Advances in Visual Tracking

Ming-Hsuan Yang

Electrical Engineering and Computer Science
University of California at Merced
Merced, CA 95344
http://faculty.ucmerced.edu/mhyang

November 8, 2010

Tracking in Computer Vision

Castle

Point

Structure from motion

Stanley

Autonomous robotics

Polar Express

Motion retargeting

- Understand geometric correspondences over time
- A fundamental problem in computer vision
- A challenging and difficult task
- Numerous applications

Applications

- Motion analysis
- Sports medicine
- Animation
- Surveillance
- Autonomous robots
- Appearance modeling
- Object recognition
- Human computer interaction
- Games
- Video indexing

HCI

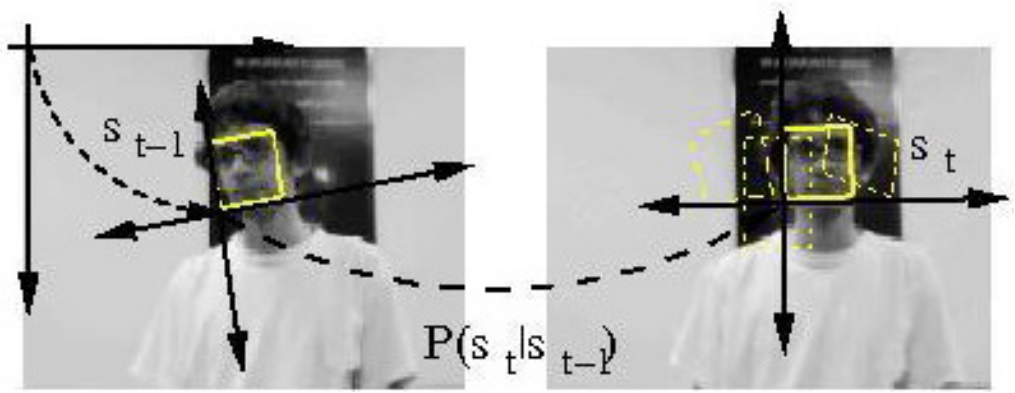
ASIMO movie

What to Track?

- High-level
  - Rigid object
    - position
    - orientation
    - bounding box or ellipse
    - motion parameters: similarity or affine transform
  - Non-rigid object
    - parts
    - pose: 2D or 3D
    - contour
    - shape deformation: thin-plate spline [Bookstein, 1989]
    - fingers, hands, etc.
- Mid-level: region, contour
- Low-level: feature

Motion Information

- How to describe the motion contents:
  - 2D/3D motion
  - position
  - scale
  - rotation
  - similarity transform
  - affine transform
  - dynamics



Feature Tracking

- Image features [Shi and Tomasi, 1994]
- Interest point operator:
  - Harris corner detector [Harris and Stephens, 1988]
  - SIFT (Scale-Invariant Feature Transform) [Lowe, 2004]
  - SURF (Speeded Up Robust Features) [Bay et al., 2006],
  - GLOH (Gradient Location and Orientation Histogram) [Mikolajczyk and Schmid, 2005]
- SIFT flow [Liu et al., 2008]
- SURFTrac [Ta et al., 2009]

Model-based Tracking

Tracking articulated objects

- Digifingers [Rehg and Kanade, 1994]
- Articulated hand tracking [Wu et al., 2001]
- Model-based 3D tracking [Lepetit and Fua, 2005]

Contour Tracking

- Snake [Kass et al., 1987]
- Active contour [Caselles et al., 1997, Isard and Blake, 1996, Cootes et al., 1998]
- Level set [Paragios and Deriche, 2000]
- Exemplar-based tracker [Toyama and Blake, 2001]

Human Tracking

- Near-view
  - 2D card board human [Ju et al., 1996] [Ioffe and Forsyth, 2001] [Cham and Rehg, 1999] [Pavlovic et al., 1999] [Hua and Wu, 2004]
  - 3D human model [Bregler and Malik, 1998] [Sidenbladh et al., 2000] [Deutscher et al., 2000] [Sminchisescu and Triggs, 2001] [Sigal et al., 2004] [Urtasun et al., 2006] [Li et al., 2006]
- Far-view
  - Pfinder [Wren et al., 1997]
  - W4 [Haritaoglu et al., 1998]
  - Multiple objects [Okuma et al., 2004] [Tao et al., 2002]

Background Subtraction

- Facilitate tracking task
- Need online update
- Representative methods:
  - Mixture of Gaussians [Stauffer and Grimson, 1999]
  - Non-parametric model [Elgammal et al., 2000] [Elgammal et al., 2002]
  - Fast Gaussian transform [Yang et al., 2004]

Looking at People and Objects

Dudek

Tracking and appearance modeling

Lee walking

3D human tracking

Tom and Jerry

Articulated object tracking

- Who, what, where, when, how?
- Detect, track, and recognize objects
- Need to account for appearance variation

Visual Tracking

- Goal:
  - Locate the object of interest
    - Object vs feature
    - Image position
    - Scale
  - Estimate object motion
    - Rigid objects
    - Non-rigid objects

David indoor

- Challenges:
  - Appearance variation due to change in illumination, view angle, shape, and by occlusions
  - Camera motion
  - Articulated objects

Figure skating

Conventional Approach

- Object representation
  - Model: Geometric/learning, 2D/3D, etc.
  - Representation: Feature, appearance, etc.
  - Invariance: Cope with variation in pose, lighting, etc.
- At time $t-1$, predict next state
  - Linear/nonlinear optimization
  - Sampling and particle filtering
- At time t, verify predictions using image observations
  - Generative model
  - Discriminative model

State of the Art: Then and Now

Isard

Condensation [ECCV96, IJCV98]

Mean

Mean shift [CVPR00, PAMI03]

Toyama

Exemplar [ICCV01, IJCV02]

Fleet

WSL [CVPR01, PAMI03]

- Most require offline training
- Most do not have high-level notion (i.e., thing) of object
- Most do not update appearance model

In this Talk

- Focus on recent advances in visual tracking
- See [Yilmaz et al., 2006] [Cannons, 2008] for surveys on object tracking
- See [Forsyth et al., 2006] [Moeslund et al., 2006] for survey on human motion
- Online visual tracking algorithms that
  - Learn and update appearance model constantly
  - Handle large illumination and pose variation
  - Operate with one moving, uncalibrated camera
  - Have real-time, robust performance
- Approach:
  - Generative algorithm
  - Discriminative algorithm
  - Multiple instance learning
  - Articulated object tracking
- Concluding remarks

Tracking: Taxonomy

- Obviously numerous ways
- High-level, mid-level, low-level
- Rigid and non-rigid object
- Single or multiple objects
- Single or multiple homogeneous/heterogeneous trackers
- Color-based or not
- Generative and discriminative
- Supervised or unsupervised
- Real-time or batch-mode
- Single or multi-view based
- Probabilistic or deterministic

Tracking: Representation

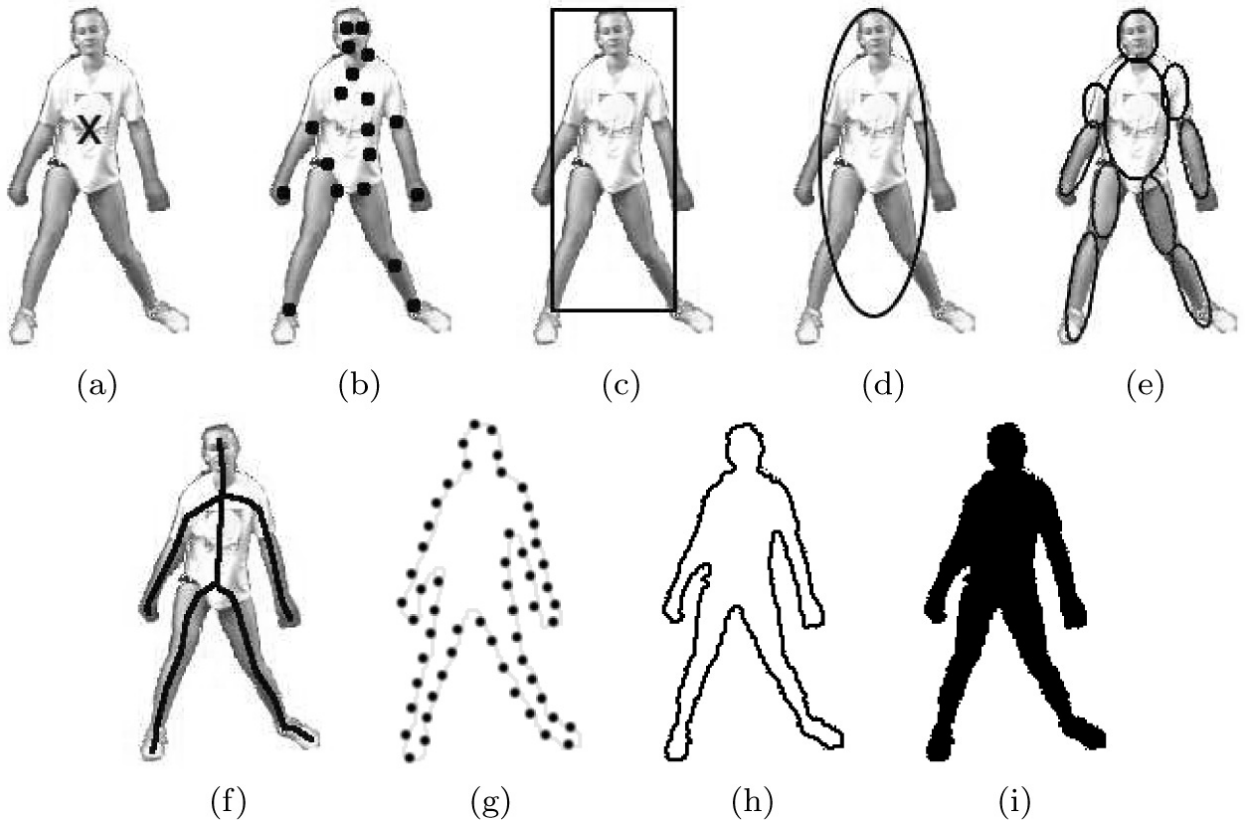


Fig. 1. Object representations. (a) Centroid, (b) multiple points, (c) rectangular patch, (d) elliptical patch, (e) part-based multiple patches, (f) object skeleton, (g) complete object contour, (h) control points on object contour, (i) object silhouette.

[Yilmaz et al., 2006]

Tracking: Prediction

- Tracking: prediction, prediction, prediction
- Kalman filter
- Maximum likelihood estimation
- Multiple hypothesis
- Non-parametric model
- Particle filter

Optical Flow I

- For a pixel at $I(x, y, t)$ that move by δx, δy, and δt between two frames,

$$I(x, y, t) = I(x + \delta x, y + \delta y, t + \delta t)$$

Assume the motion is small, expand it with Taylor series

$$I(x+\delta x, y+\delta y, t+\delta t) = I(x, y, t)+\frac{\delta I}{\delta x}\delta x+\frac{\delta I}{\delta y}\delta y+\frac{\delta I}{\delta t}\delta t+H.O.T.$$

- Assume brightness constancy, it follows

$$\frac{\delta I}{\delta x}\frac{\delta x}{\delta t} + \frac{\delta I}{\delta y}\frac{\delta y}{\delta t} + \frac{\delta I}{\delta t}\frac{\delta t}{\delta t} = 0$$

where u v are the x, y component of optical flow and $\frac{\partial I}{\partial x}$, $\frac{\partial I}{\partial y}$ and $\frac{\partial I}{\partial t}$ are derivatives of the image at (x, y, t) in the corresponding directions

Optical Flow II

- Thus,

$$uI_x + vI_y = -I_t \quad \nabla I^\top \cdot V = -I_t$$

- One equation with two unknowns, i.e., aperture problem
- Lucas-Kanade method [Lucas and Kanade, 1981]: combining information from nearby pixels (e.g., template)

$$\begin{aligned} uI_x(\mathbf{p}_1) + vI_y(\mathbf{p}_1) &= -I_t(\mathbf{p}_1) \\ uI_x(\mathbf{p}_2) + vI_y(\mathbf{p}_2) &= -I_t(\mathbf{p}_2) \\ &\vdots \\ uI_x(\mathbf{p}_n) + vI_y(\mathbf{p}_n) &= -I_t(\mathbf{p}_n) \end{aligned}$$

in matrix form, $A\mathbf{u} = \mathbf{b}$

$$A = \begin{bmatrix} I_x(\mathbf{p}_1) & I_y(\mathbf{p}_1) \\ I_x(\mathbf{p}_2) & I_y(\mathbf{p}_2) \\ \vdots & \\ I_x(\mathbf{p}_n) & I_y(\mathbf{p}_n) \end{bmatrix} \qquad \mathbf{u} = \begin{bmatrix} u \\ v \end{bmatrix} \qquad \mathbf{b} = \begin{bmatrix} -I_t(\mathbf{p}_1) \\ -I_t(\mathbf{p}_2) \\ \vdots \\ -I_t(\mathbf{p}_n) \end{bmatrix}$$

and the least squares solution is $\mathbf{u} = (A^\top A)^{-1} A^\top \mathbf{b}$

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \sum_i I_x(\mathbf{p}_i)^2 & \sum_i I_x(\mathbf{p}_i) I_y(\mathbf{p}_i) \\ \sum_i I_x(\mathbf{p}_i) I_y(\mathbf{p}_i) & \sum_i I_y(\mathbf{p}_i)^2 \end{bmatrix}^{-1} \begin{bmatrix} -\sum_i I_x(\mathbf{p}_i) I_t(\mathbf{p}_i) \\ -\sum_i I_y(\mathbf{p}_i) I_t(\mathbf{p}_i) \end{bmatrix}$$

- Horn-Schunck method [Horn and Schunck, 1981]: introduce a global smoothness constraint to solve the aperture problem

$$E = \int \int (uI_x + vI_y + I_t)^2 + \alpha^2 (|\nabla u|^2 + |\nabla v|^2) dxdy$$

where α is a regularization constant, and is minimized by

$$\begin{aligned} I_x(uI_x + vI_y + I_t) - \alpha^2 \Delta u = 0 \\ I_y(uI_x + vI_y + I_t) - \alpha^2 \Delta v = 0 \end{aligned}$$

where $\Delta = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$ is the Laplacian operator. The equations can be approximated with finite differences and solved with an iterative scheme

Optical Flow IV

- Parametric motion: Lucas-Kanade can be extended to account for other transforms (e.g., similarity, affine)
- Coarse-to-fine estimation with pyramids
- Application: motion analysis, layered motion estimation, registration, video stabilization, etc.
- Performance evaluation and database for optical flow [Barron et al., 1994] [Baker et al., 2007]
- See "Lucas-Kanade 20 years and on" [Baker and Matthews, 2004] for a discussion on alignment, warp update rule, and gradient descent approximation
- Summary:
  - Pros:
    - easy to compute
    - many applications
  - Cons:
    - difficult to handle occlusion
    - use template

Eigentracking I

- Main ideas:
  - Use eigenbasis for representation [Black and Jepson, 1998]: teat an image as "thing" rather than "stuff"
  - Use robust statistics
  - Assume subspace constancy
  - Multi-scale eigenspace
- Collect a set of images for training where each image is converted to a vector, and find the eigenvectors

$$X_{d\times n} = [\mathbf{x}_1, \ldots, \mathbf{x}_n], \quad \mathcal{C} = \frac{1}{n} X X^T, \quad \mathcal{C}\mathbf{u}_i = \lambda_i \mathbf{u}_i$$

where $\mathcal{C}$ is the covariance matrix (assume zero mean) and $\mathbf{u}_i$ is an eigenvector

Eigentracking II

- Each image can be reconstructed from a subspace spanned by a set of eigenvectors

$$\mathbf{x}_{d\times 1} \approx \tilde{\mathbf{x}}_{q\times 1} = \sum_{i=1}^{q} c_i \mathbf{u}_i$$

where c_i is computed by taking dot products of $\mathbf{x}$ and $\mathbf{u}_i$ (i.e., projection onto each eigenvector)

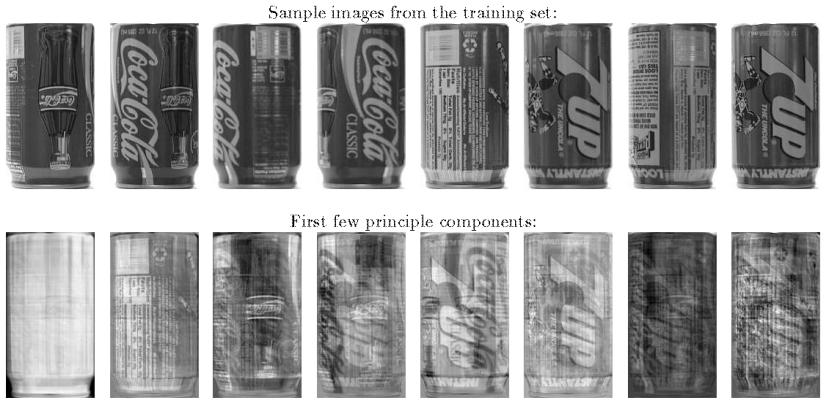
- Use singular value decomposition (SVD) to compute eigenvectors

$$X = U\Sigma V^\top,\ X = [\mathbf{x}_1, \ldots, \mathbf{x}_n],\ U = [\mathbf{u}_1, \ldots, \mathbf{u}_n]$$

where U is an orthogonal matrix whose columns are eigenvectors computed from $XX^\top$, Σ is a diagonal matrix with singular values $\sigma_1, \ldots, \sigma_n$ and V is an orthogonal matrix whose columns are eigenvectors computed from $X^\top X$

Eigentracking III

- Eigen-representation:



