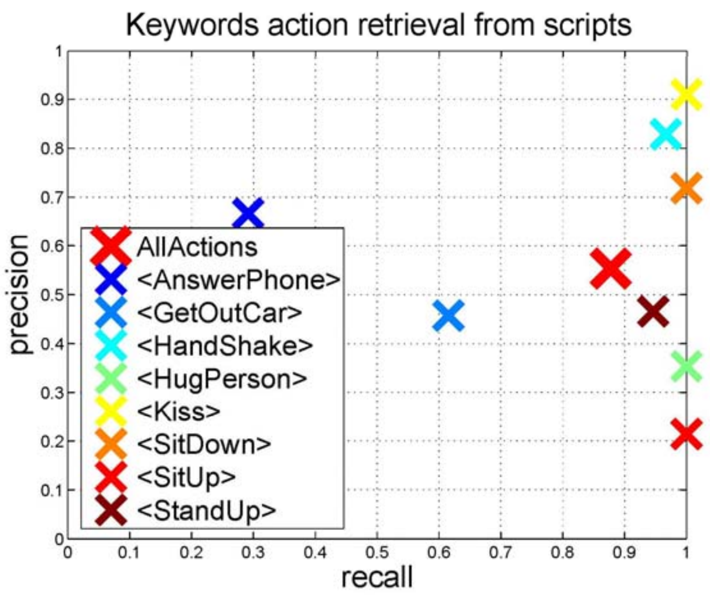
A black car pulls up, two army officers get out.

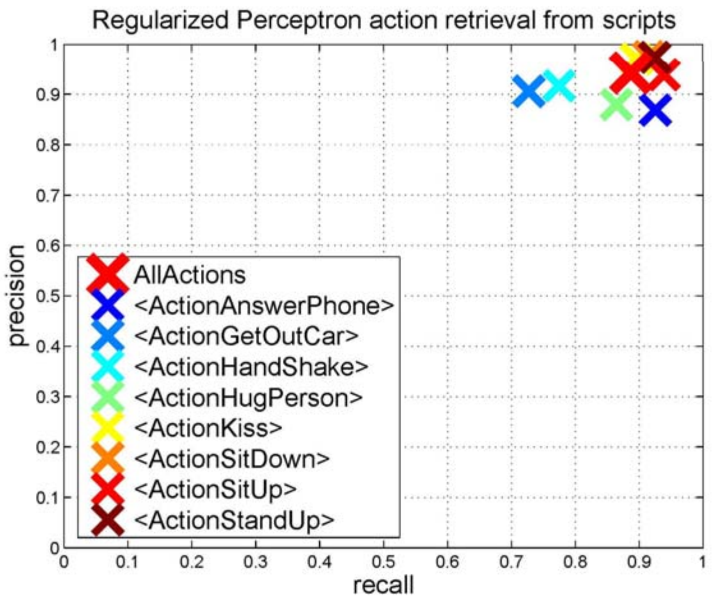
Text-based action retrieval

- Large variation of action expressions in text:

GetOutCar action:	*"… Will gets out of the Chevrolet. …"* *"… Erin exits her new truck…"*
Potential false positives:	*"…About to sit down, he freezes…"*

- => Supervised text classification approach





Automatically annotated action samples

AnswerPhone

GetOutCar

HandShake

HugPerson

Kiss

SitDown

SitUp

StandUp

[Laptev, Marszałek, Schmid, Rozenfeld 2008]

Hollywood-2 actions dataset

Actions			
	Training subset (clean)	Training subset (automatic)	Test subset (clean)
AnswerPhone	66	59	64
DriveCar	85	90	102
Eat	40	44	33
FightPerson	54	33	70
GetOutCar	51	40	57
HandShake	32	38	45
HugPerson	64	27	66
Kiss	114	125	103
Run	135	187	141
SitDown	104	87	108
SitUp	24	26	37
StandUp	132	133	146
All Samples	**823**	**810**	**884**

Training and test samples are obtained from 33 and 36 distinct movies respectively.

Hollywood-2 dataset is on-line:
http://www.irisa.fr/vista/actions/hollywood2

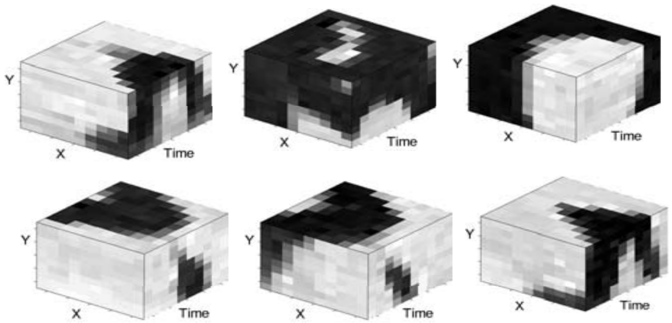
[Laptev, Marszałek, Schmid, Rozenfeld 2008]

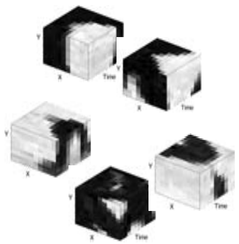
Action Classification: Overview

Bag of space-time features + multi-channel SVM

[Laptev'03, Schuldt'04, Niebles'06, Zhang'07]

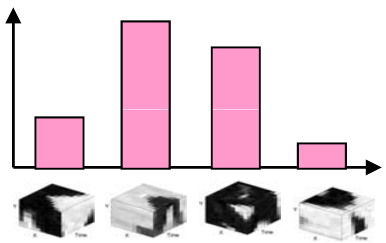
Collection of space-time patches





HOG & HOF patch descriptors

Histogram of visual words



Multi-channel SVM Classifier

Action classification (CVPR08)

Test episodes from movies “The Graduate”, “It’s a Wonderful Life”, “Indiana Jones and the Last Crusade”

Actions in Context (CVPR 2009)

- Human actions are frequently correlated with particular scene classes

  Reasons: *physical properties* and *particular purposes* of scenes

Eating -- *kitchen*

Eating -- *cafe*

Running -- *road*

Running -- *street*

Mining scene captions

01:22:00
01:22:03

ILSA

I wish I didn't love you so much.

She snuggles closer to Rick.

CUT TO:

EXT. RICK'S CAFE - NIGHT

Laszlo and Carl make their way through the darkness toward a side entrance of Rick's. They run inside the entryway.

The headlights of a speeding police car sweep toward them.

They flatten themselves against a wall to avoid detection.

The lights move past them.

01:22:15
01:22:17

CARL

I think we lost them.

...

