A glimpse at visual tracking

Patrick Pérez

ENS-INRIA VRML Summer School
ENS Paris, July 2013

technicolor

https://research.technicolor.com/~PatrickPerez

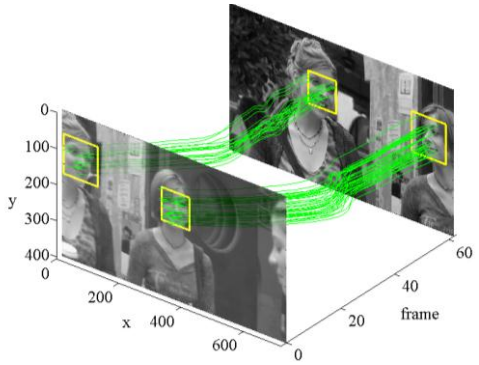
Outline

- Introduction
  - What and why?
  - Formalization

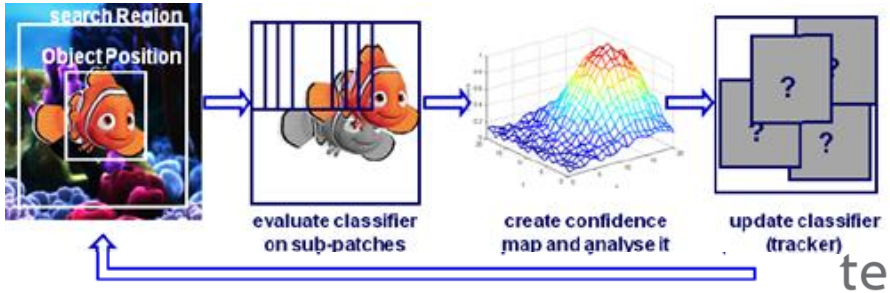


- Probabilistic filtering
  - Main concepts
  - Particle filters

- Tracking image regions
  - Point tracking
  - Arbitrary “objects”



- Online learning
  - Descriptive
  - Discriminative

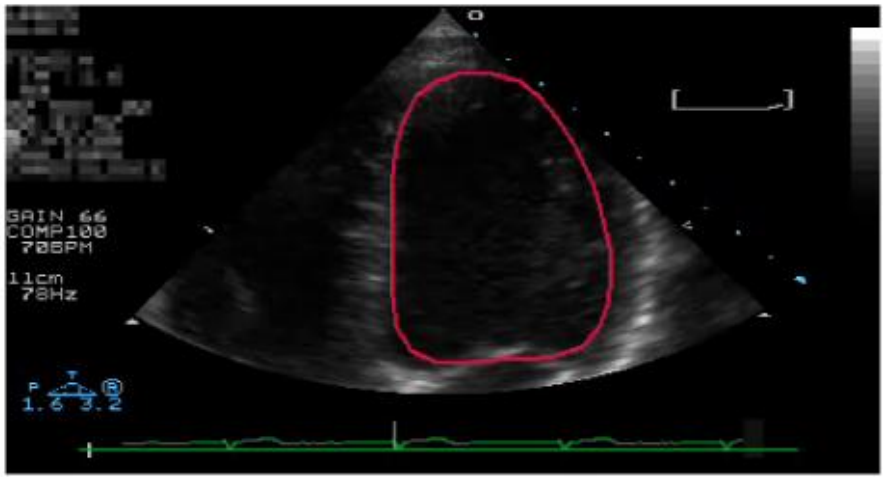


technicolor

What?

- On-line or off-line inference, from a mono- or multi-view image sequence, of state trajectories that characterize, either in image plane or in real world, some aspects of one or several target objects
- All sorts of “targets”
  - Interest points
  - Manually selected objects
  - Specific known objet
  - Cars, faces, people, etc.
  - Moving cars, walking people, talking heads
- Appearance/dynamical models and inference machineries
  - Depend on task and setting
  - Heavily influenced by CV/ML trends