- With SVD, each image is represented as

$$\mathbf{x}_{d\times 1} \approx \tilde{\mathbf{x}}_{q\times 1} = \sum_{i=1}^{q} c_i \mathbf{u}_i = U\mathbf{c}$$

Eigentracking IV

- Parameterized optical flow with eigenbasis

$$I(\mathbf{x} + \mathbf{v}(\mathbf{x}, \mathbf{a})) = U\mathbf{c}(\mathbf{x}), \forall \mathbf{x}$$

where $\mathbf{v}(\mathbf{x}, \mathbf{a}) = (u(\mathbf{x}, \mathbf{a}), v(\mathbf{x}, \mathbf{a}))$ represents an image transformation and u, v represent horizontal and vertical displacements at a pixel and $\mathbf{a}$ are the motion parameters to be estimated

$$u(\mathbf{x}, \mathbf{a}) = a_0 + a_1 x + a_2 y$$
$$v(\mathbf{x}, \mathbf{a}) = a_3 + a_4 x + a_5 y$$

where c_i are parameters for affine warp

- Robust subspace constancy objective function

$$E(\mathbf{c}, \mathbf{a}) = \sum_{\mathbf{x}} \rho(I(\mathbf{x} + \mathbf{v}(\mathbf{x}, \mathbf{a})) - U\mathbf{c}(\mathbf{x}), \sigma)$$

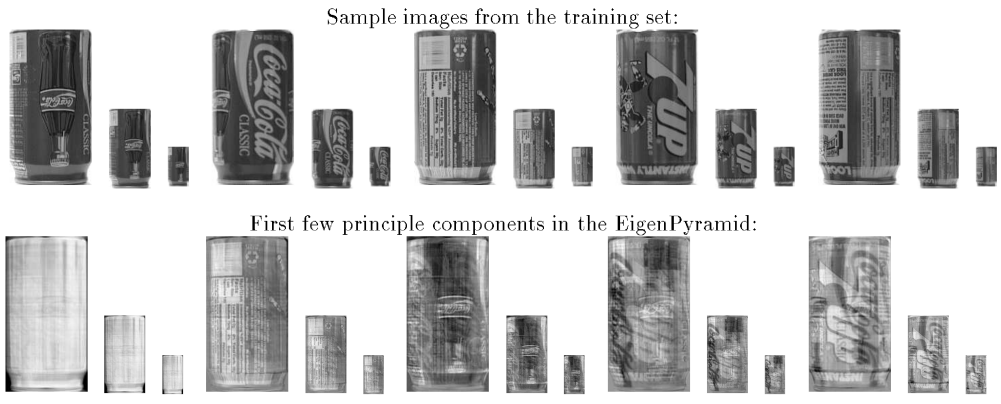
where $\rho(x, \sigma)$ is a robust norm function, $\rho(x, \sigma) = \frac{x^2}{\sigma^2 + x^2}$

Eigentracking V

- Iterative optimization
  - Fix **a**, find **c** (recognition) : matching with robust statistics
  - Fix **c**, find **a** (motion): robust regression approach for optical flow [Black and Anandan, 1996] based on

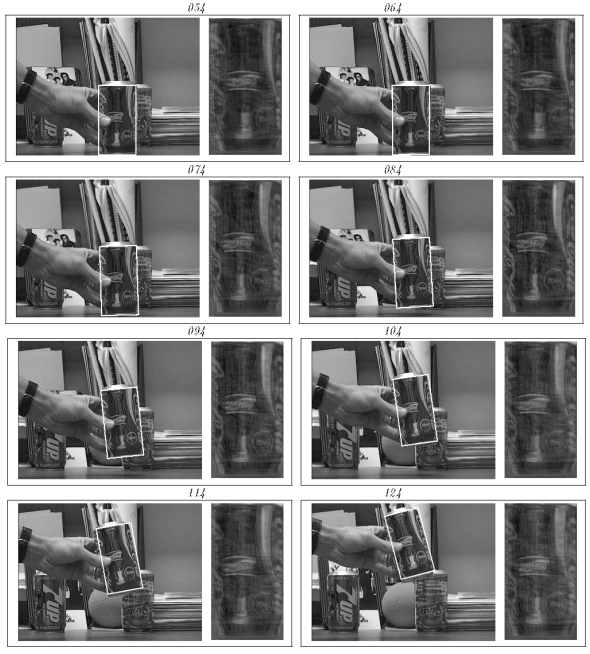
$$E(\mathbf{a}) = \sum_{\mathbf{x}} \rho(I(\mathbf{x} + \mathbf{v}(\mathbf{x}, \mathbf{a}), t) - I(\mathbf{x}, t+1), \sigma)$$

    and change $I(\mathbf{x}, t+1)$ with $U\mathbf{c}(\mathbf{x})$
  - σ is gradually reduced
- Coarse-to-fine motion estimation with multi-scale eigenspace

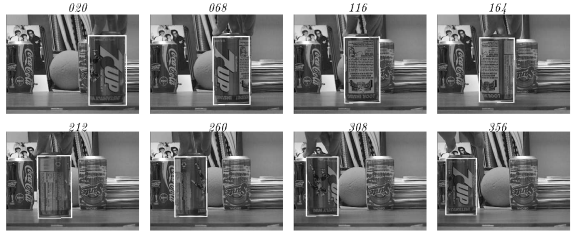


Eigentracking VI

- Experimental results



- 7Up can undergoing translation and scale change while rotating

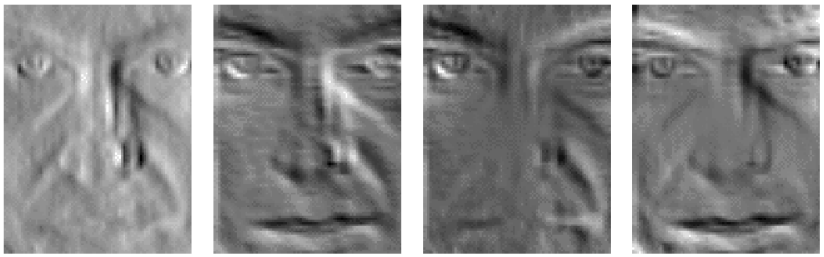


Eigentracking VII

- Summary
  - Pros:
    - tracking and recognition (“thing” vs. “stuff”)
    - subspace constancy
  - Cons:
    - iterative optimization
    - need to collect a training set at fixed views

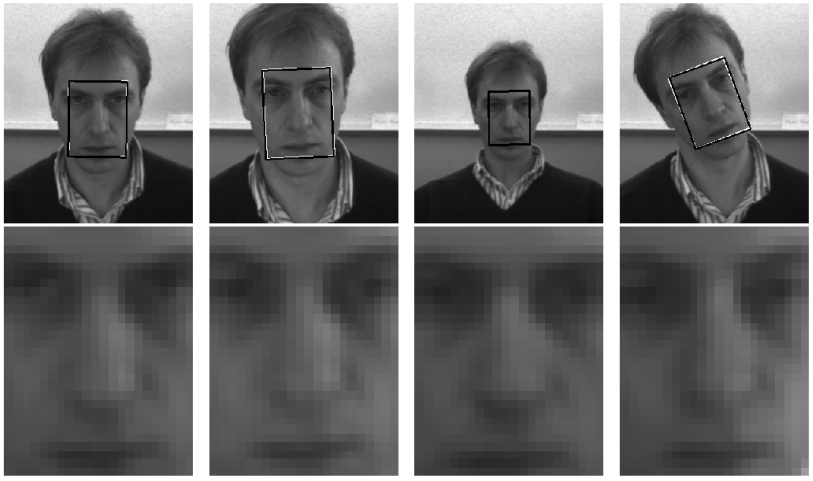
Template-based Tracking I

- Hager and Belehumer [Hager and Belhumeur, 1998]
  - efficient region tracking using parametric models of geometry and illumination
  - use a set of reference templates (e.g., basis images to account for lighting variation [Belhumeur and Kreigman, 1997])
  - efficient way to compute Jacobian matrix by factoring it into two submatrix (one involving image gradient and one motion)
  - The Jacobian matrix $\mathbf{M}(\boldsymbol{\mu}, t)$ relate variation in motion parameters to brightness values

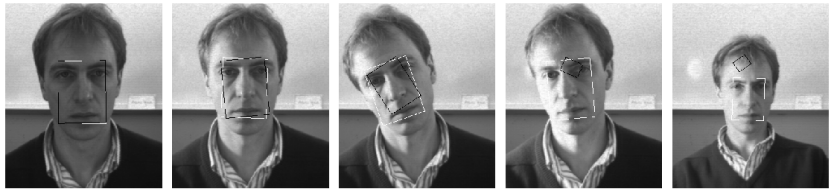


Motion template (a)(b) x, y-translation (c) rotation (d) scale

Template-based Tracking II



Without lighting change

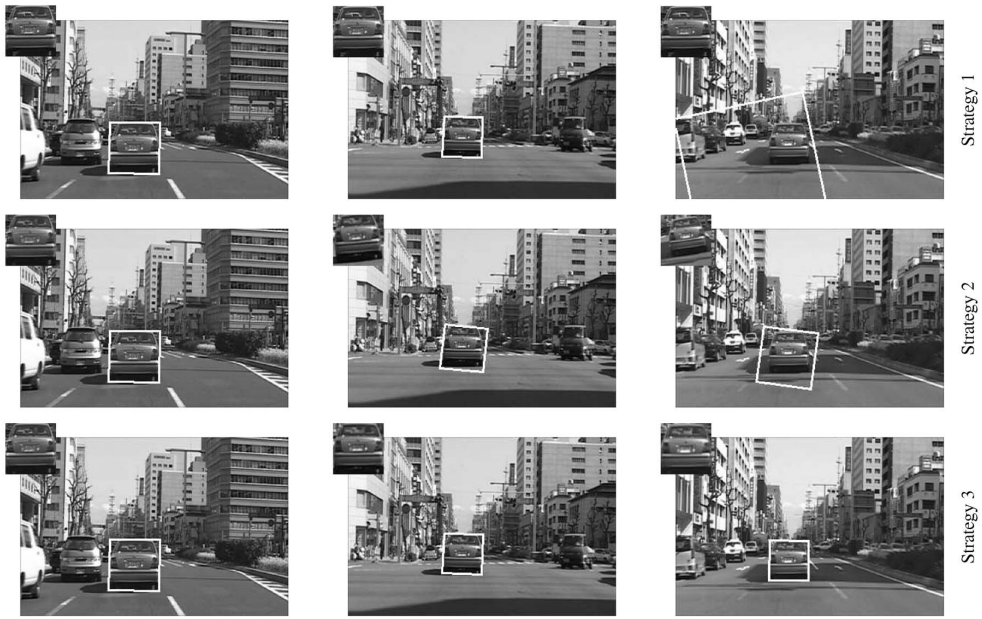


With lighting change and motion

- See an efficient direct method that computes warping parameters of thin-plate spline model for non-rigid motion [Lim and Yang, 2005]

Template-based Tracking III

- Template update problem for reducing drifts [Matthews et al., 2004]
- Use Lucas-Kanade algorithm to estimate warping parameters



(top) no update (center) update template every frame (bottom) update template every frame but use first frame for correction

- Apply similar idea for update with active appearance model

Blob Tracker I

- Birchfield [Birchfield, 1998]
  - use image gradients and color histograms
  - two modules aim to complete each other
  - enclose the head region with ellipse

$$S^* = \arg\max_{S_i \in S} \bar{\phi}_g(S_i) + \bar{\phi}_c(S_i)$$

where $\bar{\phi}_g$ and $\bar{\phi}_c$ are normalized matching scores using image gradients and color histograms

Zooming and rotation

Three people trying to steal the ellipse from the subject

Blob Tracker II

- Active blobs [Sclaroff and Isidoro, 1998]
- Spatiograms vs. histograms [Birchfield and Rangarajan, 2005]

Kernel-based Tracking I

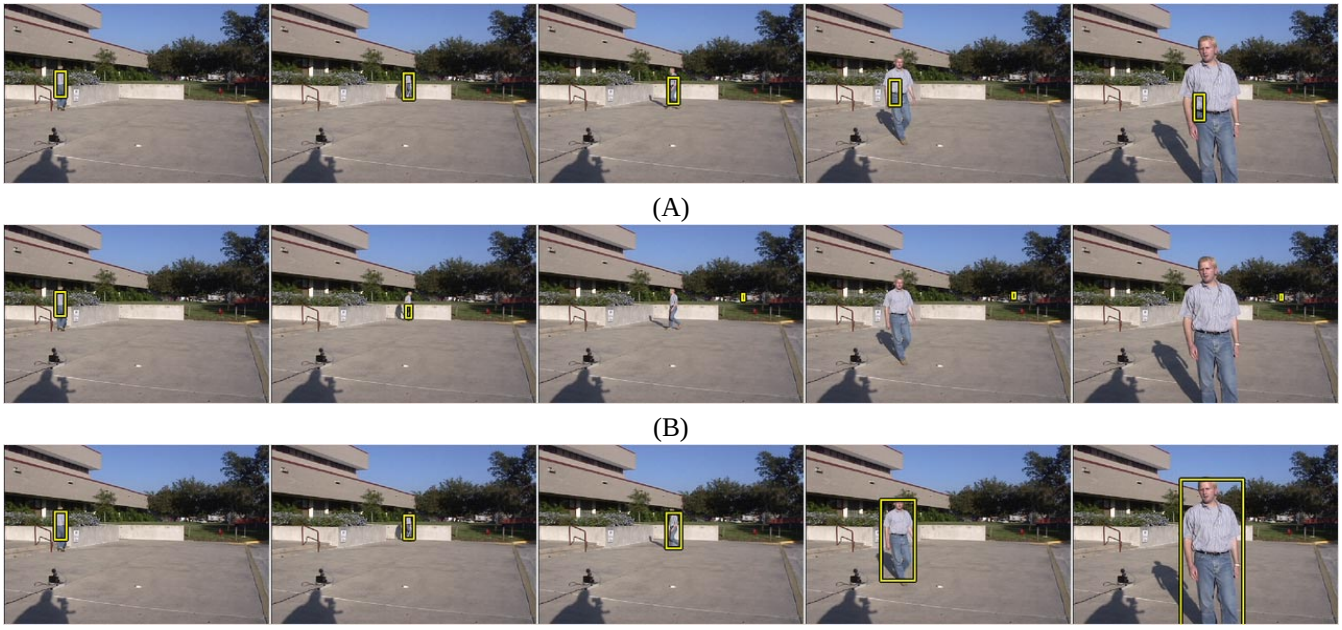
- Mean-shift tracker
  [Comaniciu et al., 2000, Comaniciu et al., 2003]
  - Non-parametric estimation with kernel density
  - Feature histogram-based representation with spatial masking and isotropic kernels
  - Gradient-descent optimization to see modes [Comaniciu and Meer, 2002]
  - Use Bhattacharyya coefficient as similarity measure
  - More effective when color features are used

football 1

football 2

Kernel-based Tracking II

- Scale-space blob [Collins, 2003]: Adapt scale-space theory [Lindeberg, 1998] with difference of Gaussian mean-shift kernel for blob tracking through scale space



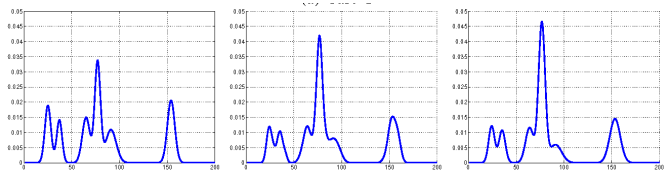
(a) no scale adaption (b) 10% scale adaption (c) scale-space blob

Sequential Kernel-based Approximation I

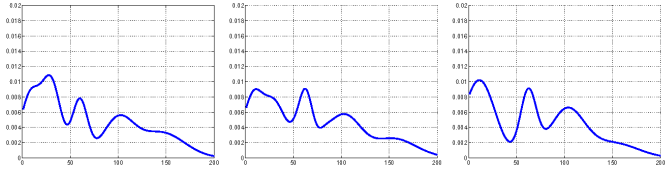
- Parametric methods such as mixture of Gaussians are often compact (with fixed number of modes) but less effective
- Non-parametric models are often flexible but memory intensive
- Idea: approximate multimodal density function with a mixture of Gaussians with kernel density approximation [Han et al., 2008]
  - modes are found by variable-bandwidth mean shift [Comaniciu, 2003]
  - covariance of each Gaussian derived by fitting the curvature around its mode
  - for tracking, use mean-shift to detect new modes with efficient sequential update

Sequential Kernel-based Approximation II

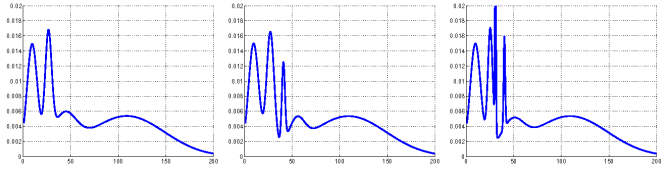
- Comparisons



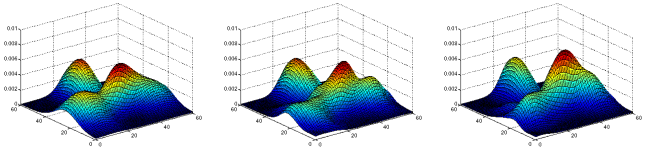
(L) original (M) kernel density estimation (R) kernel density approximation