Mining scene captions

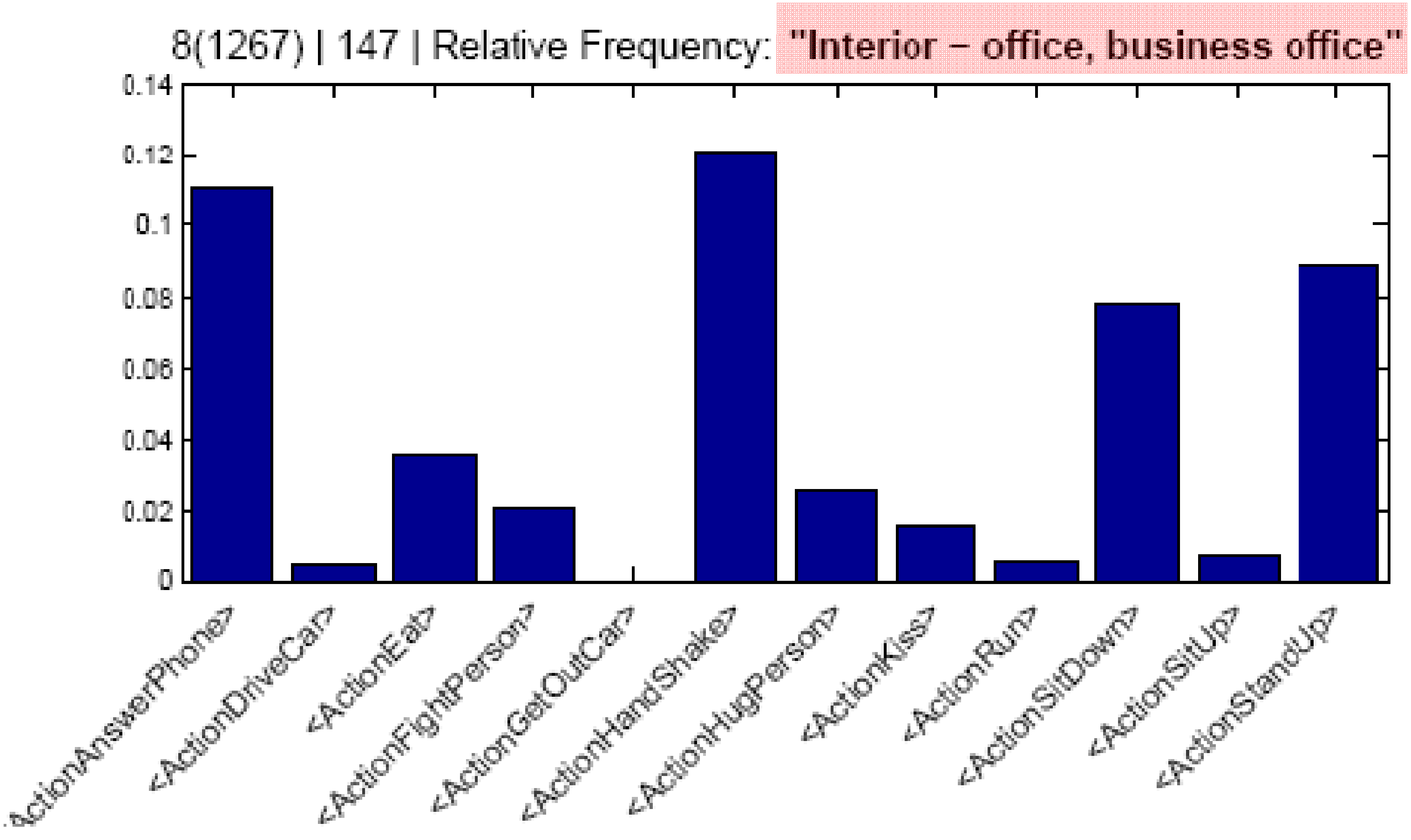
INT. TRENDY RESTAURANT - NIGHT
INT. MARSELLUS WALLACE'S DINING ROOM MORNING
EXT. STREETS BY DORA'S HOUSE - DAY.
INT. MELVIN'S APARTMENT, BATHROOM – NIGHT
EXT. NEW YORK CITY STREET NEAR CAROL'S RESTAURANT – DAY
INT. CRAIG AND LOTTE'S BATHROOM - DAY

- Maximize word frequency ⇒ street, living room, bedroom, car ….
- Merge words with similar senses using WordNet:

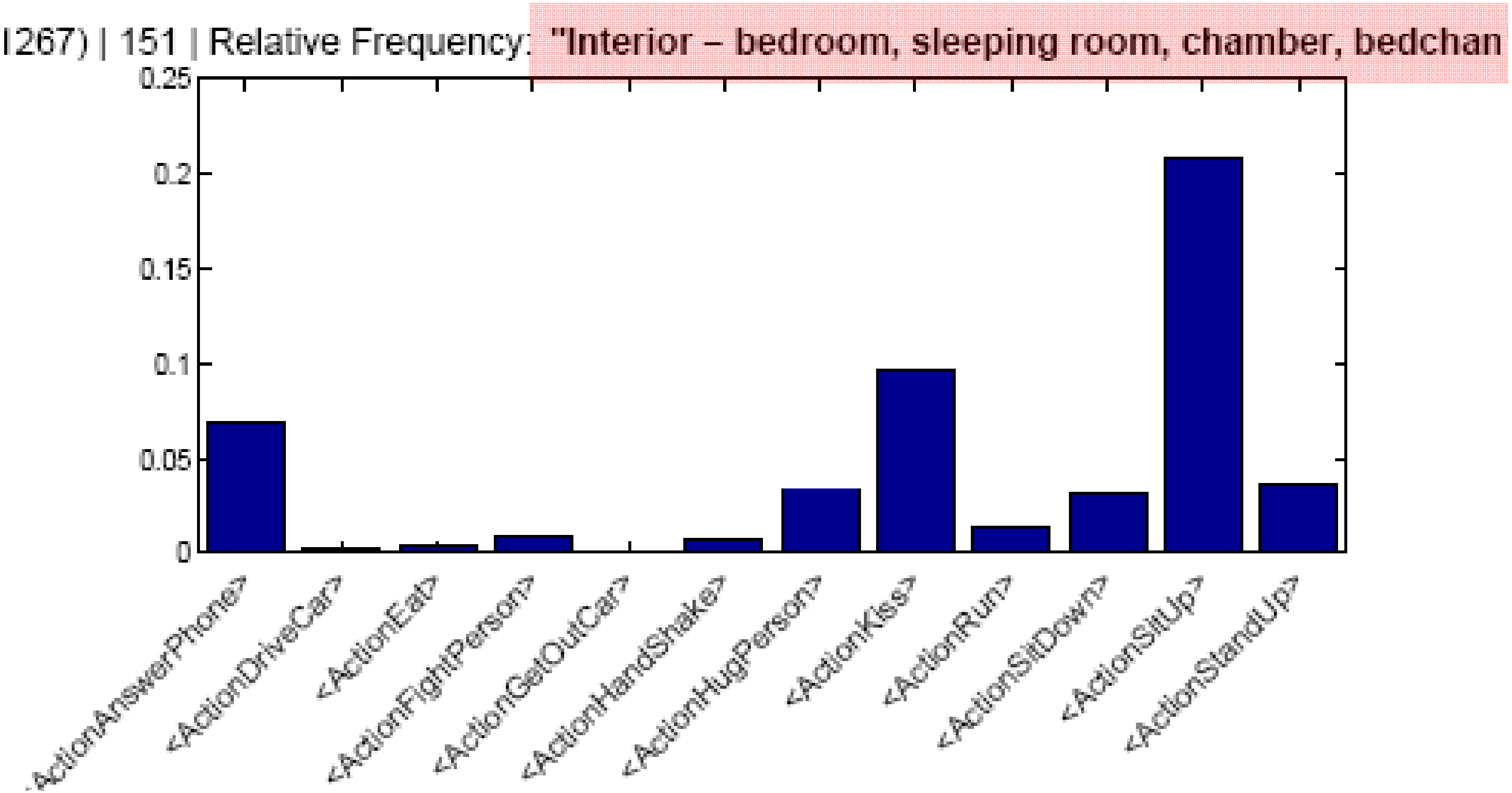
  taxi -> car, cafe -> restaurant

- Measure correlation of words with actions (in scripts) and
- Re-sort words by the entropy $S = -k \sum P_i \ln P_i$
  for P = p(action | word)