With 2D (dynamic) shape prior

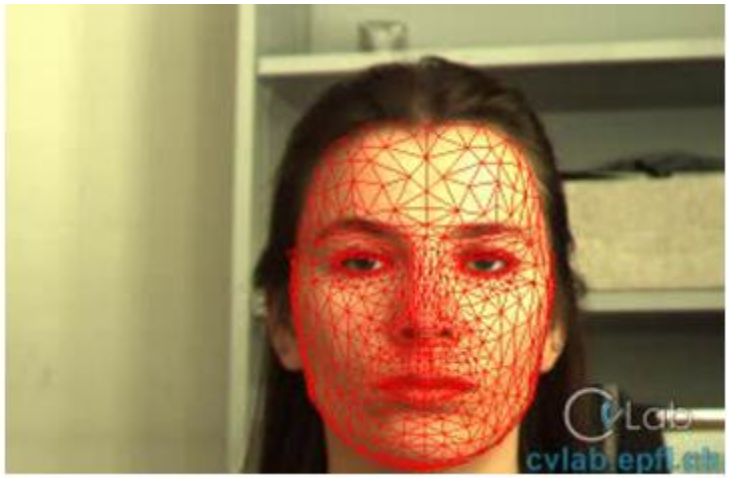


http://www2.imm.dtu.dk/~aam/tracking/

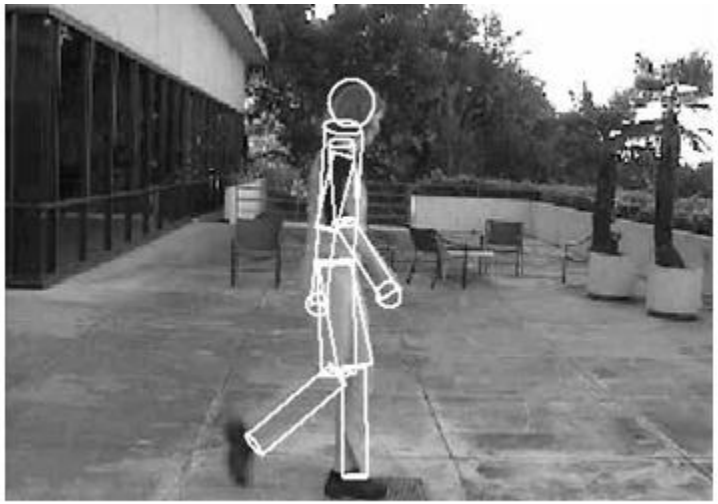
http://vision.ucsd.edu/~kbranson/research/cvpr2005.html

technicolor

With 3D (cinematic) shape prior



http://cvlab.epfl.ch/research/completed/realtime_tracking/

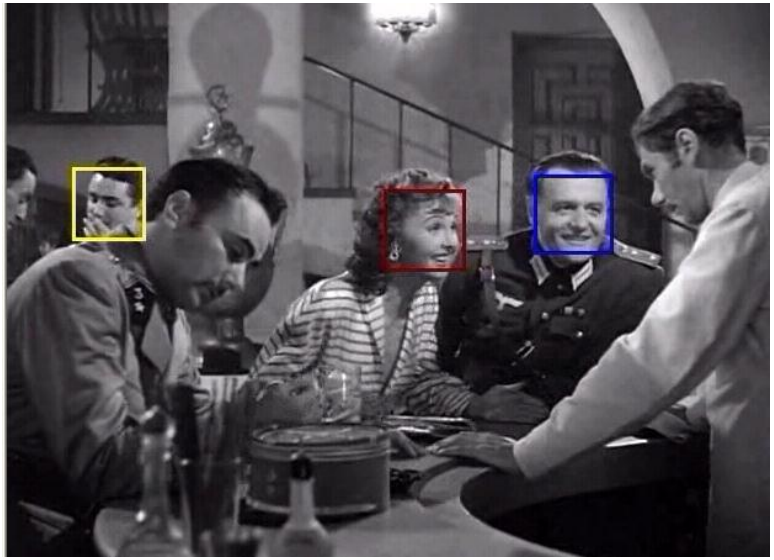


http://www.cs.brown.edu/~black/3Dtracking.html

technicolor

With appearance prior

- "Detect-before-tracking"