(L) original (M) kernel density estimation (R) kernel density



EM with MoG of (L) 4 (M) 5 (R) 6 components



(L) original (M) kernel density estimation (R) kernel density approximation

WSL I

- Learning robust, adaptive appearance model [Jepson et al., 2003]
- Mixture of Gaussian at each pixel of target
  - $\mathcal{W}$: wandering motion information
  - $\mathcal{S}$: stable model
  - $\mathcal{L}$: outlier ("lost") component
- Identity stable properties of appearance and weigh them heavily for motion estimation
- Three components, $\mathcal{W}$, $\mathcal{S}$, and $\mathcal{L}$ are combined

$$p(d_t|\mathbf{q}_t, \mathbf{m}_t, d_{t-1}) = m_w p_w(d_t|d_{t-1}) + m_s p_s(d_t|\mathbf{q}_t) + m_l p_l(d_t)$$

where $\mathbf{m} = (m_w, m_s, m_l)$ are the mixing probabilities, and $\mathbf{q}_t = (\mu_{s,t}, \sigma^2_{s,t})$ contains the mean and variance of the stable component

- Online EM algorithm to adapt appearance model parameters

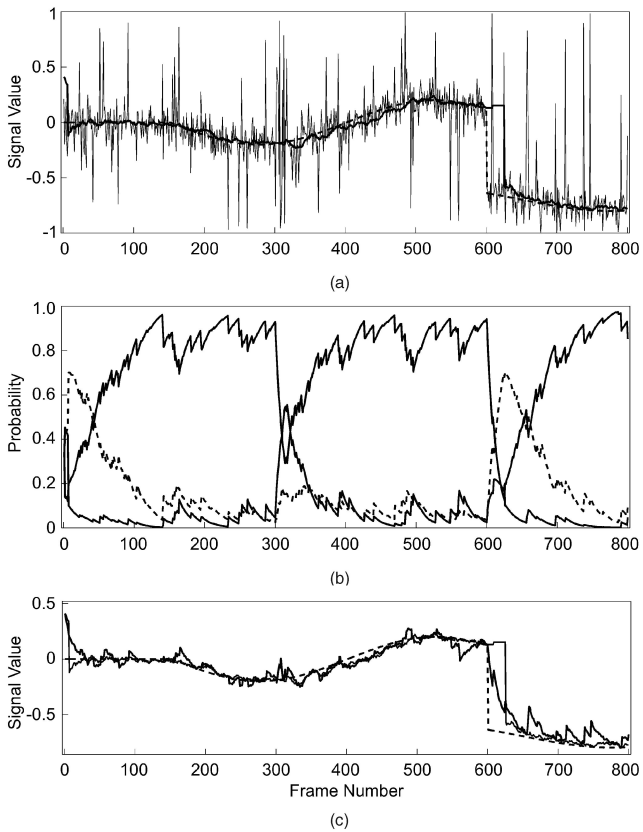
WSL II

- Wavelet-based appearance model: use phase structure of filter response
- Denote the appearance (phase) data from previous frame by $D_{t-1} \equiv \{d_{\mathbf{x},t-1}\}_{\mathbf{x}\in\mathcal{N}_{t-1}}$ Observation density for a target region $\mathcal{N}_t$ where each datum is $d_{\mathbf{x},t-1} \equiv d(\mathbf{x}, t-1)$
- With warp parameters c_t, the current data D_t is warped back to the previous frame of reference by $\hat{d}_{\mathbf{x},t} \equiv d(\mathbf{w}(\mathbf{x};\mathbf{c}_t), t)$ $D_t = \{d_{\mathbf{x},t}\}_{\mathbf{x}\in\mathcal{N}_t}$

$$L(D_t|\mathcal{A}_{t-1}, D_{t-1}, \mathbf{c}_t) =$$
$$\sum_{\mathbf{x}\in\mathcal{N}_{t-1}} \log[m_s p_s(\hat{d}_{\mathbf{x},t}|\mathbf{q}) + m_w p_w(\hat{d}_{\mathbf{x},t}|d_{\mathbf{x},t-1}) + m_l p_l]$$

where $\mathcal{A}_{t-1}$ is the appearance model at $t-1$

- Components

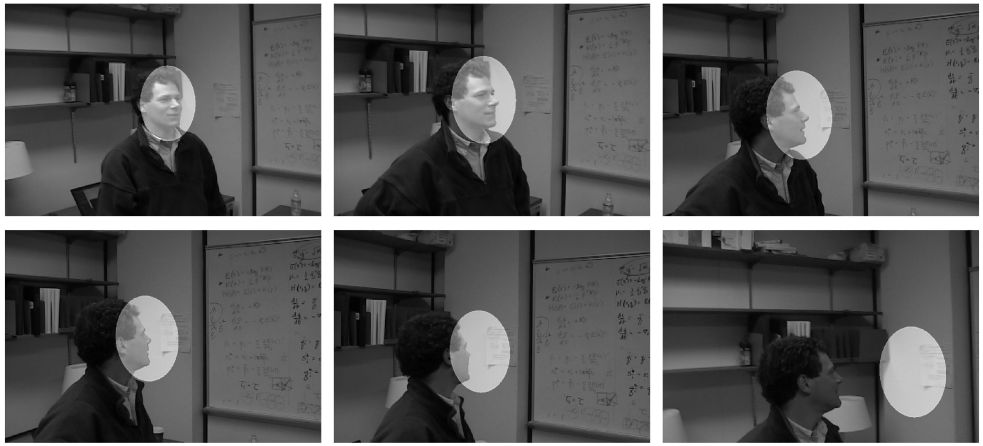


- Experimental results

WSL IV

(a) (b) (c)

- Failure case



- Summary:
  - Pros:
    - online update
    - mixture model
  - Cons:
    - large ellipse of pixels
    - adaption
    - occlusion

Kalman Filter I

- See also [Welch and Bishop, 1995] for an introductory article
- Optimal solution for linear dynamic system with Gaussian noise
- Extended Kalman filter (EKF) can handle nonlinear and non-Gaussian by linearizing the process and measurement model with first-order approximation
- Unscented Kalman filter (UKF) provides a better approximate
- Both EKF and UKF estimate and propagate a unimodal Gaussian over time

Particle Filter I

- Problem: difficult to deal with high dimensional state-space
- Extension:
  - annealed particle filter [Deutscher et al., 2000]
  - sampling methods [MacCormick and Isard, 2000] [Sullivan and Rittscher, 2001] [Sminchisescu and Triggs, 2001]
  - piecewise Gaussian [Cham and Rehg, 1999]
  - Rao-Blackwellized particle filter [Khan et al., 2004]
  - nonlinear dimensionality reduction [Lin et al., 2004]
  - density approximation [Han et al., 2009]

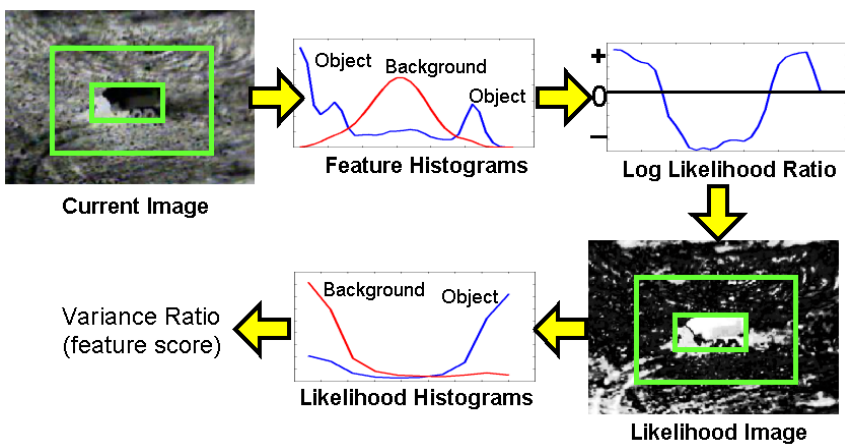
Particle Filter II

- Data association problem: When tracking multiple objects using Kalman or particle filters, one first need to associate measurement for a particular object to that object's state [Bar-Shalom, 1992]
- Two widely used methods for data association:
  - joint probability data association filtering (JPDAF) [Bar-Shalom, 1992] [Rasmussen and Hager, 2001]
  - multiple hypothesis tracking [Reid, 1979] [Cham and Rehg, 1999]

Online Feature Selection I

- Pose the tracking problem as an online feature selection problem Collins and Liu [Collins and Liu, 2003]
- Aim to find pixel-based discriminant features that best separate foreground object from the background
- Find the weight for color channels of each pixel

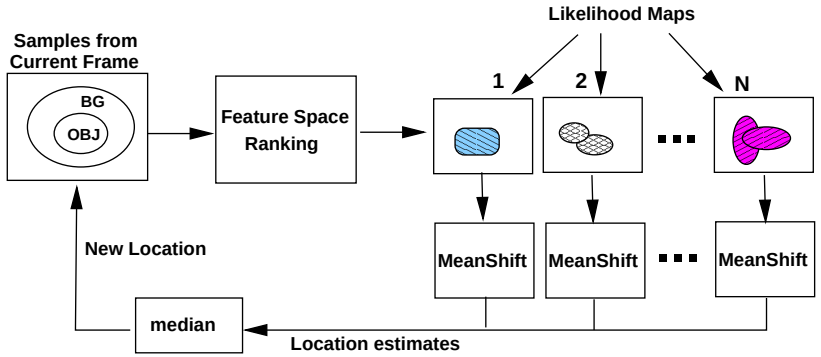
$$F_1 = \{w_1 R + w_2 G + w_3 B, w_i \in [-2, -1, 0, 1, 2]\}$$



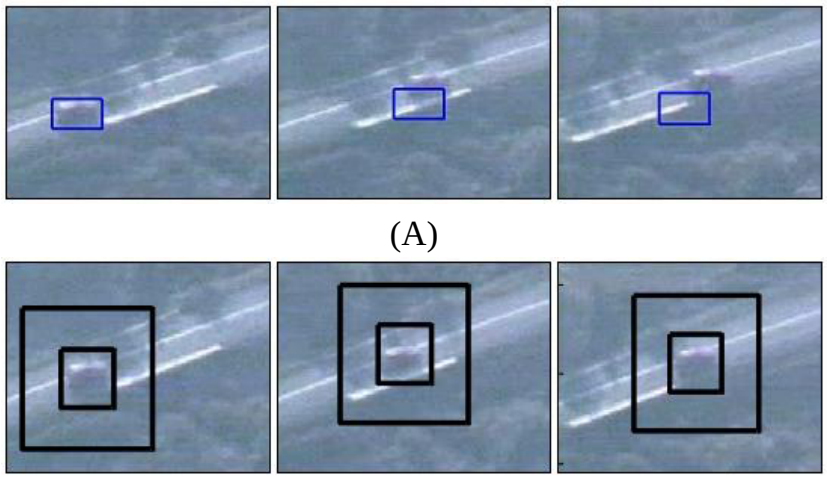
- Use linear discriminant analysis for feature selection

Online Feature Selection II

- Use mean-shift to compute 2D location



- Some results against mean-shift tracker

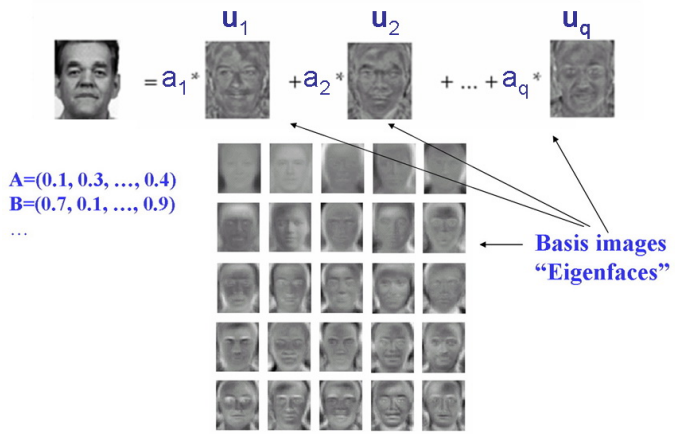


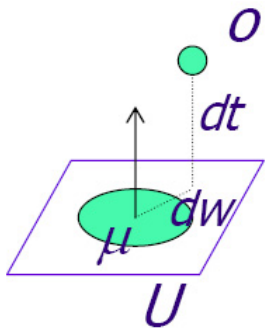
(top) mean-shift algorithm (bottom) [Collins and Liu, 2003]

Incremental Learning for Robust Visual Tracking

- Generative model
  - Learn a compact appearance model online
  - Track thing (structure information) rather than stuff (collection of pixels)
  - Simultaneously track and update appearance model
- With incremental update
  - Adapt to handle variation in lighting, pose, etc.
- Particle filter
  - Sampling rather than optimization
- Operate with moving, uncalibrated cameras with low resolution images

PCA Representation





- Recall from Principal Component Analysis (PCA),

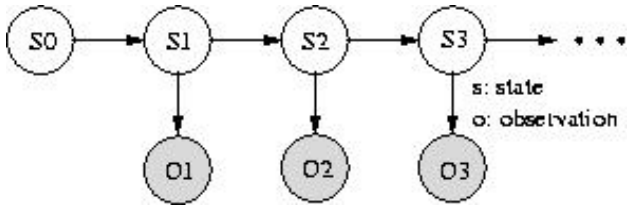
$$\mathbf{x}_{d\times 1} \approx \tilde{\mathbf{x}}_{q\times 1} = \sum_{i=1}^{q} c_i \mathbf{u}_i$$

(assume zero mean) where $\mathbf{u}_i$ are eigenvectors and λ_i are eigenvalues from covariance matrix

$$X_{d\times n} = [\mathbf{x}_1, \ldots, \mathbf{x}_n], \quad \mathcal{C} = \frac{1}{n} X X^T, \quad \mathcal{C}\mathbf{u}_i = \lambda_i \mathbf{u}_i$$

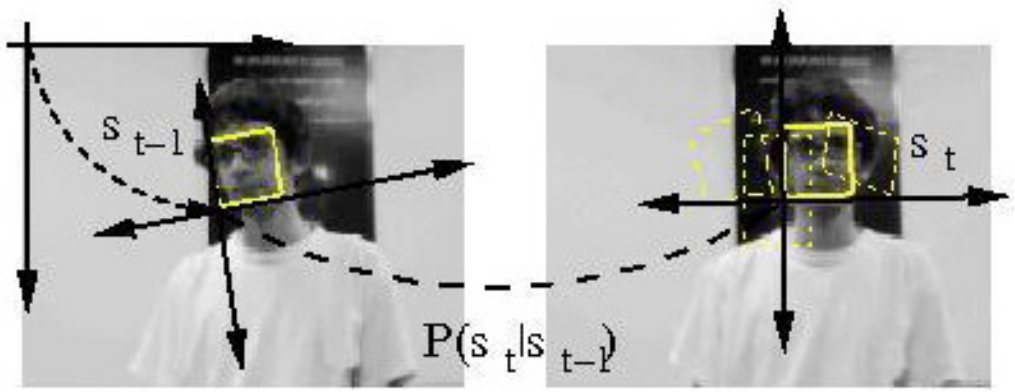
- Interpret in a generative model with probabilistic PCA, where the subspace is spanned by $\mathbf{u}_i$
- Compute with Singular Value Decomposition (SVD)

Visual Tracking as Statistical Inference



- Observation: A raster scan vector of a small image patch
- $\mathbf{o}_t$ denotes an observation at time t and $O_t = \{\mathbf{o}_1, \ldots, \mathbf{o}_t\}$ denotes a set of observations up to time t
- Assuming a Markovian state transition,
  $p(\mathbf{s}_t|O_t) = k\ p(\mathbf{o}_t|\mathbf{s}_t) \int p(\mathbf{s}_t|\mathbf{s}_{t-1}) p(\mathbf{s}_{t-1}|O_{t-1}) d\mathbf{s}_{t-1}$
  where k is a constant,
- $p(\mathbf{s}_t|\mathbf{s}_{t-1})$: Dynamic model
  Model dynamics with Brownian motion
- $p(\mathbf{o}_t|\mathbf{s}_t)$: Observation model
  Model appearance with a generative model using PCA

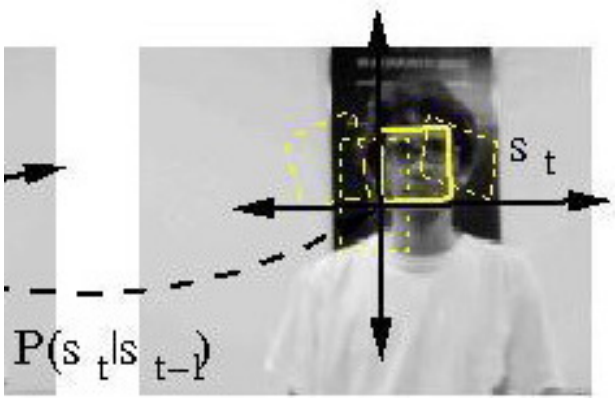
Dynamic Model: $p(s_t|s_{t-1})$

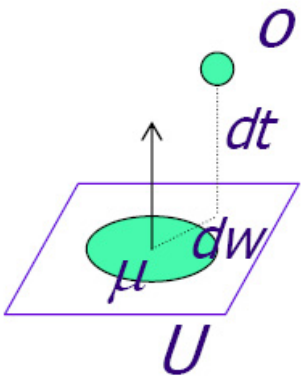


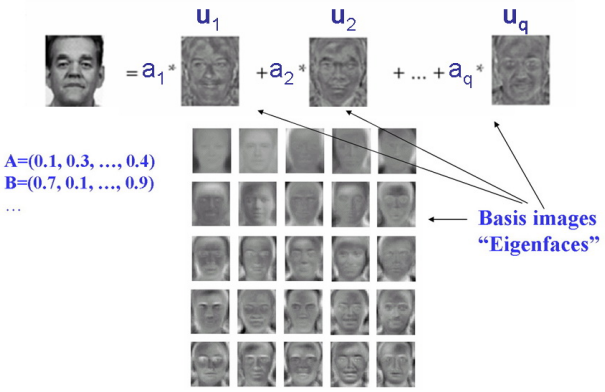
- Model object motion using similarity or affine transform
- State: $\mathbf{s}_t = [x_t, y_t, r_t, \kappa_t]$ describes the translation, rotation, and scaling in similarity transform
- State transition: Factorized Gaussians for Brownian motion, $p(\mathbf{s}_t|\mathbf{s}_{t-1}) = \mathcal{N}(x_t; x_{t-1}, \sigma_x^2)\ \mathcal{N}(y_t; y_{t-1}, \sigma_y^2)\ \mathcal{N}(r_t; r_{t-1}, \sigma_r^2)\ \mathcal{N}(\kappa_t; \kappa_{t-1}, \sigma_\kappa^2)$
- Can use other methods to learn complex motion, e.g., auto-regression and moving average (ARMA) models