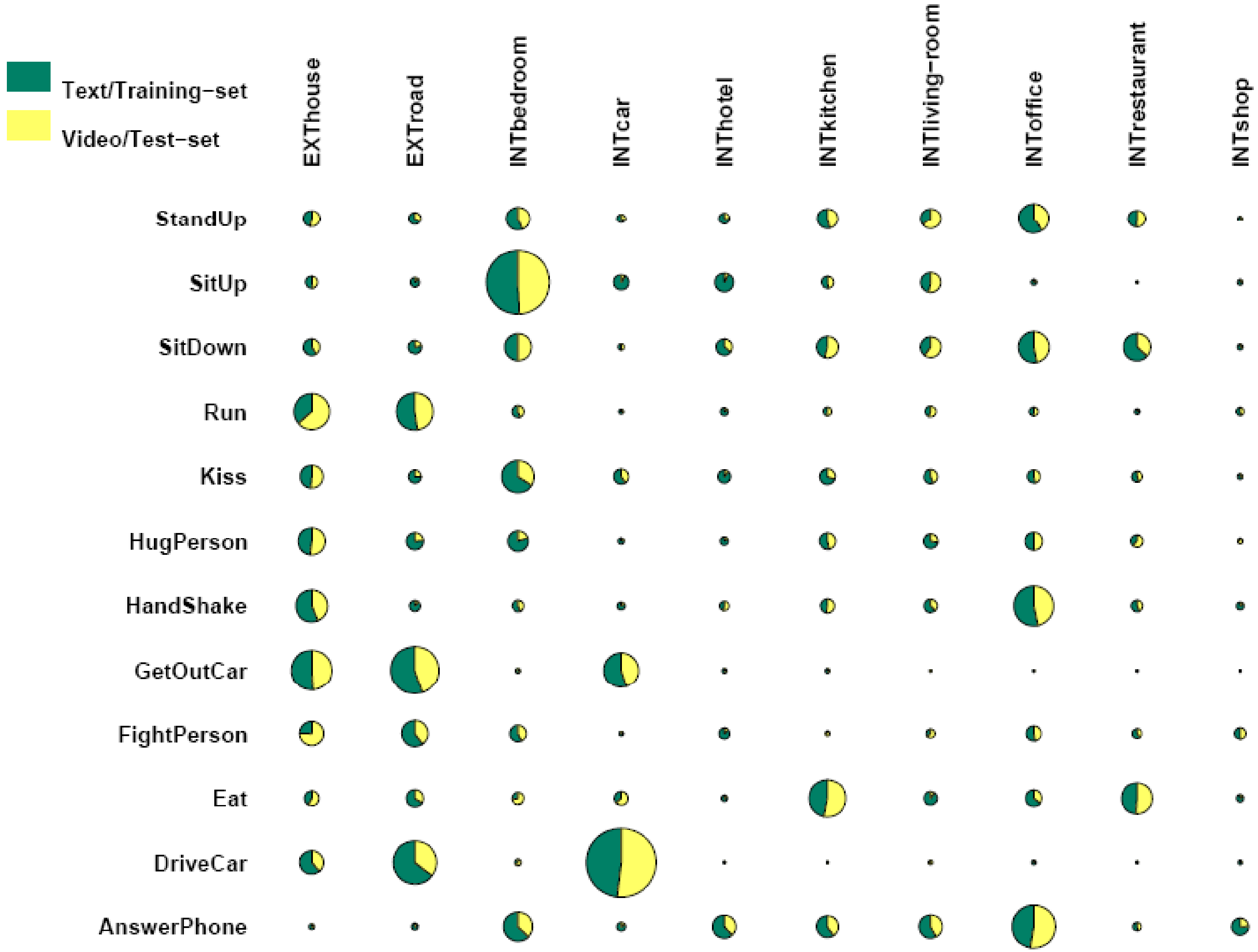
Co-occurrence of actions and scenes in scripts



Co-occurrence of actions and scenes in scripts



Co-occurrence of actions and scenes in text vs. video



Automatic gathering of relevant scene classes and visual samples

	Auto-Train-Actions	Clean-Test-Actions
AnswerPhone	59	64
DriveCar	90	102
Eat	44	33
FightPerson	33	70
GetOutCar	40	57
HandShake	38	45
HugPerson	27	66
Kiss	125	103
Run	187	141
SitDown	87	108
SitUp	26	37
StandUp	133	146
All Samples	810	884

(a) Actions

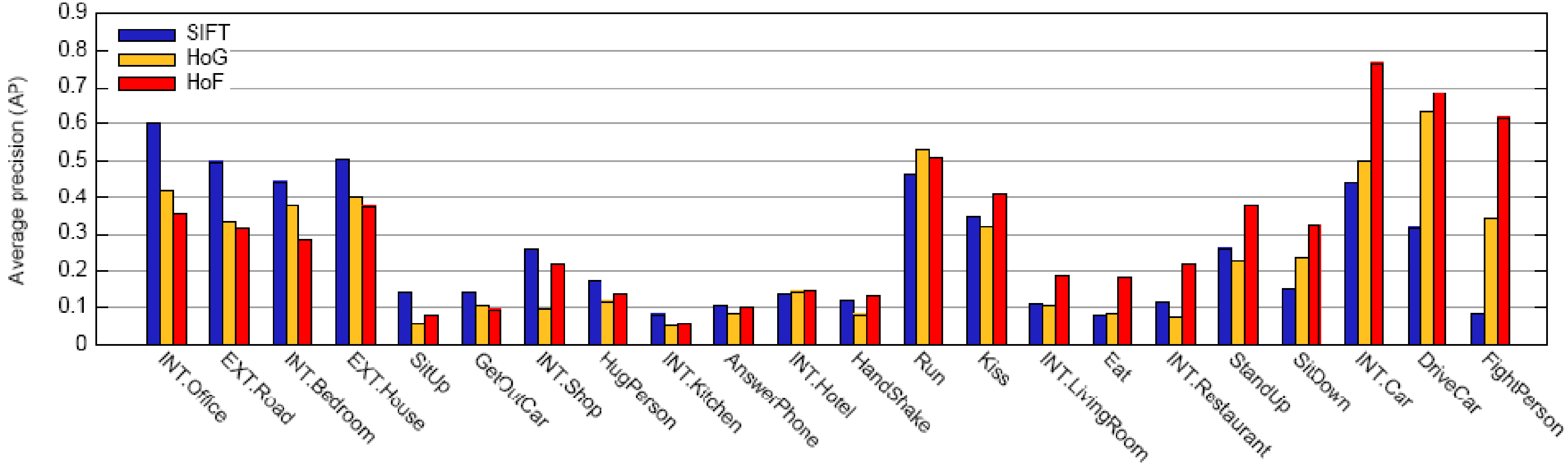
	Auto-Train-Scenes	Clean-Test-Scenes
EXT-house	81	140
EXT-road	81	114
INT-bedroom	67	69
INT-car	44	68
INT-hotel	59	37
INT-kitchen	38	24
INT-living-room	30	51
INT-office	114	110
INT-restaurant	44	36
INT-shop	47	28
All Samples	570	582

(b) Scenes

Source:
69 movies
aligned with
the scripts

Hollywood-2
dataset is on-line:
http://www.irisa.fr/vista/actions/hollywood2

Results: actions and scenes (separately)



EXT.House	**0.503**	0.363	0.491
EXT.Road	**0.498**	0.372	0.389
INT.Bedroom	**0.445**	0.362	**0.462**
INT.Car	0.444	**0.759**	**0.773**
INT.Hotel	0.141	**0.220**	**0.250**
INT.Kitchen	**0.081**	0.050	0.070
INT.LivingRoom	0.109	**0.128**	**0.152**
INT.Office	**0.602**	0.453	0.574
INT.Restaurant	**0.112**	0.103	0.108
INT.Shop	**0.257**	0.149	0.244
Scene average	*0.319*	*0.296*	*0.351*
Total average	*0.259*	*0.310*	*0.339*

	SIFT	HoG HoF	SIFT HoG HoF
AnswerPhone	**0.105**	0.088	**0.107**
DriveCar	0.313	**0.749**	0.750
Eat	0.082	**0.263**	**0.286**
FightPerson	0.081	**0.675**	0.571
GetOutCar	**0.191**	0.090	**0.116**
HandShake	**0.123**	0.116	**0.141**
HugPerson	0.129	**0.135**	**0.138**
Kiss	0.348	**0.496**	**0.556**
Run	0.458	**0.537**	**0.565**
SitDown	0.161	**0.316**	0.278
SitUp	**0.142**	0.072	**0.078**
StandUp	0.262	**0.350**	0.325
Action average	*0.200*	*0.324*	*0.326*