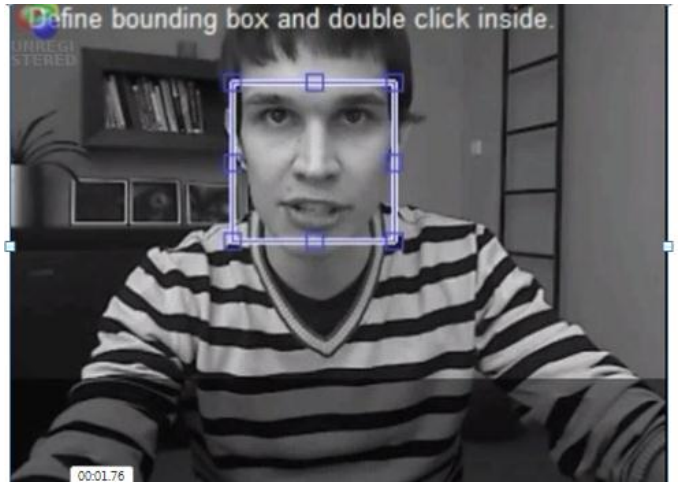


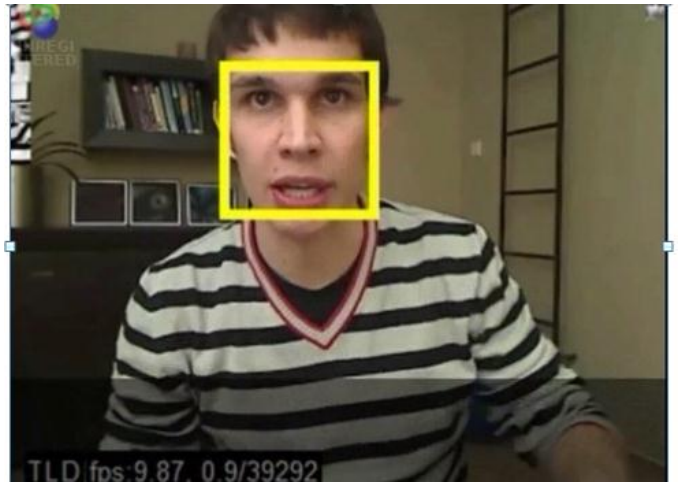
http://www.cs.washington.edu/homes/xren/research/cvpr2008_casablanca/

technicolor

With no appearance prior

- Tracking bounding box from user selection





http://info.ee.surrey.ac.uk/Personal/Z.Kalal/

technicolor

With no appearance prior

- Tracking bounding box from user selection (query expansion)

http://www.robots.ox.ac.uk/~vgg/research/vgoogle/

technicolor

With no appearance prior

- Tracking bounding box from user selection, and using context

http://server.cs.ucf.edu/~vision/projects/sali/CrowdTracking/index.html

technicolor

With no appearance prior

- Tracking bounding box and segmentation from user selection

http://www.robots.ox.ac.uk/~cbibby/index.shtml

technicolor

Why?

Elementary or principal tool for multiple CV systems

- Other sciences (neuroscience, ethology, biomechanics, sport, medicine, biology, fluid mechanics, meteorology, oceanography)
- Defense, surveillance, safety, monitoring, control, assistance
- Robotics, Human-Computer Interfaces

Disposable video (camera as a sensor)

- Video content production and post-production (compositing, augmented reality, editing, re-purposing, stereo3D authoring, motion capture for animation, clickable hyper videos, etc.)
- Video content management (indexing, annotation, search, browsing)

Valuable video

A specific problem?

More than yet another search/matching/detection problem

- Specific issues
  - Drastic appearance variability through time
  - Non planar, deformable or articulated objects
  - More image quality problems: low resolution, motion blur
  - Speed/memory/causality constraints

- But …
  - Sequential image ordering is key
  - Temporal continuity of appearance
  - Temporal continuity of object state

Formalizing tracking

Image-based "measurements": $\mathbf{z}_t \in \Gamma$

- Raw or filtered images (intensities, colors, texture)
- Low-level features (edgels, corners, blobs, optical flow)
- High-level detections (e.g., face bounding boxes)

Single target "state": $\mathbf{x}_t \in \Lambda$

- Bounding box parameters (up to 6 DoF)
- 3D rigid pose (6 DoF)
- 2D/3D articulated pose (up to 30 DoF)
- 2D/3D principal deformations
- Discrete pixel-wise labels (segmentation)
- Discrete indices (activity, visibility, expression)