Observation Model: $p(o_t|s_t)$

- Use probabilistic PCA to model image observation process
- Compute the probability of the image patch $\mathbf{o}_t$ being generated from the current eigenbasis based on distance-to-subspace, d_t, and distance-within-subspace, d_w
- $p(\mathbf{o}_t|\mathbf{s}_t) = p_{d_t}(\mathbf{o}_t|\mathbf{s}_t)\ p_{d_w}(\mathbf{o}_t|\mathbf{s}_t) = \mathcal{N}(\mathbf{o}_t; \boldsymbol{\mu}_{d_t}, UU^T + \varepsilon I)$ $\mathcal{N}(\mathbf{o}_t; \boldsymbol{\mu}_{d_w}, U\Sigma^{-2}U^T)$







Incremental Subspace Update

- Update model with new observations to account for appearance variation
- Learn a compact representation while tracking
- Almost all subspace update methods, e.g., R-SVD [Golub and Van Loan, 1996] and sequential Karhunen-Loeve [Levy and Lindenbaum, 2000], assume fixed or zero sample mean
- Propose an efficient algorithm w.r.t. running mean [Ross et al., 2008]
- See also [Hall et al., 1998]

R-SVD Algorithm

- Let old data be $X = U\Sigma V^T$ and newly arrived data be Y
- To compute SVD of $Z = [X\ Y] = U''\Sigma''V''^T$ efficiently
- Scatter matrix: $S_Z = ZZ^T = S_X + S_Y$
- Decompose Y into its projection on the subspace spanned by U and its complement, $L = U^T Y, H = Y - UL = (I - UU^T)Y$
- Let $Y = UL + JK$ where $JK = QR(H)$, $J^T J = I$
- Now, $Z = [X\ Y] = [U\ J] \begin{bmatrix} \Sigma & L \\ 0 & K \end{bmatrix} \begin{bmatrix} V & 0 \\ 0 & I \end{bmatrix}^T$
- Compute SVD of a much smaller matrix $\begin{bmatrix} \Sigma & L \\ 0 & K \end{bmatrix} = U'\Sigma'V'^T$
- Then SVD of $Z = U''\Sigma''V''^T$ where
  $U'' = [U\ J]U',\ \ \Sigma'' = \Sigma',\ \ V'' = \begin{bmatrix} V & 0 \\ 0 & I \end{bmatrix} V'$

Efficient R-SVD with Updated Mean

- Lemma: Let $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$, $Y = \{\mathbf{x}_{n+1}, \ldots, \mathbf{x}_{n+m}\}$, and $Z = \{\mathbf{x}_1, \ldots, \mathbf{x}_n, \mathbf{x}_{n+1}, \ldots, \mathbf{x}_{n+m}\}$. Denote the means and the scatter matrices of as $\boldsymbol{\mu}_X$, $\boldsymbol{\mu}_Y$, $\boldsymbol{\mu}_Z$, and S_X, S_Y, S_Z respective, then

$$S_Z = S_X + S_Y + \frac{nm}{n+m}(\boldsymbol{\mu}_X - \boldsymbol{\mu}_Y)(\boldsymbol{\mu}_X - \boldsymbol{\mu}_Y)^T$$

- Let $\hat{X} = X - \boldsymbol{\mu}_X I$, $\hat{Y} = Y - \boldsymbol{\mu}_Y I$,

$$S_Z = \begin{bmatrix} \hat{X} & \hat{Y} & \sqrt{\frac{nm}{n+m}}(\boldsymbol{\mu}_X - \boldsymbol{\mu}_Y) \end{bmatrix} \begin{bmatrix} \hat{X} \\ \hat{Y} \\ \sqrt{\frac{nm}{n+m}}(\boldsymbol{\mu}_X - \boldsymbol{\mu}_Y) \end{bmatrix}$$

- Same update formula except adding a correction term
- Important for methods that rely on sample means (e.g., Fisher linear discriminant)

Put All Together

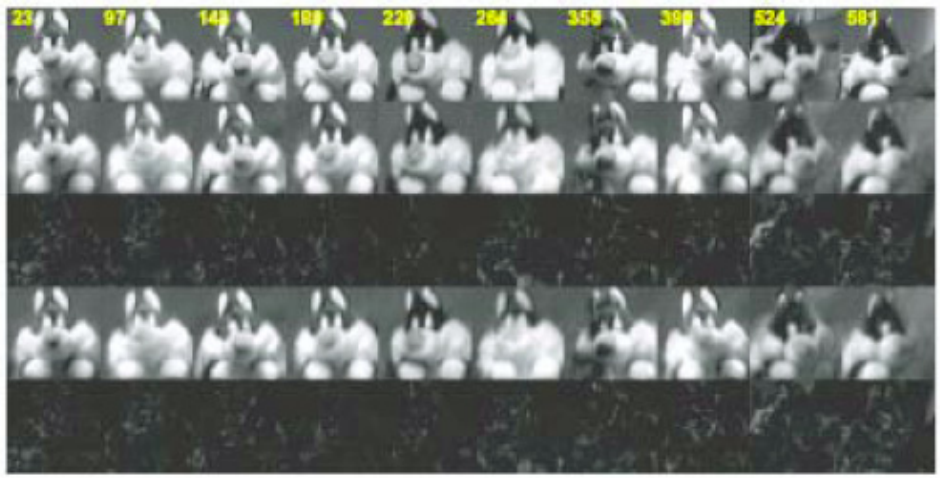
1. Initialize the location of the target
2. Draw sample state: $p(\mathbf{s}_t|\mathbf{s}_{t-1})$
3. Predict the most likely state: $p(\mathbf{s}_t|O_t)$
4. Update eigenbasis with the most likely observation
5. Goto Step 2

Experimental Setups

- Videos recorded at 15 fps with 320×240 gray scale images
- Use 6 affine motion parameters
- 15 eigenvectors
- Normalize image patches to 32×32 pixels
- 600 particles with forgetting factor of 0.95
- Update every 5 frames
- 7.5 frames per second with MATLAB and MEX implementation on a 2.8 GHz machine
- Code and data sets available on the web

Does Incremental Update Work Well?

- Compute subspace of 605 images using
  - Incremental update (every 5 frame)
  - Conventional (batch mode) PCA



tracking results

reconstruction using our method

residue: 5.65 X 10^{-2} per pixel

reconstruction using all images

residue: 5.73 X 10^{-2} per pixel

- 30% faster than another related method [Hall and Martin 02]

Experimental Results

David indoor Trellis Car

- First panel: Tracking result
- Second panel (from left to right): Mean, tracked image, residue, reconstruction results
- Third panel: Top 10 eigenvectors

Comparisons with State of the Art

David indoor Dudek Sylvester

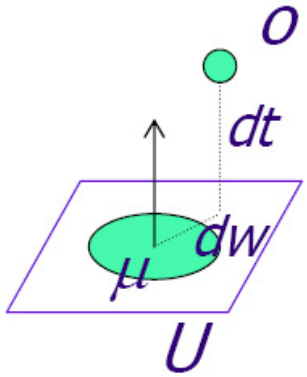
- Yellow box: Our tracker, ellipse: WSL tracker [Jepson et al., 2001], dashed green box: Mean shift tracker [Comaniciu et al., 2000]
- Simultaneously track and update appearance model
- More qualitative/quantitative comparisons in [Ross et al., 2008]
- Can be explained with
  - View-based eigenbasis for object recognition (handling pose variation) [Murase and Nayar, 1995]
  - Illumination cone (handling illumination variation) [Belhumeur and Kriegman, 1998]

Generative Model and Distance Metrics

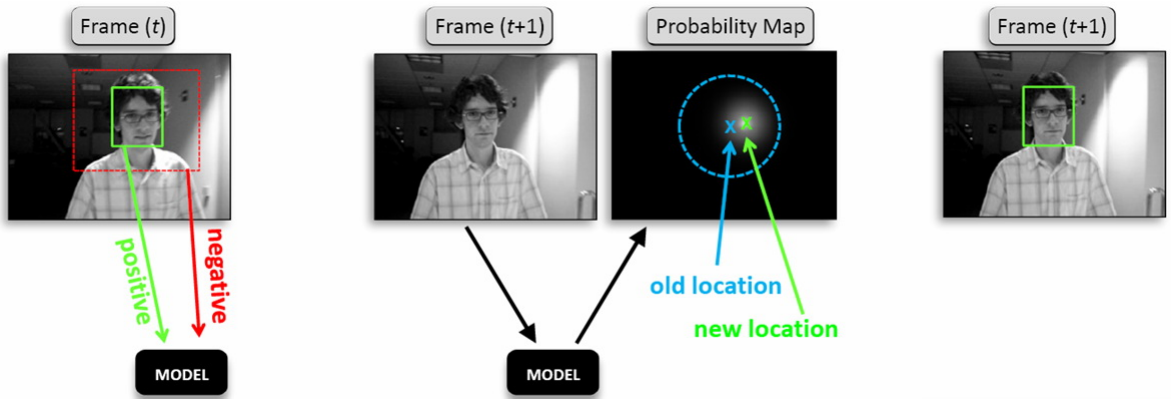
- Background patches may be confused with foreground ones
- A generative model with Gaussian noise σ
- Recall $p(\mathbf{o}_t|\mathbf{s}_t) = p_{d_t}(\mathbf{o}_t|\mathbf{s}_t)\ p_{d_w}(\mathbf{o}_t|\mathbf{s}_t) = \mathcal{N}(\mathbf{o}_t; \boldsymbol{\mu}_{d_t}, UU^T + \varepsilon I)\ \mathcal{N}(\mathbf{o}_t; \boldsymbol{\mu}_{d_w}, U\Sigma^{-2}U^T)$
- The joint log likelihood of U, $\boldsymbol{\mu}$, and σ depends on

$$\bar{\mathbf{x}}^T C^{-1} \bar{\mathbf{x}} = \underbrace{\bar{\mathbf{x}}^T U \Sigma^{-1} U^T \bar{\mathbf{x}}}_{\text{Mahalanobis distance}} + \underbrace{\frac{1}{\sigma^2} \bar{\mathbf{x}}^T (I - UU^T) \bar{\mathbf{x}}}_{\text{distance to subspace}}$$

  where $\bar{\mathbf{x}} = \mathbf{x} - \boldsymbol{\mu}$
- Small $\sigma \Rightarrow$ weigh more on distance to subspace d_t
- Large $\sigma \Rightarrow$ weigh more on Mahalanobis distance d_w



Adaptive Discriminative Generative Model



- Discriminative generative model: Find the optimal classifier V^* to separate positive/negative examples of two classes
- Sampling: Draw a set of samples that are likely to "fool" the generative model, and treat these as negative examples
- Maximizing log likelihood of (V, U, μ, and σ) where U is the orthonormal basis of the generative model [Ross et al., 2008]
- Imperative to update the means of between-scatter matrix S_b and within-scatter matrix S_w

$$V^* = \arg\max_V \frac{|VS_bV^T|}{|VS_wV^T|}$$

- Learn both appearance (i.e, U) and projection matrix (i.e., V)

Experimental Results

Pedestrian Dudek Joyce

- Use particle filter as incremental visual tracker
- First row of second panel: Positive examples
- Second row of second panel: Negative examples
- Some negative examples may be similar to positive ones

Discussion

- Subspace: Orthonormal basis can also be computed efficiently using the Gram-Schmidt algorithm
- Distance metric: Uniform ℓ_2 norm [Ho et al., 2004]
- First order vs. second order statistics for object tracking
- Better sampling scheme
- For certain applications, it suffices to locate object positions
- Treat visual tracking as a object detection problem
- Exploit ensemble of weak classifiers and local features

Support Vector Tracking I

- Integrate support vector machine (SVM) classifier with optical flow [Avidan, 2004]
- Instead of minimizing intensity difference, SVT maximizes the SVM classification score
- Given a data set $\{\mathbf{x}_i, y_i\}$ of n examples $\mathbf{x}_i$ with labels $y_i \in \{-1, +1\}$, the SVM classifier is

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

where $\mathbf{x}_j$ are support vectors, y_j is the label, and α_j are Lagrange multiplier, $k(I, x_j)$ is the kernel

Support Vector Tracking II

- Optical flow

$$I_{final} = I_{init} + uI_x + vI_y \tag{2}$$

  where I_x, I_y are the image gradients in the x, y directions, u, v are the motion parameters.

- Put these two together

$$\max \sum_{j=1}^{n} y_j \alpha_j k(I + uI_x + vI_y, \mathbf{x}_j) \tag{3}$$

- Use quadratic polynomial kernel, $k(\mathbf{x}, \mathbf{x}_j) = (\mathbf{x}^\top \mathbf{x}_j)^2$

Support Vector Tracking III

- The function can be maximized as

$$E(u,v) = \sum_{j=1}^{n} y_j \alpha_j k(I + uI_x + vI_y, \mathbf{x}_j) \quad (4)$$

$$= \sum_{j=1}^{n} y_j \alpha_j ((I + uI_x + vI_y)\mathbf{x}_j)^2 \quad (5)$$

- Taking the derivatives w.r.t. u and v

$$\frac{\partial E}{\partial u} = \sum_{j=1}^{n} y_j \alpha_j I_x^\top \mathbf{x}_j (I + uI_x + vI_y)^\top \mathbf{x}_j = 0 \quad (6)$$

$$\frac{\partial E}{\partial v} = \sum_{j=1}^{n} y_j \alpha_j I_y^\top \mathbf{x}_j (I + uI_x + vI_y)^\top \mathbf{x}_j = 0 \quad (7)$$

Support Vector Tracking IV

- After rearranging the terms

$$\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} \tag{8}$$

where

$$A_{11} = \sum_{j=1}^{n} \alpha_j y_j (\mathbf{x}_j^\top I_x)^2 \tag{9}$$

$$A_{12} = A_{21} = \sum_{j=1}^{n} \alpha_j y_j (\mathbf{x}_j^\top I_x)(\mathbf{x}_j^\top I_y) \tag{10}$$

$$A_{22} = \sum_{j=1}^{n} \alpha_j y_j (\mathbf{x}_j^\top I_y)^2 \tag{11}$$

$$b_1 = -\sum_{j=1}^{n} \alpha_j y_j (\mathbf{x}_j^\top I_x)(\mathbf{x}_j^\top I) \tag{12}$$

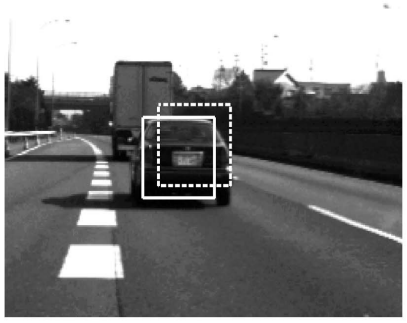
$$b_2 = -\sum_{j=1}^{n} \alpha_j y_j (\mathbf{x}_j^\top I_y)(\mathbf{x}_j^\top I) \tag{13}$$

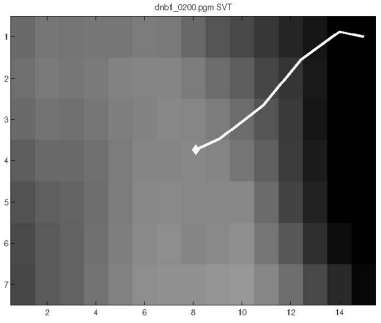
Support Vector Tracking V

- Resemble the standard optical flow equations
- Support vectors replace the role of the second image
- All computations are done on a single frame (not on a pair of successive frame)
- Similar to optical flow, large motion is handled by pyramid
- Train on a set of 10,000 images of vehicles (sedans, SUVs, trucks) and non-vehicles
- Each example is normalized to the size of 20×20 pixels and about 2,000 support vectors are extracted
- Speed up the classification with a reduced set method with 400 SVs

Support Vector Tracking VI

- Error surface

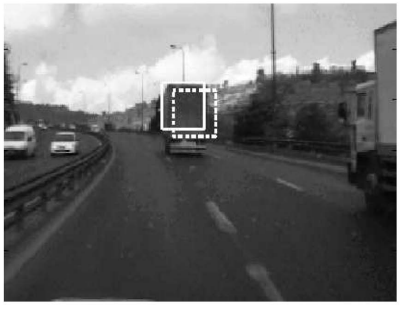


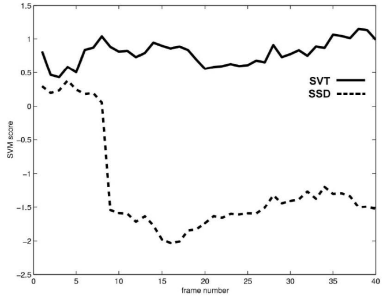


- Comparison

(a)

(b)