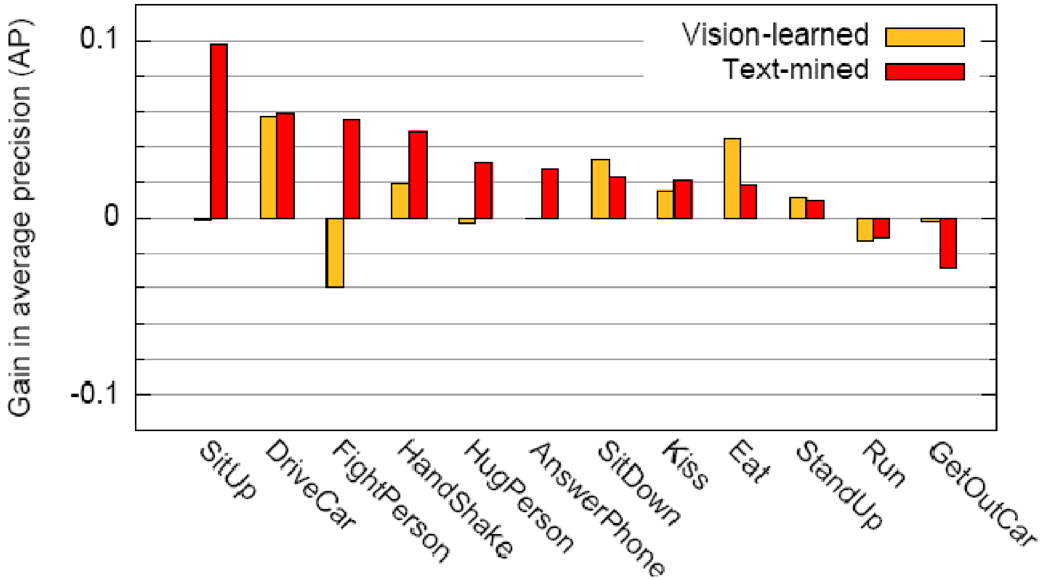
Classification with the help of context

$$a'_i(\boldsymbol{x}) = a_i(\boldsymbol{x}) + \tau \sum_{j \in \mathcal{S}} w_{ij} s_j(\boldsymbol{x})$$

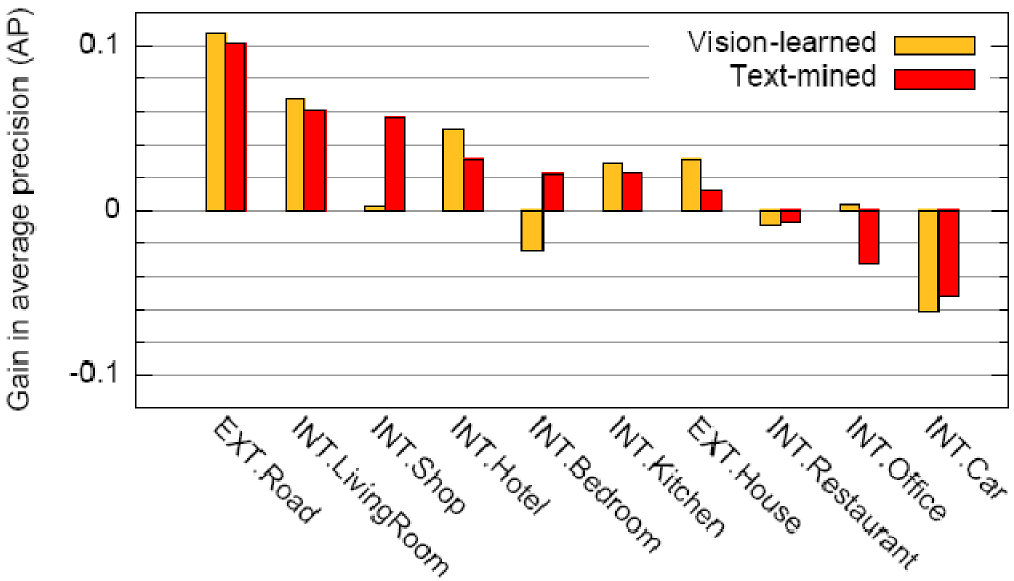
$a_i(\boldsymbol{x})$	Action classification score
$s_j(\boldsymbol{x})$	Scene classification score
w_{ij}	Weight, estimated from text: $p(Scene \mid Action)$
$a'_i(\boldsymbol{x})$	New action score

Results: actions and scenes (jointly)

Actions in the context of Scenes

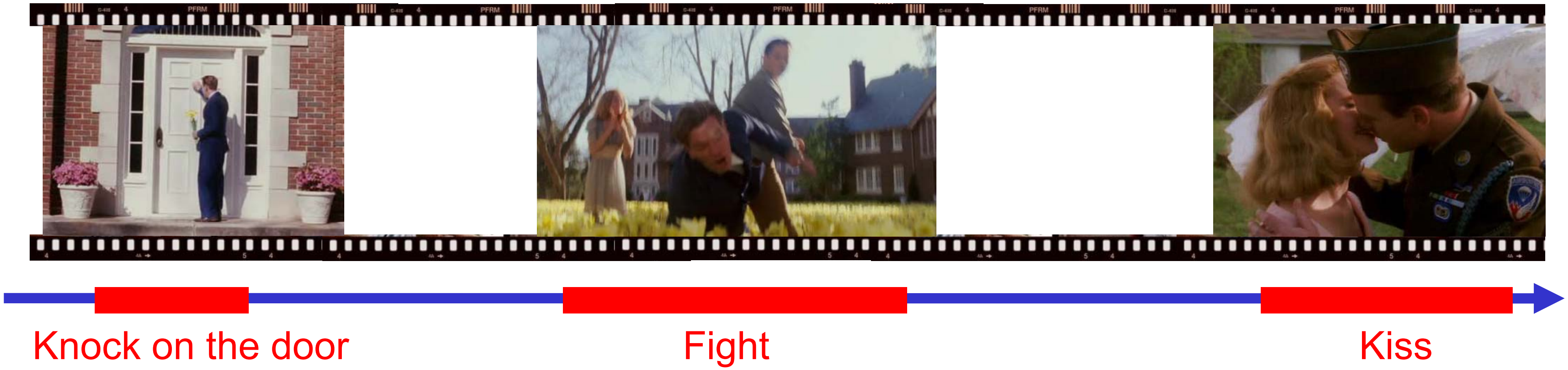


Scenes in the context of Actions



Weakly-Supervised Temporal Action Annotation

- Answer questions: *WHAT actions and WHEN they happened* ?