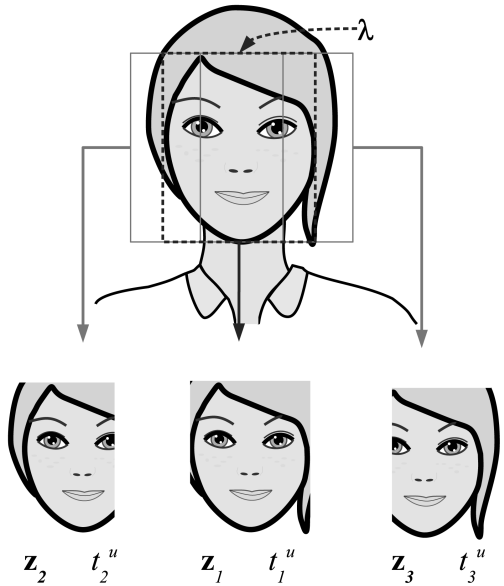


Support Vector Tracking VII

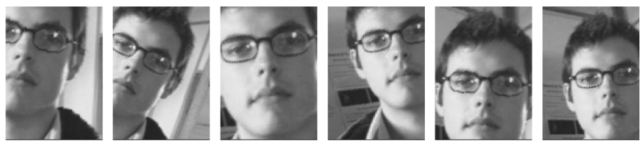
- Summary:
  - Pros:
    - integrate SVM classifier into visual tracking
    - more robust than template-based visual tracking
  - Cons:
    - need to collect a large training set
    - handle simple motion model
    - does not handle occlusion

Relevance Vector Machine I

- Pose the tracking problem as a regression problem [Williams et al., 2005]
- RVM: a probabilistic sparse SVM
- Training



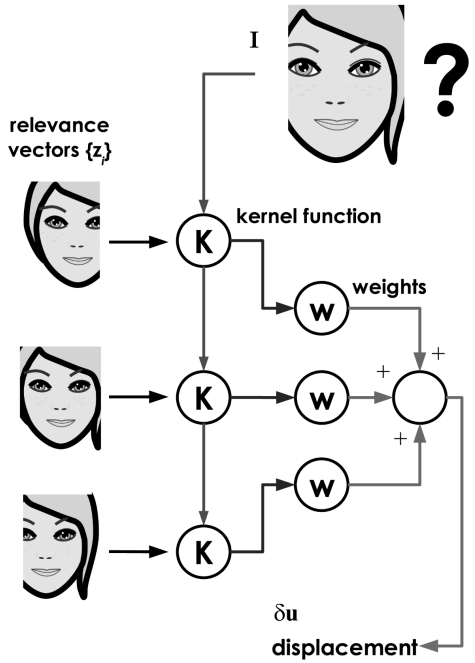
(a)



(b)

Relevance Vector Machine II

- Learn the displacement expert with RVM regression
- Test



- Works most effective with in-plane image transformation

Tracking by Detection I

- Formulate the tracking problem as a detection
- Multi-target detection and tracking [Okuma et al., 2004]
- Learn foreground/background classifiers
- Collins [Collins and Liu, 2003]
- Avidan [Avidan, 2007]
- Grabner [Grabner and Bischof, 2006a]
- See also P-N Learning [Kalal et al., 2010]

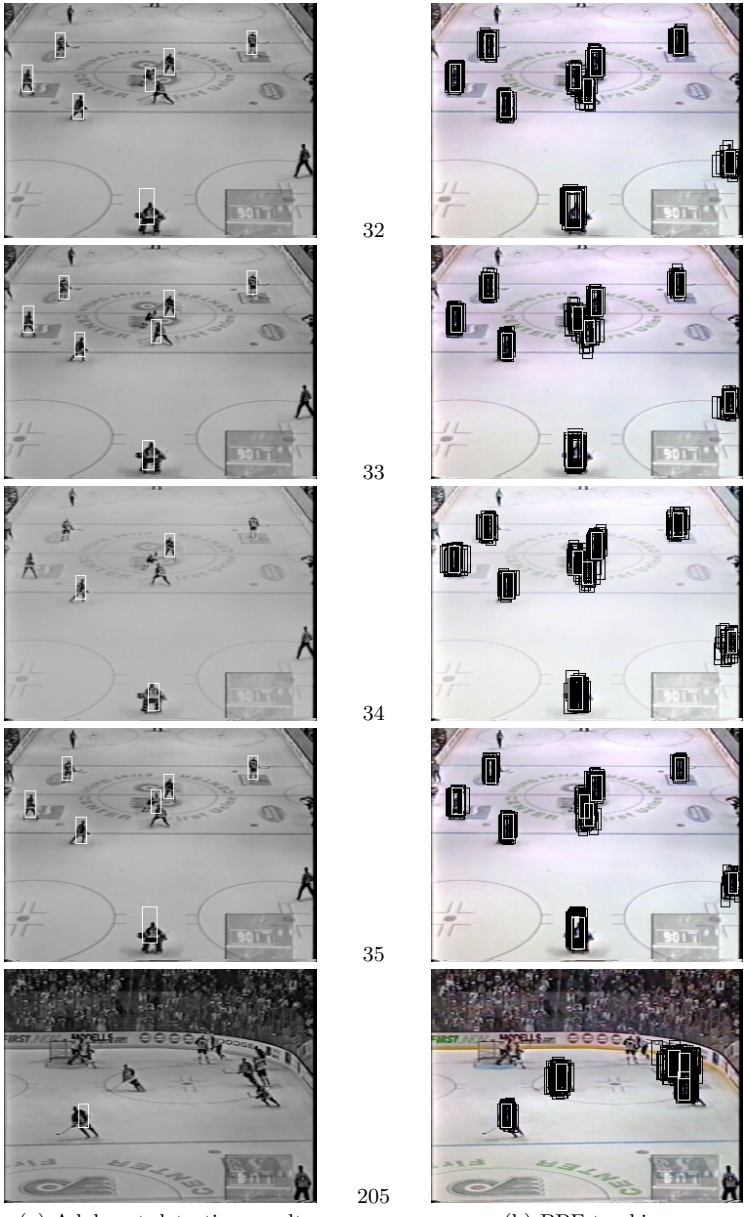
Tracking and Detection I

- Track varying number of non-rigid objects
- Two main components:
  - mixture of particle filters
  - Adaboost-based object detection

(a) (b) (c)

- Numerous false positives from detection
- Detection helps tracking and vice versa

Tracking and Detection II

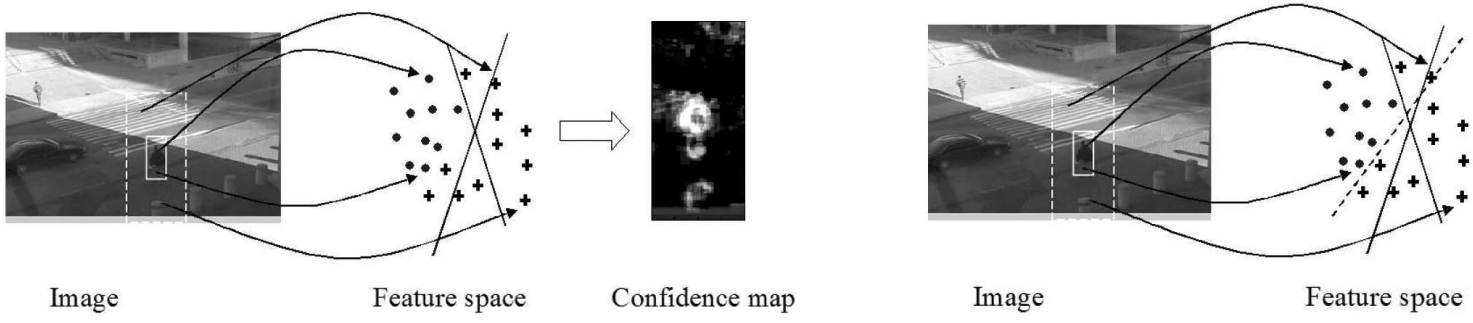


(a) Adaboost detection result

(b) BPF tracking

Ensemble Tracking I

- Train an ensemble of weak classifiers online using Adaboost [Avidan, 2007]
- Each pixel is classified to belong to the object or background, thereby giving a confidence map
- The peak of the map is found using mean-shift



Ensemble Tracking II

- Weak classifier h: represent each pixel as a d-dimensional feature, x_i with its label y_i, using weighted least squares regression

$$A\mathbf{x} = \mathbf{y} \qquad h = (A^\top A)^{-1} A^\top \mathbf{y} \tag{14}$$

$$WA\mathbf{x} = W\mathbf{y} \quad h = (A^\top W^\top WA)^{-1} A^\top W^\top W\mathbf{y} \tag{15}$$

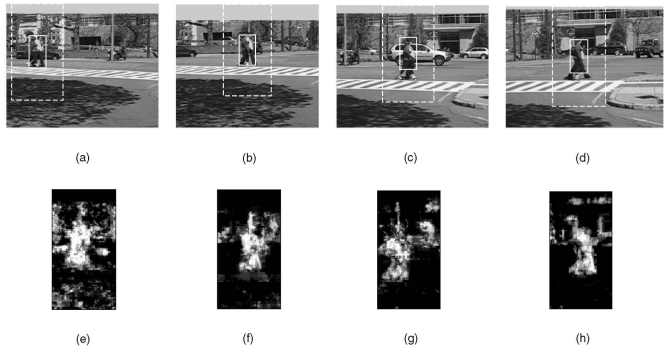
where each row of A is $\mathbf{x}_i$ and W is a diagonal weight matrix

Ensemble Tracking III

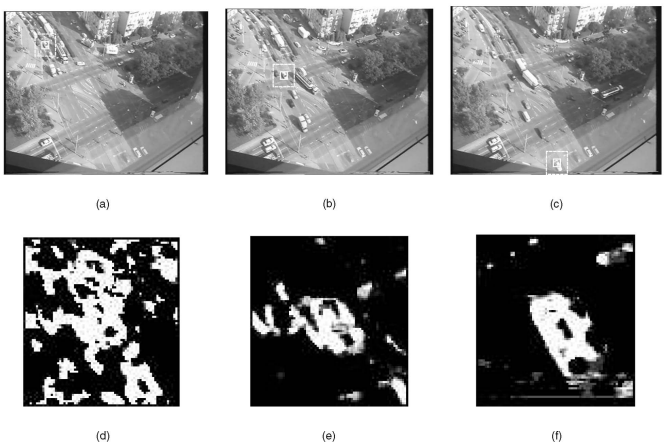
- General ensemble tracking
  Initialization:
  Train T weak classifiers and add them to the ensemble
  For each new frame I_j do:
  - test all pixels using the current strong classifier and create a confidence map L_j
  - run mean shift on the confidence map and report new object rectangle r_j
  - label pixels within r_j as object and all those outside as background
  - keep K "best" weak classifiers
  - train new $T - K$ weak classifiers on frame I_j and add them to the ensemble
- Use Adaboost to construct a boosted classifier

Ensemble Tracking IV

- Using 11-d feature vector (8-bin local histogram of oriented gradients and RGB)



- Using 9-d feature vector (8-bin local histogram of oriented gradients and intensity)



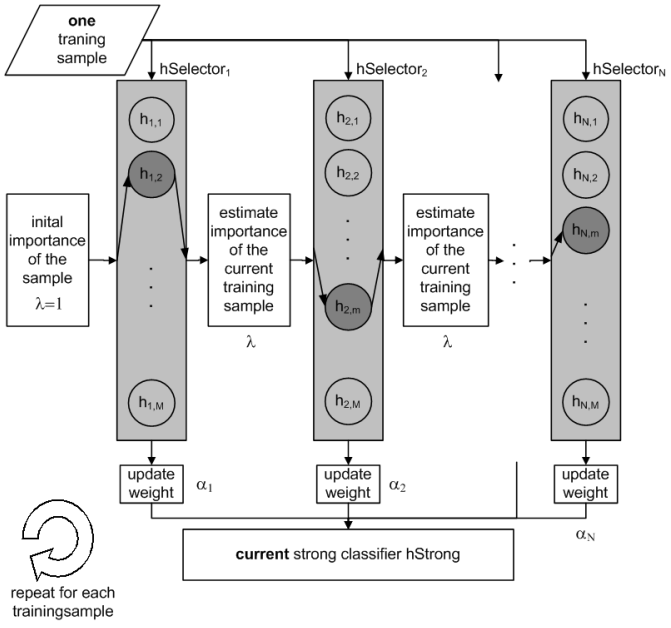
Ensemble Tracking V

- Summary:
  - Pros:
    - tracking as online classification/detection problem
    - online update
    - handle partial occlusion
    - work on gray scale images
  - Cons:
    - treat the target as a bag of pixels
    - drifting problem

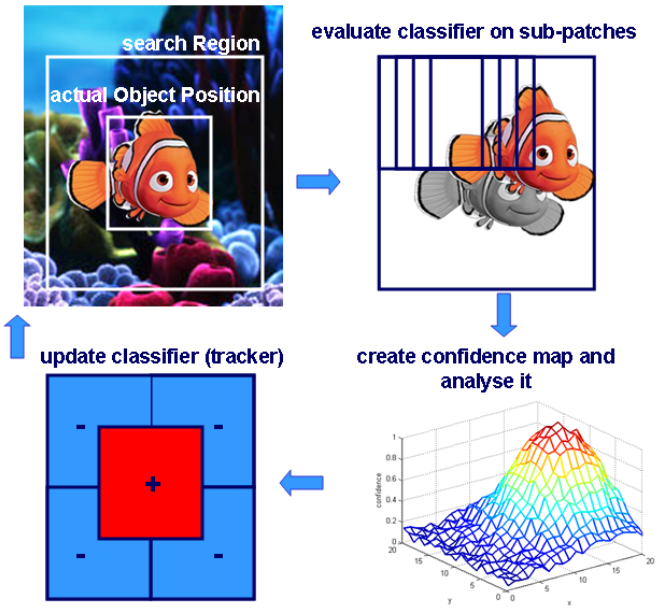
Online Boosting I

- Exploit the online Adaboost algorithm by Oza [Oza, 2001]
- Importance (difficulty) of a sample can be estimated by propagating it through the set of weak classifiers
- Offline setting: all samples are used to update and select one weak classifier
- Online setting: one sample is used to update all weak classifiers and the corresponding weight
- Grabner and Bischof [Grabner and Bischof, 2006b] present an online boosting algorithm for visual tracking using a pool of weak classifiers
- Online boosting for feature selection

Online Boosting II



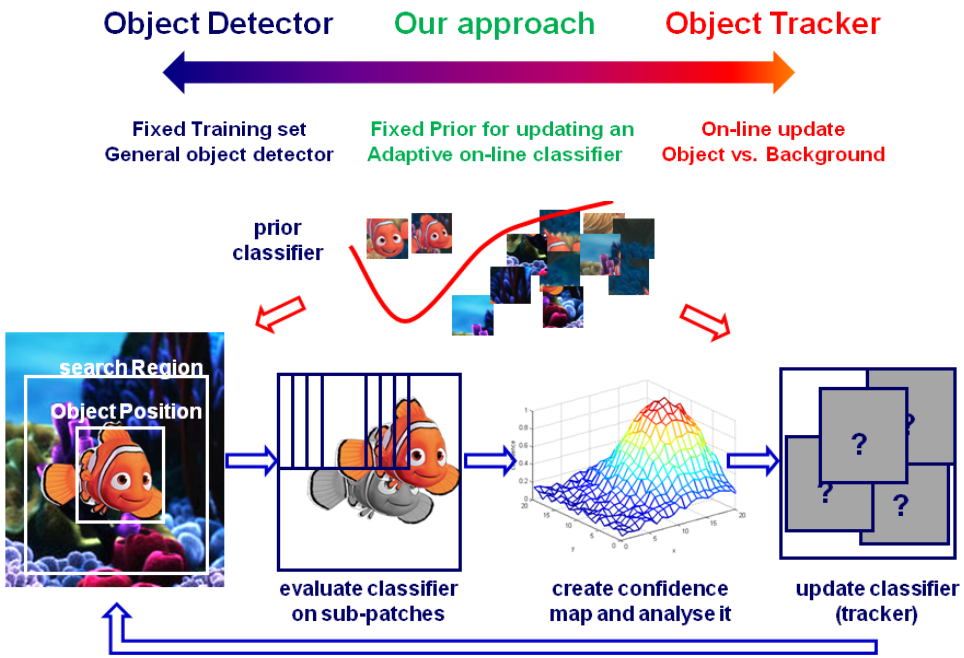
Online Boosting III



- Weak classifier: Gaussian distributions of Haar-like, HOG (histogram of gradients), and LBP (local binary pattern) features with Kalman filter
- 50 selectors, and each can choose from 250 weak classifiers
- Experimental results

Semi-supervised Tracking I

- Idea: Pure online visual tracking often leads to drifting problem
- [Grabner et al., 2008] apply semi-boost algorithm [Mallapragada et al., 2007] to visual tracking



- Use first frame to learn the prior

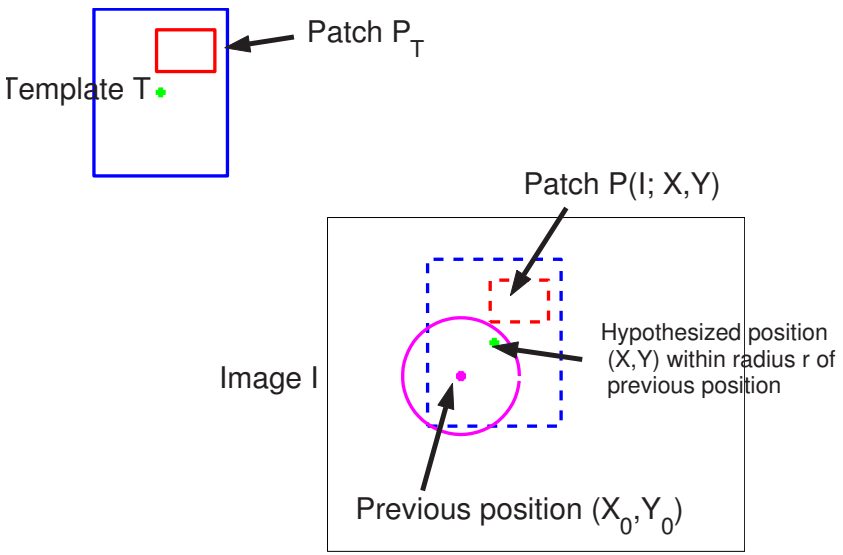
Semi-supervised Tracking II

Semiboost

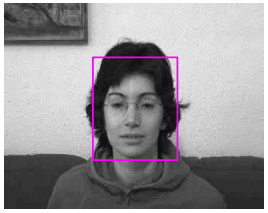
- Other semi-supervised methods
  - Co-inference [Wu and Huang, 2001]
  - Co-training [Javed et al., 2005]

FragTrack I

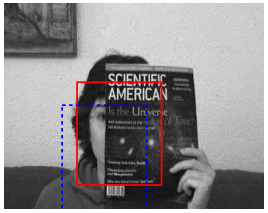
- FragTrack [Adam et al., 2006]
  - use multiple local image fragments or patches
  - use integral histogram [Porikli, 2005]
  - every patch votes on the possible position and scale
  - minimize a robust statistic to combine vote maps
  - use Earth Mover Distance (EMD) [Rubner et al., 2000] to compute similarity between histograms



FragTrack II

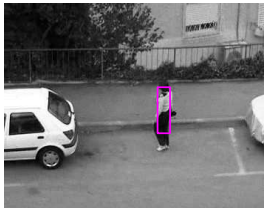

initial template

frame 222



frame 539

frame 849


initial template


frame 66

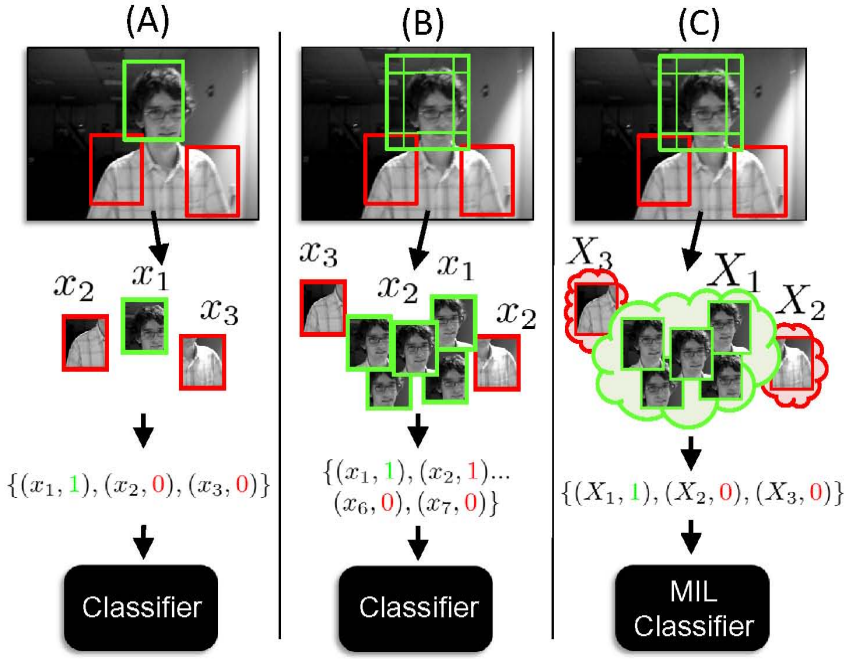

frame 134


frame 456

FragTrack face

FragTrack woman

Online Multiple Instance Learning (MIL)



- Inherent ambiguity of positive examples in sampling
- Multiple Instance Learning [Dietterich et al., 1997]
- Learning with positive bags and negative bags offline
  - Positive bag: At least one instance is positive
  - Negative bag: All instances are negative
- Batch mode discrete MILBoost for face detection [Viola et al., 2005]
- Develop continuous online MILBoost for visual tracking [Babenko et al., 2009]

Boosting and MILBoost

- Boosting: Given $x_i \rightarrow y_i$ (instance)

$$\mathbf{H}(x) = \sum_{k=1}^{K} \alpha_k \mathbf{h}_k(x)$$

where $\mathbf{h}_k(s)$ is a weak classifier and prediction via $\text{sgn}(\mathbf{H}_K(x))$

- MILBoost: Given $X_i \rightarrow y_i$ (bag)

$$\mathcal{L}(\mathbf{H}) = \sum_i y_i \log(p_i) + (1 - y_i) \log(1 - p_i)$$

where

$$\begin{array}{rcll} p_i & = & p(y_i = 1|X_i) = 1 - \prod_j (1 - p_{ij}) & \text{(Noisy-Or)} \\ p_{ij} & = & p(y_{ij} = 1|x_{ij}) & \text{(as LogitBoost)} \end{array}$$

Batch MILBoost

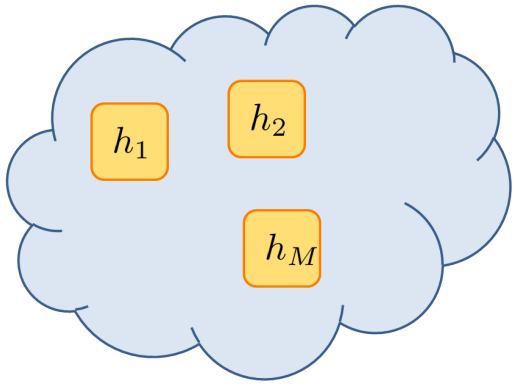
- Train weak classifiers in a greedy fashion

$$\mathbf{h}_{k+1} = \arg\max_{\mathbf{h}\in\mathcal{H}} \mathcal{L}(\mathbf{H}_k + \mathbf{h})$$

- For batch MILBoost, can optimize using functional gradient descent [Viola et al., 2005]
- For tracking, we need an online version

Online MILBoost

- At all times, keep a pool of $M \gg K$ weak classifier candidates

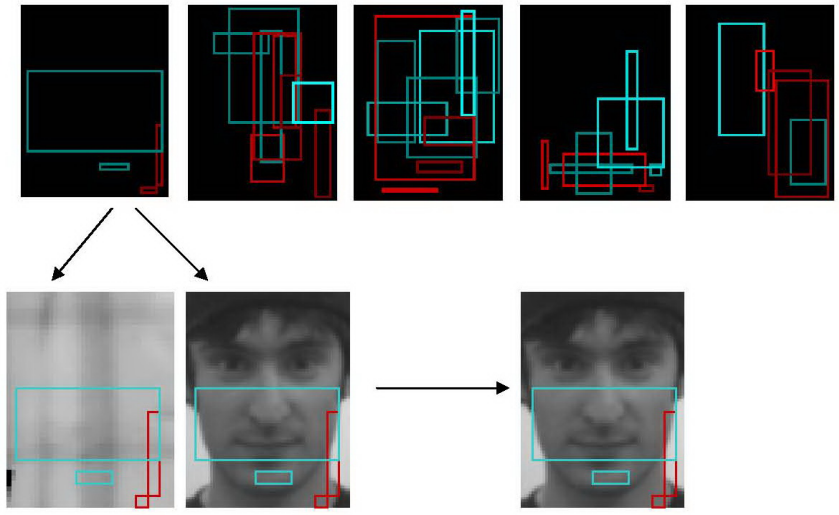


- At time t get more training data
  - Update all candidate classifiers
  - Pick best K in a greedy fashion

$$\mathbf{h}_{k+1} = \arg \max_{\mathbf{h} \in \{h_1, h_2, \ldots, h_M\}} \mathcal{L}(\mathbf{H}_k + \mathbf{h})$$

- Can be applied for classification and regression problems

Online MILBoost for Tracking



- Tracking by learning boosted detector online
- Online MILBoost: $\mathbf{H}(x) = \sum_{k=1}^{K} \alpha_k \mathbf{h}_k(x)$
- Weak classifier $\mathbf{h}_k(x)$: Univariate Gaussian densities of generalized Haar-like features
- Maximize bag likelihoods via stochastic gradient descent
- Online selection of weak classifiers (i.e., Haar-like features) that best separate foreground and background

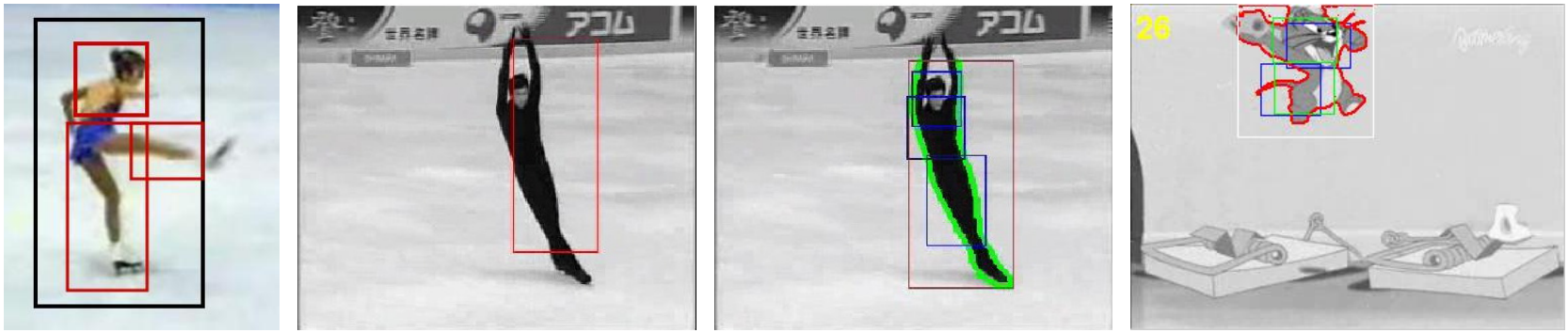
Visual Tracking with Online MILBoost: Results

- 20 fps on 3GHz machine with C++ implementation
- Handle fast motion, lighting/pose variation, and occlusion
- Only a few fixed parameters (no tuning)
- Quantitative and qualitative results
- Code and data available on the web

David Indoor | Tiger | Face

Boris | Tom and Jerry | Tom and Jerry 2

Online Articulated Object Tracking



- Tracking by detection with parts-based representation [Nejhum et al., 2008]
- Appearance: Represent an articulated object, $\mathbf{W}$, with weighted integral histograms of blocks $\lambda_i \mathbf{H}_i^{\mathbf{W}}$
- Shape: Find contour with fast graph cut segmentation
- Tracking articulated objects:
  - Detection: Scan the image and find $\mathbf{W}^* = \max_{\mathbf{W}'} \mathbf{S}(\mathbf{W}', \mathbf{W})$, where $\mathbf{S}(\mathbf{W}', \mathbf{W}) = \sum_{i=1}^{K} \lambda_i \rho(\mathbf{H}_i^{\mathbf{W}'}, \mathbf{H}_i^{\mathbf{W}})$
  - Refinement: Apply segmentation locally to find foreground and background
  - Update: Adjust block configuration locally, $\mathbf{H}_f = \sum_{i=1}^{K} \alpha_i \mathbf{H}_i^{\mathbf{W}}$

Online Articulated Object Tracking: Results

- Online update of shape and appearance [Nejhum et al., 2008]
- Use efficient graph cut algorithm for segmentation
- 4 frames per second on a 3GHz machine with MATLAB and MEX implementation
- Code and data available on the web

Lipinski

Lysacek

Guillem

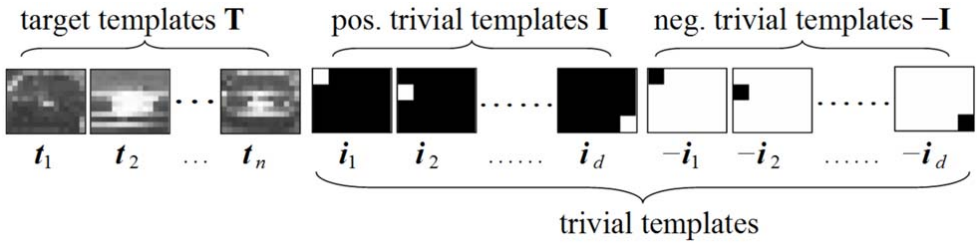
Tom and Jerry

Summary

- Online visual tracking algorithms with robust performance
  - Simultaneously track and update appearance model
  - Adaptive discriminative generative model
  - Online MILBoost for visual tracking
  - Tracking articulated objects with appearance and shape
- Able to track objects undergoing change in illumination and pose, with occlusions, and motion blurs
- Code and data are available on the web

Sparse Representation I

- Motivated by recent success in sparse representation
- Sparsify object representation with trivial templates
- Each target candidate is sparsely represented by target and trivial templates [Mei and Ling, 2009]

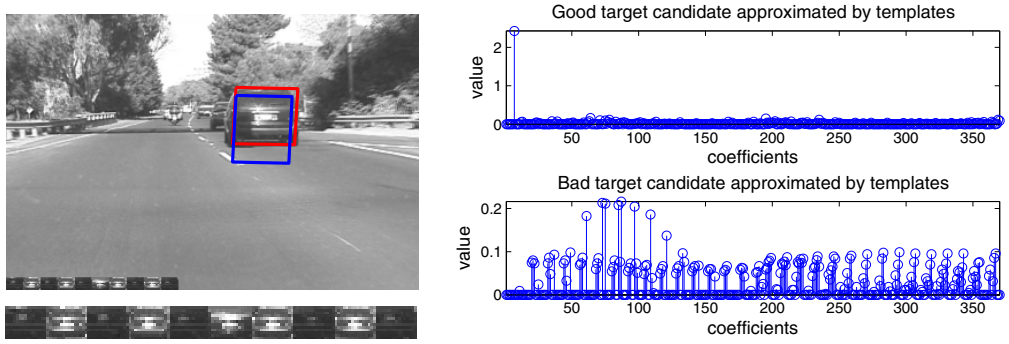


$$\mathbf{y} = \mathbf{T} = a_1\mathbf{t}_1 + a_2\mathbf{t}_2 + \cdots + a_n\mathbf{t}_n$$

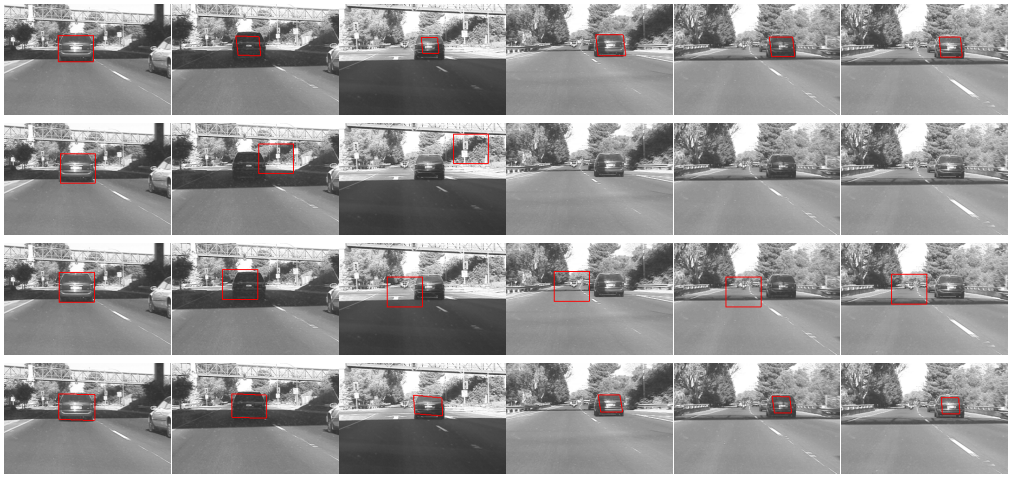
$$\mathbf{y} = \begin{bmatrix}\mathbf{T} & \mathbf{I}\end{bmatrix}\begin{bmatrix}\mathbf{a} \\ \mathbf{e}\end{bmatrix}$$

- Enforce non-negative coefficients
- Sparsify is achieved by ℓ_1 minimization
- Used with a particle filter

Sparse Representation II



- Tracking results



Top to bottom: [Mei and Ling, 2009] mean-shift tracker, covariance-based tracker [Porikli et al., 2006], and appearance-based particle filter [Zhou et al., 2004]

Sparse Representation III

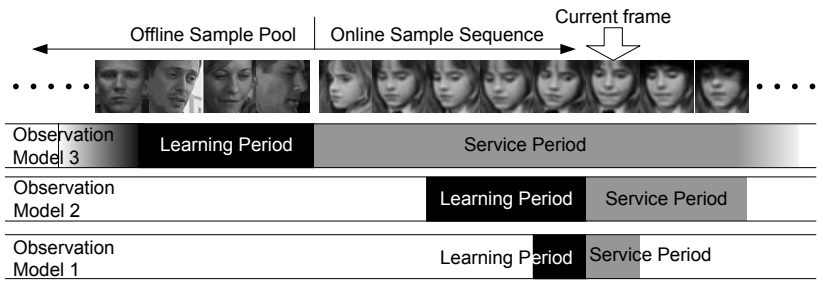
Top to bottom: [Mei and Ling, 2009] mean-shift tracker, covariance-based tracker [Porikli et al., 2006], and appearance-based particle filter [Zhou et al., 2004]

Multiple Trackers

- In layer [Toyama and Hager, 1999]
- In parallel [Birchfield, 1998] [Perez et al., 2002a] [vermaak et al., 2003]
- Homogeneous tracker [Li et al., 2007] [Kwon and Lee, 2010]
- Heterogeneous trackers [Stenger et al., 2009] [Santner et al., 2010]
- Feature level: [Bar-Shalom, 1992] [Isard and MacCormick, 2001] [Perez et al., 2002a] [Yu and Wu, 2004] [Wu and Huang, 2004] [Perez et al., 2002b]
- Tracker level:
  - Rapid face motion [Li et al., 2007]
  - Black box approach [Leichter et al., 2006]
  - Multiple observers [Stenger et al., 2009]: off-line training to find the best combination of tracking methods
  - Visual tracking decomposition [Kwon and Lee, 2010]
  - PROST [Santner et al., 2010]
- Connection with multi-view learning