- Train visual action detectors and annotate actions with the minimal manual supervision

WHAT **actions?**

- Automatic discovery of action classes in text (movie scripts)

  -- Text processing:

  > *Part of Speech (POS) tagging;*
  > *Named Entity Recognition (NER);*
  > *WordNet pruning; Visual Noun filtering*

  -- Search action patterns

Person+Verb	Person+Verb+Prep.	Person+Verb+Prep+Vis.Noun
3725 /PERSON .* is	989 /PERSON .* looks .* at	41 /PERSON .* sits .* in .* chair
2644 /PERSON .* looks	384 /PERSON .* is .* in	37 /PERSON .* sits .* at .* table
1300 /PERSON .* turns	363 /PERSON .* looks .* up	31 /PERSON .* sits .* on .* bed
916 /PERSON .* takes	234 /PERSON .* is .* on	29 /PERSON .* sits .* at .* desk
840 /PERSON .* sits	215 /PERSON .* picks .* up	26 /PERSON .* picks .* up .* phone
829 /PERSON .* has	196 /PERSON .* is .* at	23 /PERSON .* gets .* out .* car
807 /PERSON .* walks	139 /PERSON .* sits .* in	23 /PERSON .* looks .* out .* window
701 /PERSON .* stands	138 /PERSON .* is .* with	21 /PERSON .* looks .* around .* room
622 /PERSON .* goes	134 /PERSON .* stares .* at	18 /PERSON .* is .* at .* desk
591 /PERSON .* starts	129 /PERSON .* is .* by	17 /PERSON .* hangs .* up .* phone
585 /PERSON .* does	126 /PERSON .* looks .* down	17 /PERSON .* is .* on .* phone
569 /PERSON .* gets	124 /PERSON .* sits .* on	17 /PERSON .* looks .* at .* watch
552 /PERSON .* pulls	122 /PERSON .* is .* of	16 /PERSON .* sits .* on .* couch
503 /PERSON .* comes	114 /PERSON .* gets .* up	15 /PERSON .* opens .* of .* door
493 /PERSON .* sees	109 /PERSON .* sits .* at	15 /PERSON .* walks .* into .* room
462 /PERSON .* are/VBP	107 /PERSON .* sits .* down	14 /PERSON .* goes .* into .* room

WHEN: Video Data and Annotation

- Want to target realistic video data
- Want to avoid manual video annotation for training

→ Use movies + scripts for automatic annotation of training samples

Subtitles

00:24:22 –> 00:24:25
– Yes, Monsieur Laszlo. Right this way.

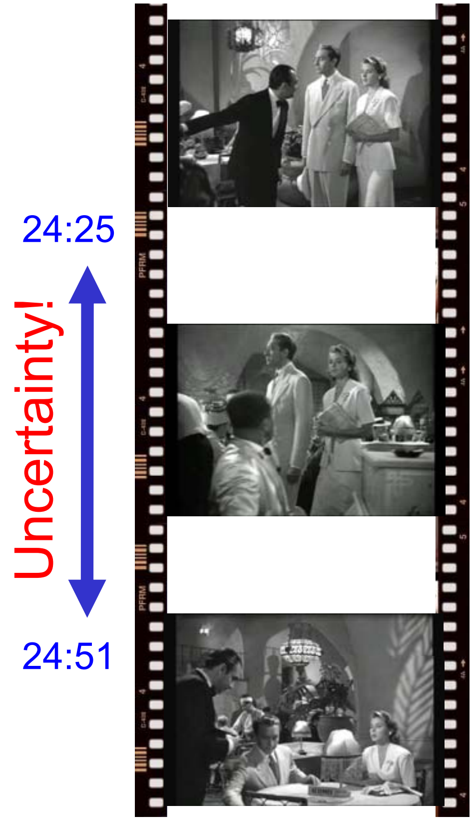
00:24:51 –> 00:24:53
Two Cointreaux, please.

Script

Speech
Monsieur Laszlo. Right this way.

Scene description
As the headwaiter takes them to a table they pass by the piano, and the woman looks at Sam. Sam, with a conscious effort, keeps his eyes on the keyboard as they go past. The headwaiter seats Ilsa...

Speech
Two cointreaux, please.



Overview

Input:

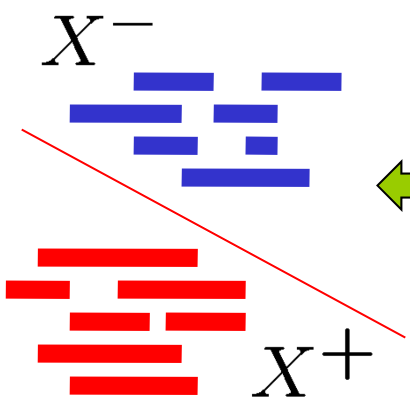
- Action type, e.g.
  Person Opens Door
- Videos + aligned scripts

Automatic collection of training clips

… **Jane** jumps up and **opens** the **door** …
… **Carolyn opens** the front **door** …
… **Jane opens** her bedroom **door** …

Clustering of positive segments

Training classifier

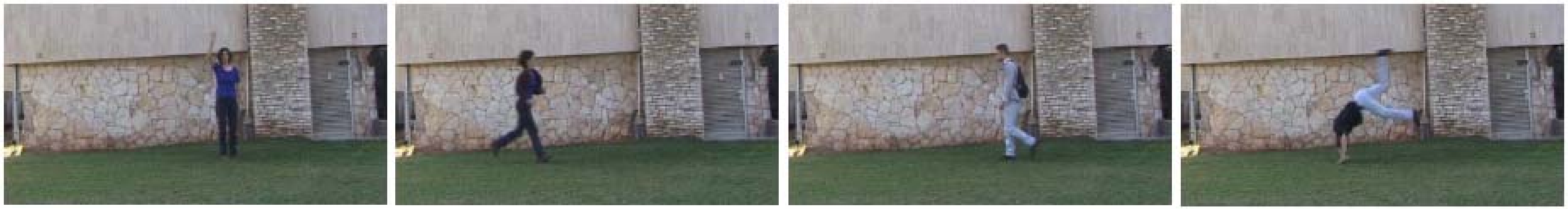


Output:

Sliding-window-style temporal action localization

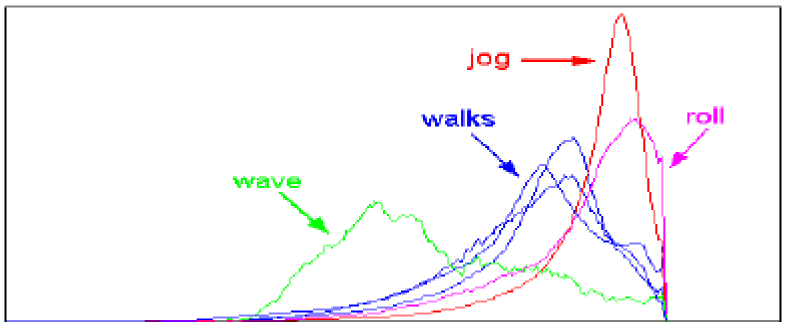
Action clustering

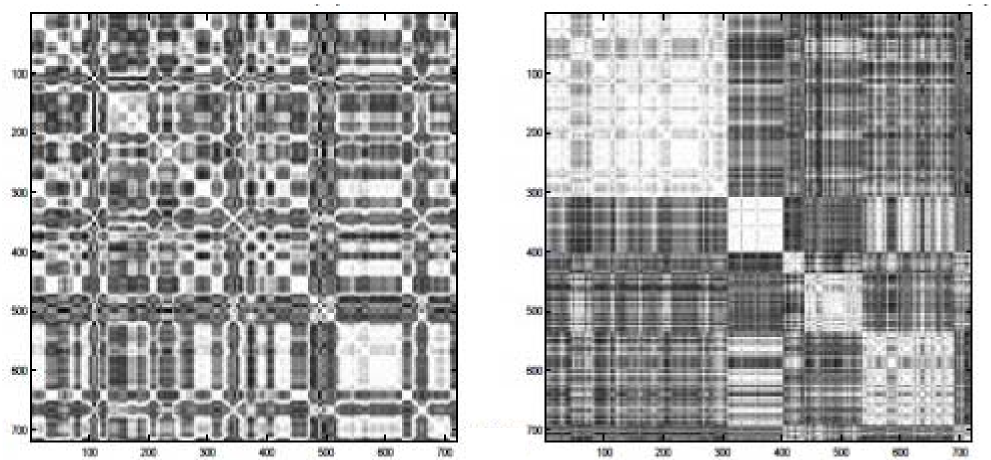
[Lihi Zelnik-Manor and Michal Irani CVPR 2001]



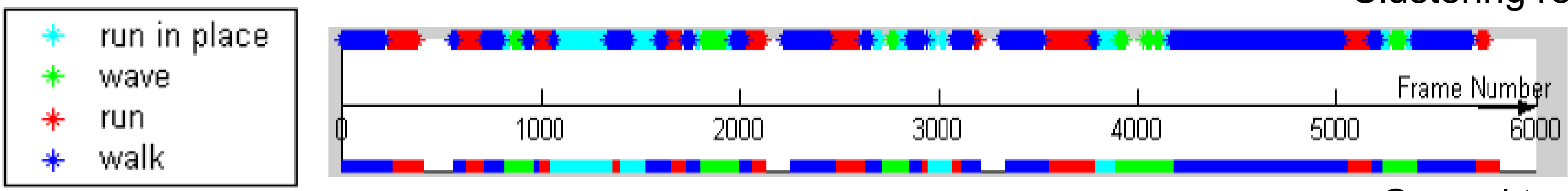
Spectral clustering

Descriptor space





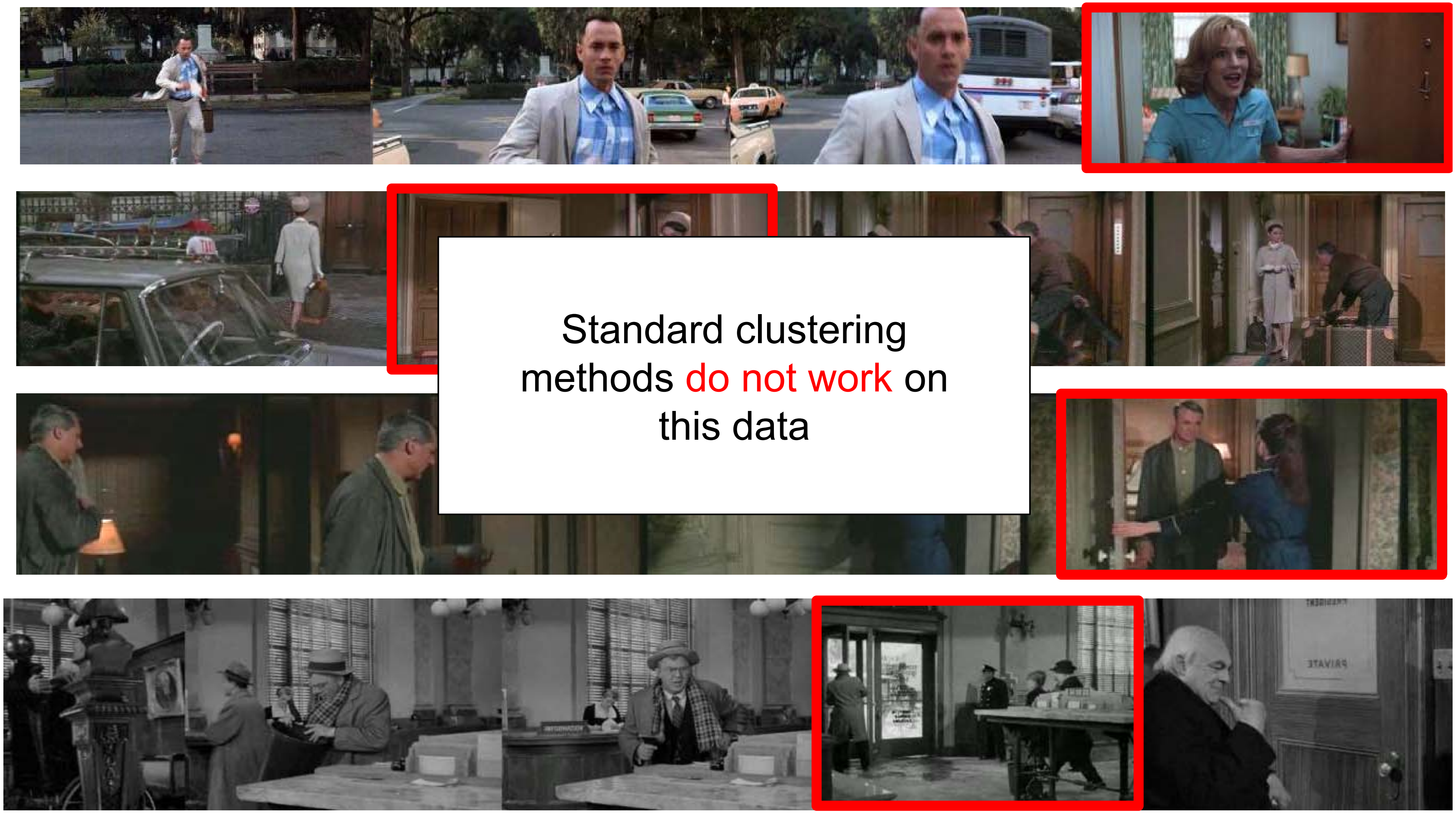
Clustering results



Ground truth

Action clustering

Complex data:

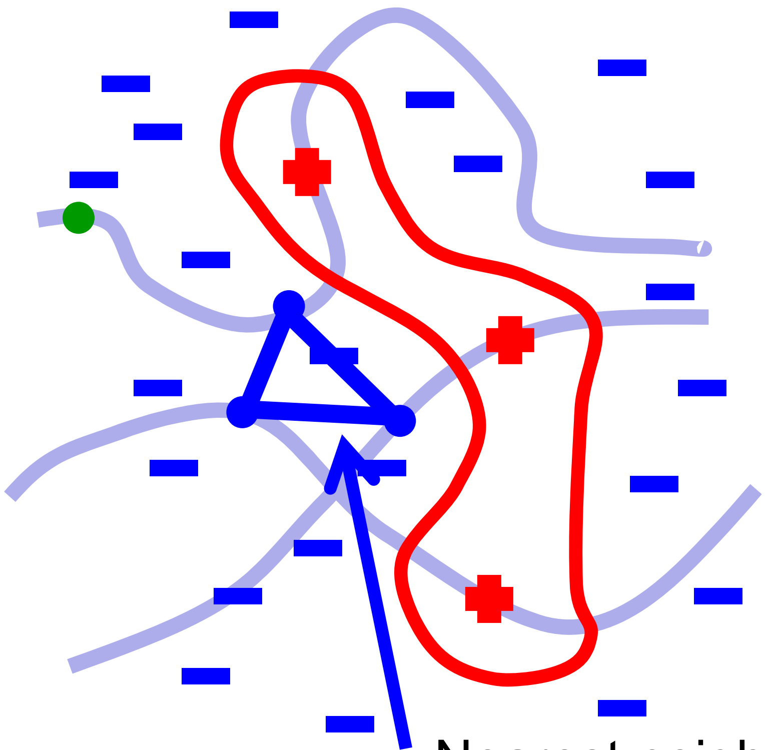


Standard clustering methods do not work on this data

Action clustering

Our view at the problem

Feature space



Nearest neighbor solution: wrong!

Video space

Negative samples!