Multiple Observers with Different Lifespans I

- Multiple observers (observation model) for face tracking [Li et al., 2007]
- Handle fast and abrupt motion with low frame rate images
- Each observer is learned from different ranges of samples, with different subsets of features



Multiple Observers with Different Lifespans II

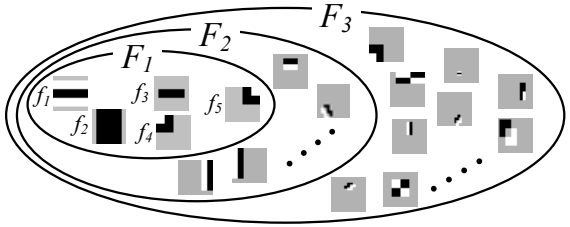


Figure 3. Feature set of each observation model.

k	1	2	3
L_k	A 5-dimension LDA classifier	Discrete AdaBoost on a pool of P LDA classifiers	Real AdaBoost on histogram weak classifiers
F_k	5 pre-selected Haar-like features	50 pre-selected Haar-like features	10,000s of Haar-like features
$\lvert\hat{F}_k\rvert$	5	≤ 50	≈ 500 per view
S_k	Samples of the previous frame	Samples of the previous 5 frames	10,000s of offline samples
$\tau_{k,on}$	$O(\lvert F_1\rvert^2 \lvert S_1\rvert)$	$O(\lvert F_2\rvert^2 \lvert S_2\rvert + \lvert S_2\rvert P^2)$	0
$\tau_{k,off}$	Negligible	Negligible	Several days
$\tau_{k,test}$	$O(\lvert\hat{F}_1\rvert)$	$O(\lvert\hat{F}_2\rvert)$	$O(\lvert\hat{F}_3\rvert)$

- With a cascade particle filter
- Tracking results

Multiple Observers with Different Lifespans III

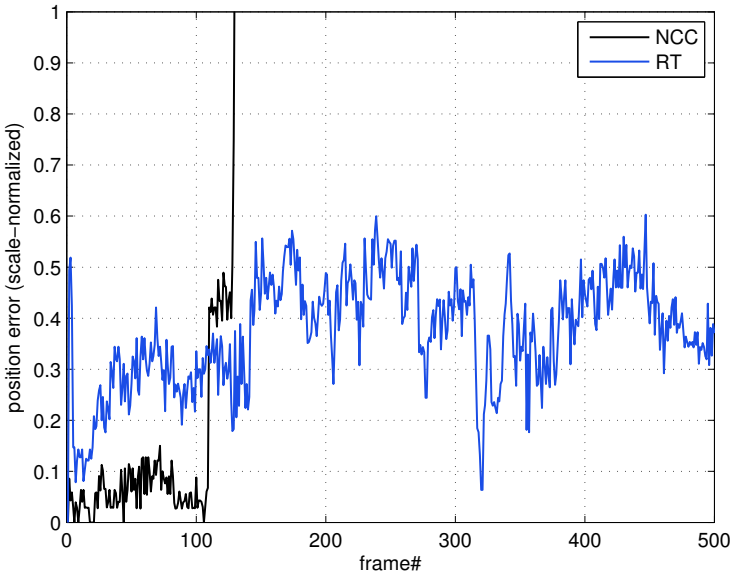
#59 #64 #86 #99 #101 #105

(a) Tracking by proposed approach under poor illumination, significant camera motion and fast zooming (yellow rectangle denotes target in the previous frame).

Li CVPR07

Learning with Multiple Trackers I

- Learning to fuse multiple trackers for face and hand tracking off-line [Stenger et al., 2009]
- Motivation example: tracking with single template using normalized cross correlation (NCC), and local features with randomized tree (RT)



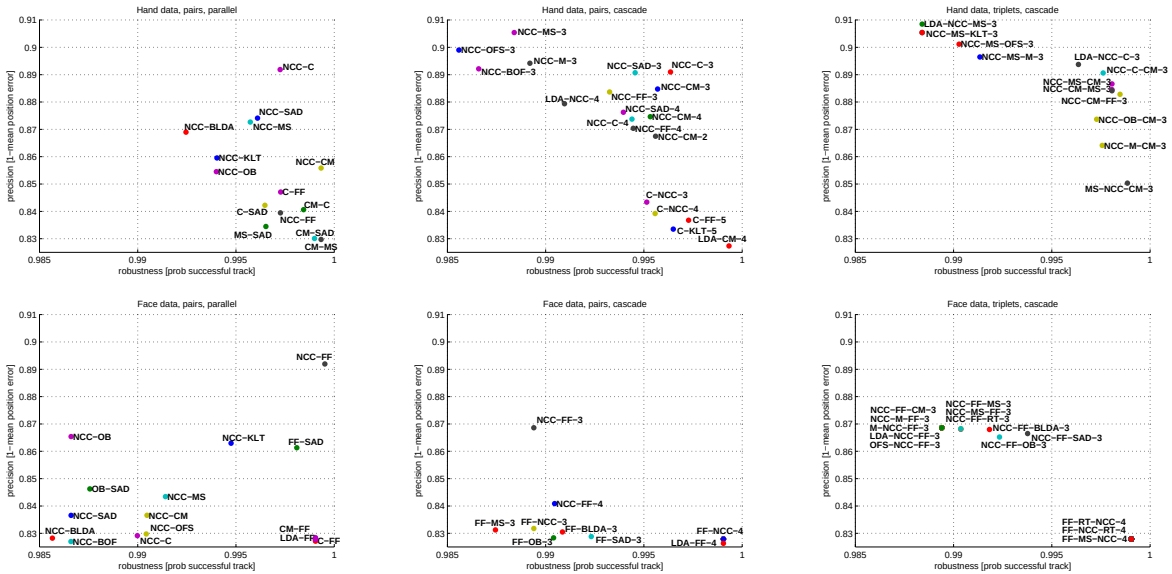
Learning with Multiple Trackers II

- 14 observers

Method	Observation	Estimate	Confidence value
NCC	Normalized cross correlation	max correlation	correlation score
SAD	Sum of absolute differences	min distance	distance score
BOF	Block-based optical flow of 3 $\times$ 3 templates	mean motion	mean NCC score
KLT [17]	Kanade-Lucas-Tomasi sparse optical flow using 50 features	centroid of good features	fraction of good features
FF [13]	Flocks of features: Tracking 50 local features with high color probability and 'flocking' constraints	centroid of good features	fraction of good features
RT [2]	Randomized templates: NCC track of eight subwindows, with motion consensus and resampling	centroid of good features	fraction of good features
MS [6]	Mean shift: Color histogram-based mean shift tracking with background weighting	min histogram distance	histogram distance
C [22]	Color probability map, blob detection	scale space maximum	probability score
M [22]	Motion probability map, blob detection	scale space maximum	probability score
CM [14]	Color and motion probability map	scale space maximum	probability score
OBD [9]	On-line boosted detector: Classifier boosted from pool of rectangle features updated on-line	max classifier output	classifier margin
LDA [16]	LDA classifier computed from five rectangle features in the previous frame (Observer 1 in [16])	max classifier output	classifier margin
BLDA [16]	Boosted LDA classifier using 50 LDA classifiers from a pool of 150, trained on the previous five frames (Obs. 2 in [16])	max classifier output	classifier margin
OFS [4]	On-line feature selection of 3 out of 49 color-based features based on fg/bg variance ratio	centroid of top features	mean variance ratio of selected features

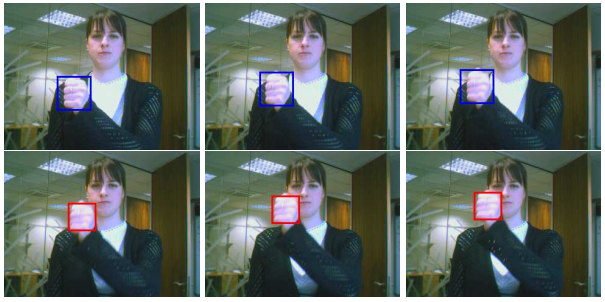
Learning with Multiple Trackers III

- Evaluation of observer combinations



(left) pairs, parallel evaluation, (middle) pairs, cascaded evaluation, (right) triplets, cascaded evaluation. Only a small subset of data points near the upper right frontier with both high robustness and precision are shown here.

Learning with Multiple Trackers IV



NCC-CM-FF cascade

NCC-FF-MS cascade

NCC-FF-MS cascade

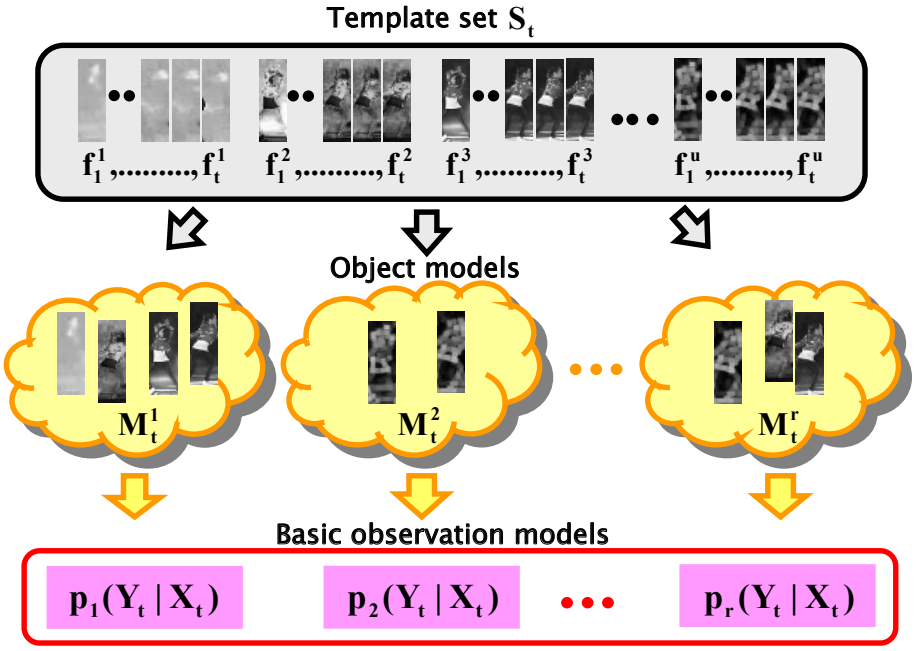
Observation: cascade evaluation gives similar performance to parallel evaluation at much higher efficiency

Visual Tracking Decomposition I

- Bayesian formulation and its weighted components [Kwon and Lee, 2010]
$$p(X_t|Y_{1:t}) \propto p(Y_t|X_t) \int p(X_t|X_{t-1})p(X_t|X_{t-1}P(X_{t-1}|Y_{1:t-1})dX_{t-1}$$
- Observation model : Decompose into multiple basic basic ones
$$p(Y_t|X_t) = \sum_{i=1}^{r} w_t^i p_i(Y_t|X_t), \sum_{i=1}^{r} w_t^i = 1 \quad (16)$$
- Motion model: Decompose into multiple basic motion models
$$p(X_t|X_{t-1}) = \sum_{j=1}^{s} w_t^j p_j(X_t|X_{t-1}), \sum_{j=1}^{s} w_t^j = 1 \quad (17)$$
- Multiple basic trackers are designed by associating the basic observation and motion models and each account for certain change of the object
- All basic trackers are integrated with an interactive Makrov Chain Monte Carlo framework

Visual Tracking Decomposition II

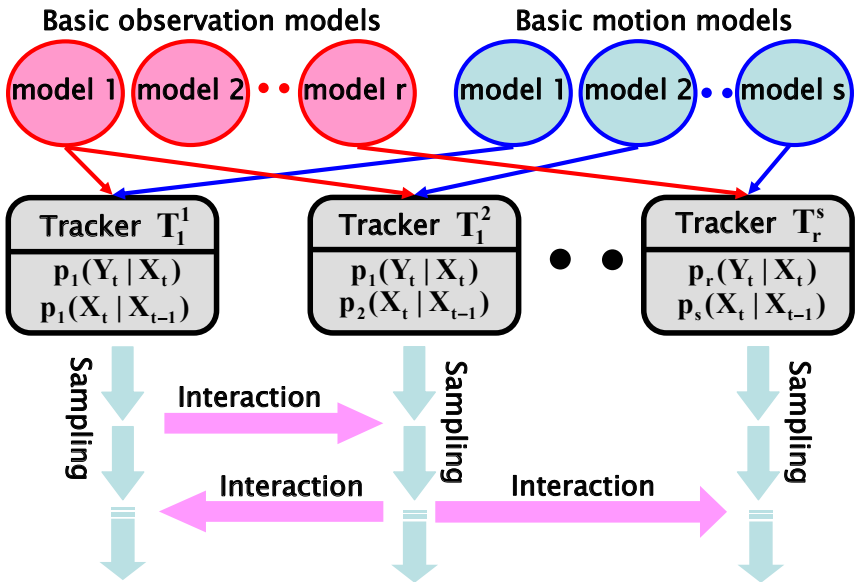
- Basic observation model



  - mixture of different types of feature templates, e.g., hue, saturation, intensity, and edge
  - find a sparse set of templates by sparse PCA
  - using first 5 frames and the most recent 4 frames
  - use a diffusion distance to compute distance between histograms

Visual Tracking Decomposition III

- Basic motion model
  - random walk: $p_j(X_t|X_{t-1}) = N(X_{t-1}, \sigma_j^2)$
  - two types of motion with small and large variance
- Basic tracker models



  - Construct a Markov chain modeled by one pair of basic observation and motion model
  - MAP estimate via the Metropolis Hasting algorithm

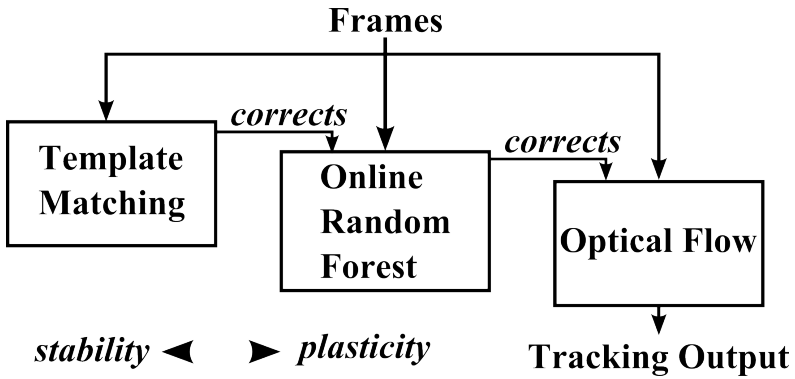
Visual Tracking Decomposition IV

VTD results

- Summary:
  - Pros:
    - use mixture of representations and motion models
  - Cons:
    - numerous parameters
    - time consuming

PROST I

- PROST (German word for "Cheers") [Santner et al., 2010]:
  - Template correlation with normalized correlation (NCC): use the first frame
  - Mean-shift in conjunction with a variant of optical flow (FLOW)
  - Online random forest (ORF)
- Tracker combination
  - FLOW is overruled by ORF if they are not overlapping and ORF has a confidence above a threshold
  - ORF is updated only if it overlaps with NCC or FLOW



Tracking with Reference Objects

- Object-centered activity analysis: With known objects, it is easier to analyze the activity associated with objects [Laxton et al., 2007]
- Context-aware visual tracking [Yang et al., 2009]
- Tracking with the invisible (using relationship between target and surrouding objects) [Grabner et al., 2010]
- Human tracking via interactive objects [Kjellstrom et al., 2010]

Performance Evaluation

- Evaluation metrics:
  - time
  - accuracy: position, overlapping area, angle
  - motion information: similarity/affine transform
  - consistency
  - off-line training
  - recover from failure
  - qualitative and quantitative
  - lighting
  - feature
  - multiple objects
  - image sensor
  - single tracker
- Data sets:
  - "ground truth"

Open Issues

- Heavy occlusion
- Articulated non-rigid motions
- Failure recovery
- Drifting problems
- Multiple targets
- Markless 3D human tracking
- Context and prior knowledge
- Simultaneous detection, tracking, and recognition
- Long term and short term memory

Concluding Remarks

- Application-dependent
- Much work has been done, and yet much more work is to be done
- “Robust” tracking
- Cognitive vision

References I

Adam, A., Rivlin, E., and Shimshoni, I. (2006).
Robust fragments-based tracking using the integral histogram.
In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 798–805.

Avidan, S. (2004).
Support vector tracking.
IEEE Transactions on Pattern Analysis and Machine Intelligence, 26(8):1064–1072.

Avidan, S. (2007).
Ensemble tracking.
IEEE Transactions on Pattern Analysis and Machine Intelligence, 29(2):261–271.

Babenko, B., Yang, M.-H., and Belongie, S. (2009).
Visual tracking with online multiple instance learning.
In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 983–990.

Baker, S. and Matthews, I. (2004).
Lucas-Kanade 20 years on: A unifying framework.
International Journal of Computer Vision, 56(3):221–255.

Baker, S., Scharstein, D., Lewis, J. P., Roth, S., Black, M., and Szeliski, R. (2007).
A database and evaluation methodology for optical flow.
In *Proceedings of the IEEE International Conference on Computer Vision*.

Bar-Shalom, Y., editor (1992).
Multitarget-multisensor tracking.
Artech House.

References II

Barron, J. L., Fleet, D. J., and Beauchemin, S. S. (1994).
Performance of optical flow techniques.
International Journal of Computer Vision, 12(1):43–77.

Bay, H., Tuytelaars, T., and Gool, L. V. (2006).
Surf: Speeded up robust features.
In *Proceedings of European Conference on Computer Vision*, pages 404–417.

Belhumeur, P. and Kreigman, D. (1997).
What is the set of images of an object under all possible lighting conditions.
In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 270–277.

Belhumeur, P. N. and Kriegman, D. J. (1998).
What is the set of images of an object under all possible illumination conditions?
International Journal of Computer Vision, 28(3):245–260.