Random video samples: lots of them, very low chance to be positives

Action clustering

Formulation

[Xu et al. NIPS'04]
[Bach & Harchaoui NIPS'07]

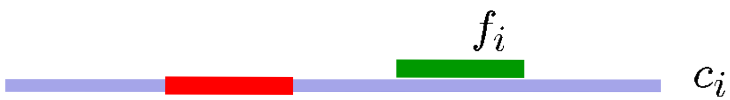
discriminative cost

$$J(f,w,b) = C_+ \boxed{\underbrace{\sum_{i=1}^{M} \max\{0, 1 - w^\top \Phi(c_i[f_i]) - b\}}_{\text{Loss on positive samples}}} +$$

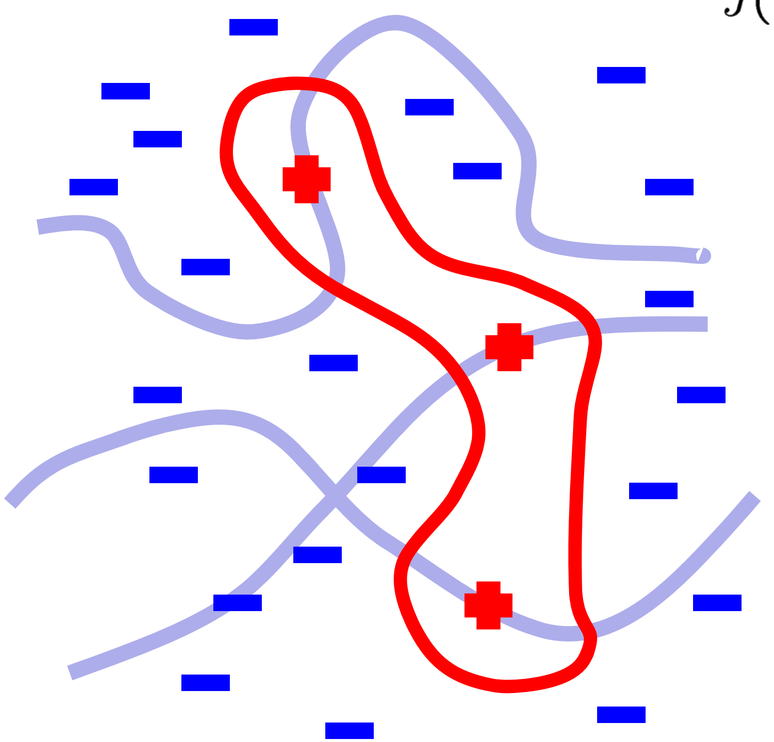
$$+ C_- \boxed{\underbrace{\sum_{i=1}^{P} \max\{0, 1 + w^\top \Phi(x_i^-) + b\}}_{\text{Loss on negative samples}}} + \|w\|^2$$

x_i^- negative samples

$c_i[f_i]$ parameterized positive samples



Feature space

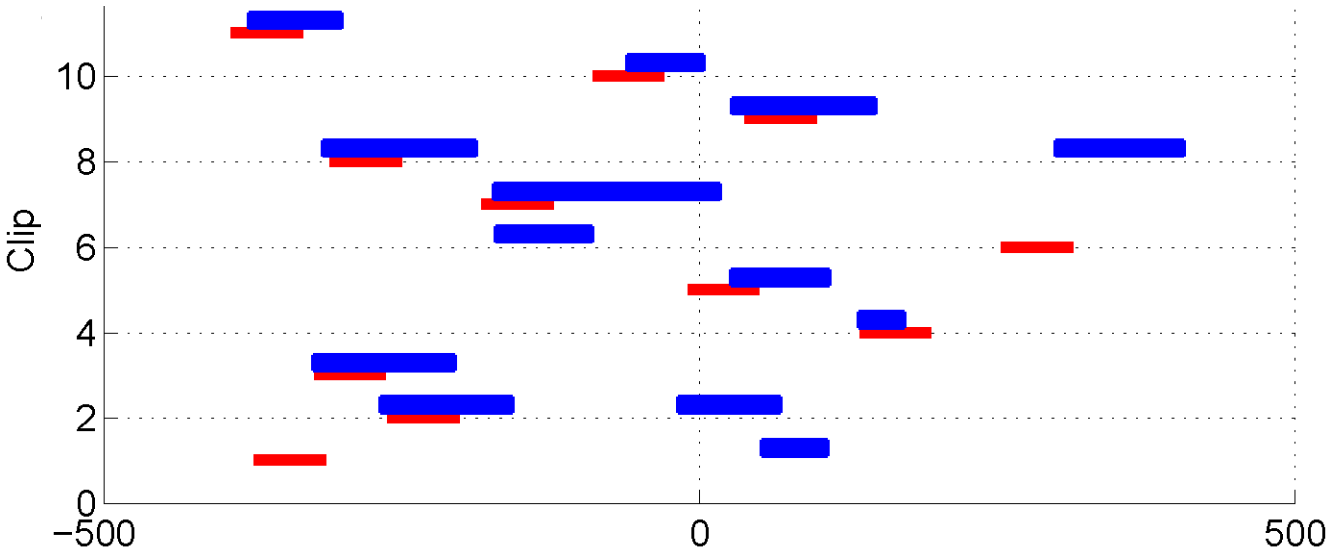


Optimization

SVM solution for w, b
Coordinate descent on f_i

Clustering results

Drinking actions in Coffee and Cigarettes

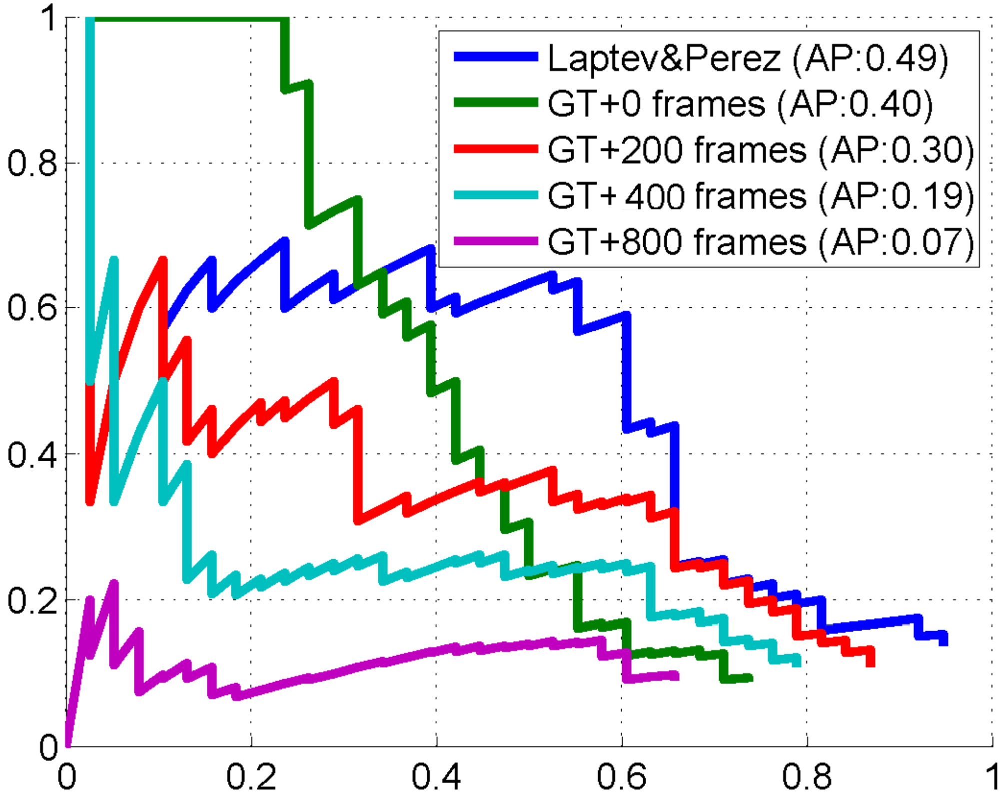


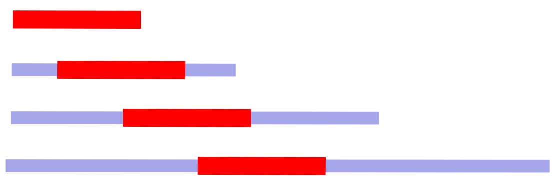
Detection results

Drinking actions in Coffee and Cigarettes

- Training Bag-of-Features classifier
- Temporal sliding window classification
- Non-maximum suppression

Detection trained on *simulated clusters*





Test set:

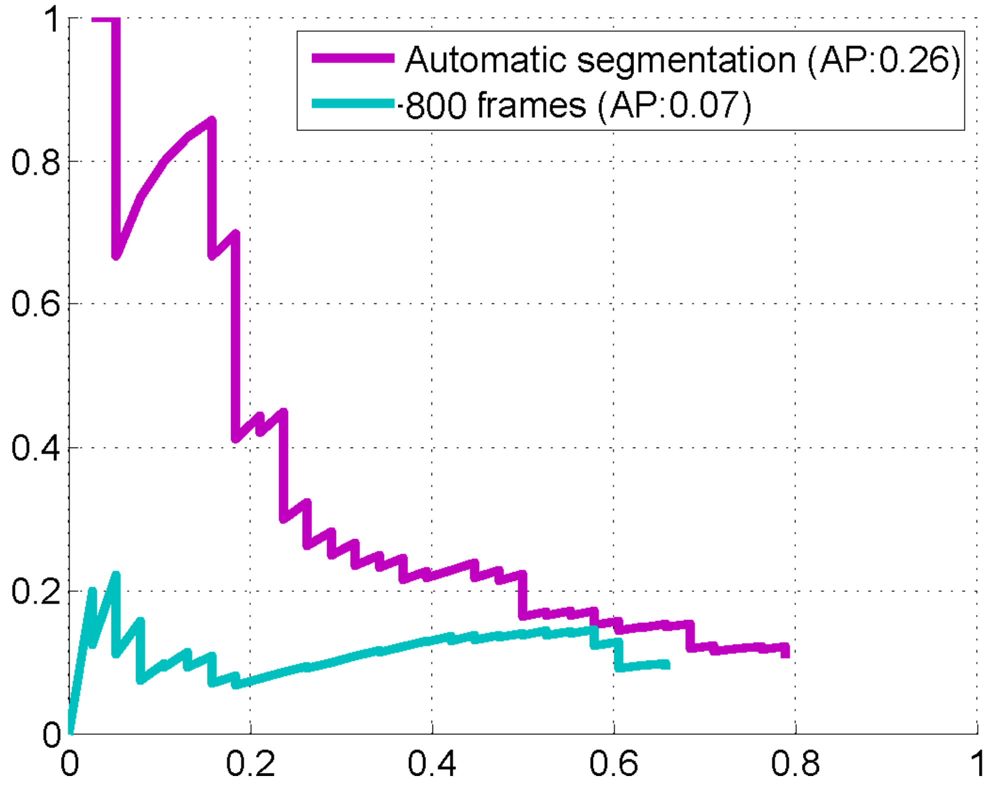
- 25min from “Coffee and Cigarettes” with GT 38 drinking actions

Detection results

Drinking actions in Coffee and Cigarettes

- Training Bag-of-Features classifier
- Temporal sliding window classification
- Non-maximum suppression

Detection trained on *automatic clusters*

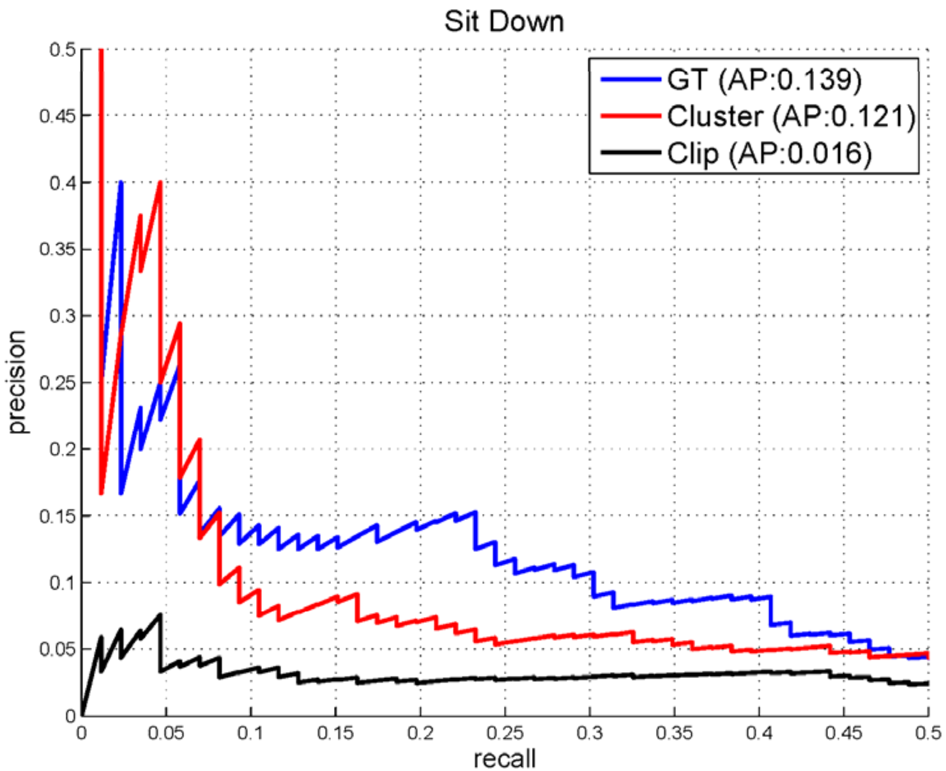


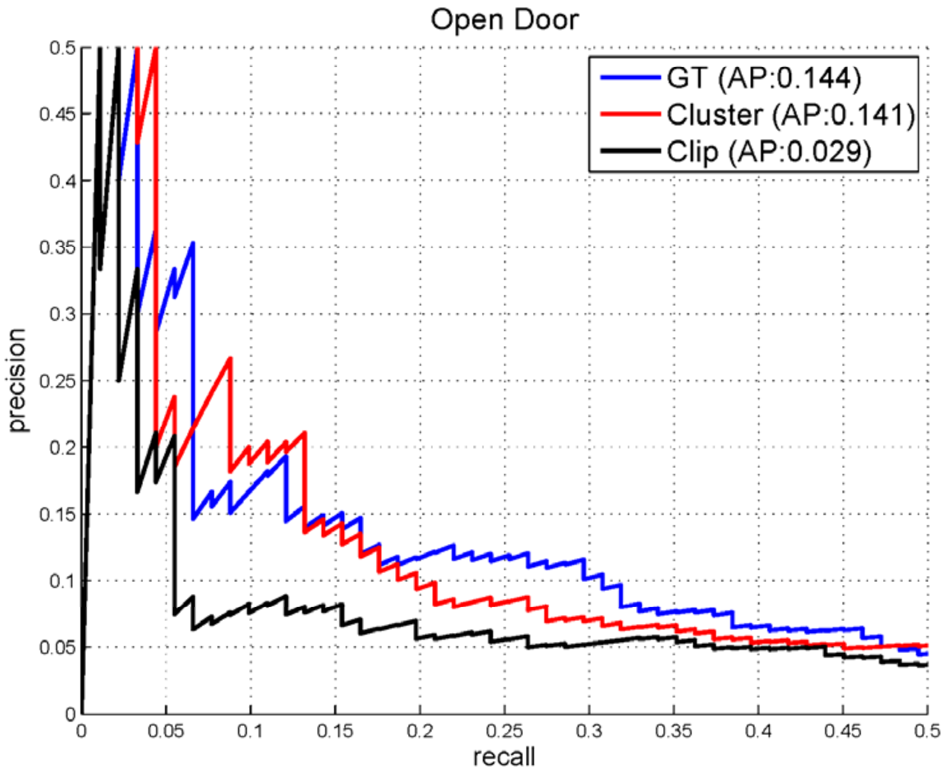
Test set:

- 25min from “Coffee and Cigarettes” with GT 38 drinking actions

Detection results

“Sit Down” and “Open Door” actions in ~5 hours of movies





Automatic Annotation of Human Actions in Video

ICCV 2009 DEMO

O.Duchenne, I.Laptev, J.Sivic, F.Bach and J.Ponce

Temporal detection of actions OpenDoor and SitDown in episodes of The Graduate, The Crying Game, Living in Oblivion

Temporal detection of “Sit Down” and “Open Door” actions in movies:
The Graduate, The Crying Game, Living in Oblivion