Birchfield, S. (1998).
Elliptical head tracking using intensity gradient and color histograms.
In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 232–37.

Birchfield, S. and Rangarajan, S. (2005).
Spatiograms vs. histograms.
In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 1158–1163.

Black, M. and Anandan, P. (1996).
The robust estimation of multiple motions: Parametric and piecewise-smooth flow fields.
Computer Vision and Image Understanding, 63(1):75–104.

References III

Black, M. J. and Jepson, A. D. (1998).

Eigentracking: Robust matching and tracking of articulated objects using a view-based representation.
International Journal of Computer Vision, 26(1):63–84.

Bookstein, F. (1989).

Principal warps: Thin-plate splines and the decomposition of deformations.
IEEE Transactions on Pattern Analysis and Machine Intelligence, 11(6):567–585.

Bregler, C. and Malik, J. (1998).

Tracking people with twists and exponential map.
In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 8–15.

Cannons, K. (2008).

A review of visual tracking.
Technical Report CSE-2008-07, York University.

Caselles, V., Kimmel, R., and Sapiro, G. (1997).

Geodesic active contours.
International Journal of Computer Vision, 22(1):61–79.

Cham, T.-J. and Rehg, J. (1999).

A multiple hypothesis approach to figure tracking.
In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 239–245.

Collins, R. T. (2003).

Mean-shift blob tracking through scale space.
In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 234–240.

References IV

Collins, R. T. and Liu, Y. (2003).

On-line selection of discriminative tracking features.

In *Proceedings of the IEEE International Conference on Computer Vision*, pages 346–352.

Comaniciu, D. (2003).

An algorithm for data-driven bandwidth selection.

IEEE Transactions on Pattern Analysis and Machine Intelligence, 25(2):281–288.

Comaniciu, D. and Meer, P. (2002).

Mean shift: a robust approach toward feature space analysis.

IEEE Transactions on Pattern Analysis and Machine Intelligence, 24(5):603 –619.

Comaniciu, D., Ramesh, V., and Meer, P. (2000).

Real-time tracking of non-rigid objects using mean shift.

In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 142–149.

Comaniciu, D., Ramesh, V., and Meer, P. (2003).

Kernel-based object tracking.

IEEE Transactions on Pattern Analysis and Machine Intelligence, 25(5):564–577.

Cootes, T. F., Edwards, G. J., and Taylor, C. J. (1998).

Active appearance models.

In *Proceedings of European Conference on Computer Vision*, pages 383–498.

Deutscher, J., Blake, A., and Reid, I. (2000).

Articulated body motion capture by annealed particle filtering.

In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 126–133.

References V

Dietterich, T. G., Lathrop, R. H., and Perez, L. T. (1997).
Solving the multiple-instance problem with axis parallel rectangles.
Artificial Intelligence, 89(1-2):31–71.

Elgammal, A., Duraiswami, R., Harwood, D., and Davis, L. S. (2002).
Background and foreground modeling using non-parametric kernel density estimation for visual surveillance.
Proceedings of the IEEE, 90(7):1151–1163.

Elgammal, A. M., Harwood, D., and Davis, L. S. (2000).
Non-parametric model for background subtraction.
In *Proceedings of European Conference on Computer Vision*, pages 751–767.

Forsyth, D., Arikan, O., Ikemoto, L., O'Brien, J., and Ramanan, D. (2006).
Computational studies of human motion: Part 1, Tracking and motion synthesis.
Now publishers.

Golub, G. H. and Van Loan, C. F. (1996).
Matrix Computations.
The Johns Hopkins University Press.

Grabner, H. and Bischof, H. (2006a).
On-line boosting and vision.
In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 260–267.

Grabner, H. and Bischof, H. (2006b).
On-line boosting and vision.
In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 260–267.

References VI

Grabner, H., Leistner, C., and Bischof, H. (2008).
Semi-supervised on-line boosting for robust tracking.
In *Proceedings of European Conference on Computer Vision*, pages 234–247.

Grabner, H., Matas, J., Gool, L. J. V., and Cattin, P. C. (2010).
Tracking the invisible: Learning where the object might be.
In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 1285–1292.

Hager, G. D. and Belhumeur, P. N. (1998).
Efficient region tracking with parametric models of geometry and illumination.
IEEE Transactions on Pattern Analysis and Machine Intelligence, 20(10):1025–1039.

Hall, P., Marshall, D., and Martin, R. (1998).
Incremental eigenanalysis for classification.
In *Proceedings of British Machine Vision Conference*, pages 286–295.

Han, B., Comaniciu, D., Zhu, Y., and Davis, L. S. (2008).
Sequential kernel density approximation and its application to real-time visual tracking.
IEEE Transactions on Pattern Analysis and Machine Intelligence, 30(7):1186–1197.

Han, B., Zhu, Y., Comaniciu, D., and Davis, L. S. (2009).
Visual tracking by continuous density propagation in sequential Bayesian filtering framework.
IEEE Transactions on Pattern Analysis and Machine Intelligence, 31(5):919–930.

Haritaoglu, I., Harwood, D., and Davis, L. S. (1998).
W4: A real time system for detecting and tracking people.
In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, page 962.

References VII

Harris, C. and Stephens, M. (1988).
A combined corner and edge detector.
In *Proceedings of The Fourth Alvey Vision Conference*, pages 147–151.

Ho, J., Lee, K.-C., Yang, M.-H., and Kriegman, D. (2004).
Visual tracking using learned linear subspaces.
In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 782–789.

Horn, B. K. P. and Schunck, B. (1981).
Determining optical flow.
Artificial Intelligence, 17:185–203.

Hua, G. and Wu, Y. (2004).
Multi-scale visual tracking by sequential belief propagation.
In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 826–833.

Ioffe, S. and Forsyth, D. (2001).
Human tracking with mixtures of trees.
In *Proceedings of the IEEE International Conference on Computer Vision*, pages 690–695.

Isard, M. and Blake, A. (1996).
Contour tracking by stochastic propagation of conditional density.
In *Proceedings of European Conference on Computer Vision*, pages 343–356.

Isard, M. and MacCormick, J. (2001).
BraMBLe: A Bayesian multiple blob tracker.
In *Proceedings of the IEEE International Conference on Computer Vision*, pages 34–41.

References VIII

Javed, O., Ali, S., and Shah, M. (2005).
Online detection and classification of moving objects using progressively improving detectors.
In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 696–701.

Jepson, A. D., Fleet, D. J., and El-Maraghi, T. F. (2001).
Robust online appearance models for visual tracking.
In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 415–422.

Jepson, A. D., Fleet, D. J., and El-Maraghi, T. F. (2003).
Robust online appearance models for visual tracking.
IEEE Transactions on Pattern Analysis and Machine Intelligence, 25(10):1296–1311.

Ju, S. X., Black, M. J., and Yacoob, Y. (1996).
Cardboard people: A parameterized model of articulated image motion.
In *Proceedings of IEEE International Conference on Automatic Face and Gesture Recognition*, pages 38–44.

Kalal, Z., Matas, J., and Mikolajczyk, K. (2010).
P-n learning: Boostrapping binary classifiers by structural constraints.
In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*.

Kass, M., Witkin, A., and Terzopoulos, D. (1987).
Snakes: Active contour models.
International Journal of Computer Vision, 1(4):321–331.

Khan, Z., Balch, T. R., and Dellaert, F. (2004).
A rao-blackwellized particle filter for eigentracking.
In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 980–986.

References IX

Kjellstrom, H., Kragic, D., and Black, M. J. (2010).
Tracking people interacting with objects.
In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 747–754.

Kwon, J. and Lee, K. M. (2010).
Visual tracking decomposition.
In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*.

Laxton, B., Lim, J., and Kriegman, D. (2007).
Leveraging temporal, contextual and ordering constraints for recognizing complex activities.
In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*.

Leichter, I., Lindenbaum, M., and Rivlin, E. (2006).
A general framework for combining visual trackers - The "black boxes" approach.
International Journal of Computer Vision, 67(3):343–363.

Lepetit, V. and Fua, P. (2005).
Monocular model-based 3d tracking of rigid objects: A survey.
Foundations and trends in computer graphics and vision, 1(1):1–89.

Levy, A. and Lindenbaum, M. (2000).
Sequential Karhunen-Loeve basis extraction and its application to images.
IEEE Transactions on Image Processing, 9(8):1371–1374.

Li, R., Yang, M.-H., Sclaroff, S., and Tian, T.-P. (2006).
Monocular tracking of 3D human motion with a coordinated mixture of factor analyzers.
In *Proceedings of European Conference on Computer Vision*, pages 137–150.

References X

Li, Y., Ai, H., Yamashita, T., Lao, S., and Kawade, M. (2007).
Tracking in low frame rate video: a cascade particle filter with discriminative observers of different lifespans.
In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 1728–1740.

Lim, J. and Yang, M.-H. (2005).
A direct method for modeling non-rigid motion with thin plate spline.
In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 1196–1202.

Lin, R.-S., Liu, C.-B., Yang, M.-H., Ahuja, N., and Levinson, S. (2004).
Learning nonlinear manifolds from time series.
In *Proceedings of European Conference on Computer Vision*, pages 239–250.

Lindeberg, T. (1998).
Feature detection with automatic scale selection.
International Journal of Computer Vision, 30(2):79–116.

Liu, C., Yuen, J., Torralba, A., Sivic, J., and Freeman, W. T. (2008).
Sift flow: Dense correspondence across different scenes.
In *Proceedings of European Conference on Computer Vision*, pages 28–42.

Lowe, D. (2004).
Distinctive image features from scale-invariant keypoints.
International Journal of Computer Vision, 60(2):91–110.

Lucas, B. and Kanade, T. (1981).
An iterative image registration technique with an application to stereo vision.
In *Proceedings of International Joint Conference on Artificial Intelligence*, pages 674–679.

References XI

MacCormick, J. and Isard, M. (2000).

Partitioned sampling, articulated objects, and interface-quality hand tracking.
In *Proceedings of European Conference on Computer Vision*, pages 3–19.

Mallapragada, P., Jin, R., Jain, A., and Liu, Y. (2007).

Semiboost: Boosting for semisupervised learning.
Technical report, Michigan State University.

Matthews, I., Ishikawa, T., and Baker, S. (2004).

The template update problem.
IEEE Transactions on Pattern Analysis and Machine Intelligence, 26(6):810–815.

Mei, X. and Ling, H. (2009).

Robust visual tracking using ℓ_1 minimization.
In *Proceedings of the IEEE International Conference on Computer Vision*.

Mikolajczyk, K. and Schmid, C. (2005).

A performance evaluation of local descriptors.
IEEE Transactions on Pattern Analysis and Machine Intelligence, 27(10):1615–1630.

Moeslund, T., Hilton, A., and Kruger, V. (2006).

A survey of advances in vision-based human motion capture and analysis.
Computer Vision and Image Understanding, 104(2):90–126.

Murase, H. and Nayar, S. K. (1995).

Visual learning and recognition of 3-D objects from appearance.
International Journal of Computer Vision, 14:5–24.

References XII

Nejhum, S. M. S., Ho, J., and Yang, M.-H. (2008).
Online articulate object tracking with appearance and shape.
In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*.

Okuma, K., Taleghani, A., de Freitas, N., Little, J., and Lowe, D. (2004).
A boosted particle filter: Multitarget detection and tracking.
In *Proceedings of European Conference on Computer Vision*, pages 28–39.

Oza, N. C. (2001).
Online Ensemble Learning.
Ph.D. Thesis, University of California, Berkeley.

Paragios, N. and Deriche, R. (2000).
Geodesic active contours and level sets for the detection and tracking of moving objects.
IEEE Transactions on Pattern Analysis and Machine Intelligence, 22(3):266–280.

Pavlovic, V., Rehg, J. M., Cham, T.-J., and Murphy, K. P. (1999).
A dynamic Bayesian network approach to figure tracking using learned dynamic models.
In *Proceedings of the IEEE International Conference on Computer Vision*, pages 94–101.

Perez, P., Hue, C., Vermaak, J., and Gangnet, M. (2002a).
Color-based probabilistic tracking.
In *Proceedings of European Conference on Computer Vision*, pages 661–674.

Perez, P., Vermaak, J., and Blake, A. (2002b).
Data fusion for visual tracking with particles.
In *Proceedings of European Conference on Computer Vision*, pages 495–513.

References XIII

Porikli, F. (2005).

Integral histogram: A fast way to extract histograms in Cartesian spaces.
In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 829–836.

Porikli, F., Tuzel, O., and Meer, P. (2006).

Covariance tracking using model update based on Lie algebra.
In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 728–735.

Rasmussen, C. and Hager, G. D. (2001).

Probabilistic data association methods for tracking complex visual objects.
IEEE Transactions on Pattern Analysis and Machine Intelligence, 23(6):560–576.

Rehg, J. and Kanade, T. (1994).

Visual tracking of high DOF articulated structures: An approach to human hand tracking.
In *Proceedings of the IEEE International Conference on Computer Vision*, pages 35–46.

Reid, D. (1979).

An algorithm for tracking multiple targets.
IEEE Transactions on Automatic Control, 24(6):843–854.

Ross, D., Lim, J., Lin, R.-S., and Yang, M.-H. (2008).

Incremental learning for robust visual tracking.
International Journal of Computer Vision, 77(1-3):125–141.

Rubner, Y., Tomasi, C., and Guibas, L. (2000).

The earth mover's distance as a metric for image retrieval.
International Journal of Computer Vision, 40(2):91–121.

References XIV


Santner, J., Leistner, C., Saffari, A., Pock, T., and Bischof, H. (2010).
PROST: Parallel robust online simple tracking.
In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*.


Sclaroff, S. and Isidoro, J. (1998).
Active blobs.
In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1146–1153.


Shi, J. and Tomasi, C. (1994).
Good features to track.
In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 593–600.


Sidenbladh, H., Black, M., and Fleet, D. (2000).
Stochastic tracking of 3D human figures using 2D image motion.
In *Proceedings of European Conference on Computer Vision*, pages 702–718.


Sigal, L., Bhatia, S., Roth, S., Black, M., and Isard, M. (2004).
Tracking loose-limbed people.
In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 421–428.


Sminchisescu, C. and Triggs, B. (2001).
Covariance scaled sampling for monocular 3D body tracking.
In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 447–454.


Stauffer, C. and Grimson, W. E. L. (1999).
Adaptive background mixture models for real-time tracking.
In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 246–252.

References XV


Stenger, B., Woodley, T., and Cipolla, R. (2009).
Learning to track with multiple observers.
In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*.


Sullivan, J. and Rittscher, J. (2001).
Guiding random particles by deterministic search.
In *Proceedings of the IEEE International Conference on Computer Vision*, pages 323–330.


Ta, D.-N., Chen, W.-C., Gelfand, N., and Pulli, K. (2009).
Surftrac: Efficient tracking and continuous object recognition using local feature descriptors.
In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 2937–2944.


Tao, H., Sawhney, H., and Kumar, R. (2002).
Object tracking with Bayesian estimation of dynamic layer representations.
IEEE Transactions on Pattern Analysis and Machine Intelligence, 24(1):75–89.


Toyama, K. and Blake, A. (2001).
Probabilistic tracking in a metric space.
In *Proceedings of the IEEE International Conference on Computer Vision*, pages 50–57.


Toyama, K. and Hager, G. (1999).
Incremental focus of attention for robust vision-based tracking.
International Journal of Computer Vision, 35(1):45–63.


Urtasun, R., Fleet, D., and Fua, P. (2006).
3d people tracking with Gaussian process dynamical models.
In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 238–245.

References XVI

vermaak, J., Doucet, A., and Perez, P. (2003).
Maintaining multimodality through mixture tracking.
In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1110–1106.

Viola, P., Platt, J. C., and Zhang, C. (2005).
Multiple instance boosting for object detection.
In *Advances in Neural Information Processing Systems*, pages 1417–1426.

Welch, G. and Bishop, G. (1995).
An introduction to the Kalman filter.
Technical Report TR 95-041, University of North Carolina at Chapel Hill.

Williams, O. M. C., Blake, A., and Cipolla, R. (2005).
Sparse Bayesian learning for efficient visual tracking.
IEEE Transactions on Pattern Analysis and Machine Intelligence, 27(8):1292–1304.

Wren, C., Azarbayejani, A., Darrell, T., and Pentland, A. (1997).
Pfinder: real-time tracking of the human body.
IEEE Transactions on Pattern Analysis and Machine Intelligence, 19(7):780–785.

Wu, Y. and Huang, T. (2001).
A co-inference approach for robust visual tracking.
In *Proceedings of the IEEE International Conference on Computer Vision*, pages 26–33.

Wu, Y. and Huang, T. S. (2004).
Robust visual tracking by integrating multiple cues based on co-inference learning.
International Journal of Computer Vision, 58(1):55–71.

References XVII

Wu, Y., Lin, J., and Huang, T. (2001).
Capturing natural hand articulation.
In *Proceedings of the IEEE International Conference on Computer Vision*, pages 426–432.

Yang, C., Duraiswami, R., and Davis, L. S. (2004).
Efficient kernel machines using the improved fast Gauss transform.
In *Advances in Neural Information Processing Systems*, pages 1561–1568.

Yang, M., Wu, Y., and Hua, G. (2009).
Context-aware visual tracking.
IEEE Transactions on Pattern Analysis and Machine Intelligence, 31(7):1195–1209.

Yilmaz, A., Javed, O., and Shah, M. (2006).
Object tracking: A survey.
ACM Computing Surveys, 38(4):1–45.

Yu, T. and Wu, Y. (2004).
Collaborative tracking of multiple targets.
In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 834–841.

Zhou, S. K., Chellappa, R., and Moghaddam, B. (2004).
Visual tracking and recognition using appearance-adaptive models in particle filters.
IEEE Transactions on Image Processing, 13(11):1491–1506.