Formalizing tracking

- Given past and current measurements $\mathbf{z}_{1:t} := (\mathbf{z}_1 \cdots \mathbf{z}_t)$

  Output an estimate of current hidden state

$$\hat{\mathbf{x}}_t = \text{function}(\mathbf{z}_{1:t})$$

Deterministic tracking

- Optimization of ad-hoc objective function

$$\hat{\mathbf{x}}_t = \arg\min E(\mathbf{x}_t; \hat{\mathbf{x}}_{t-1}, \mathbf{z}_t)$$

  or minimization of function $E(\mathbf{x}_t; \mathbf{z}_t)$ "around" $\hat{\mathbf{x}}_{t-1}$

Probabilistic tracking

- Computation of the *filtering pdf* $p(\mathbf{x}_t|\mathbf{z}_{1:t})$ and point estimate:

$$\hat{\mathbf{x}}_t = \arg\max p(\mathbf{x}_t|\mathbf{z}_{1:t}) \text{ or } \mathbb{E}[\mathbf{x}_t|\mathbf{z}_{1:t}]$$

Probabilistic tracking

- Pros: transports full distribution knowledge
  - Takes uncertainty into account (helps with clutter, occlusions, weak model)
  - Provides some confidence assessment

- Cons
  - More computations
  - Curse of dimensionality

Probabilistic tracking

Hidden Markov chain/dynamic state space model

- Evolution model (dynamics), typically 1st-order Markov chain

$$p(\mathbf{x}_t|\mathbf{x}_{1:t-1}) = p(\mathbf{x}_t|\mathbf{x}_{t-1})$$

- Observation model

$$p(\mathbf{z}_t|\mathbf{z}_{1:t-1}, \mathbf{x}_{0:t}) = p(\mathbf{z}_t|\mathbf{x}_t)$$

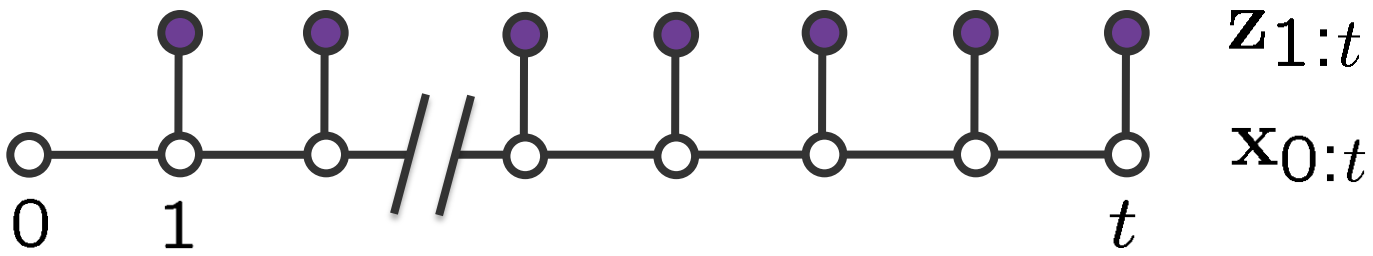
- Joint distribution

$$p(\mathbf{x}_{0:t}, \mathbf{z}_{1:t}) = p(\mathbf{x}_0) \prod_{i=1}^{t} p(\mathbf{x}_i|\mathbf{x}_{i-1}) p(\mathbf{z}_i|\mathbf{x}_i)$$

Probabilistic tracking

$$p(\mathbf{x}_{0:t}, \mathbf{z}_{1:t}) = p(\mathbf{x}_0) \prod_{i=1}^{t} p(\mathbf{x}_i|\mathbf{x}_{i-1}) p(\mathbf{z}_i|\mathbf{x}_i)$$

Associated graphical model



- Tree: exact inference with two-pass belief propagation (in theory)
- Conditional independence properties: past ⊥ future | present state

technicolor

Bayesian filtering

- Chapman-Kolmogorov recursion

$$p(\mathbf{x}_t|\mathbf{z}_{1:t}) = \frac{p(\mathbf{z}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{z}_{1:t-1})}{p(\mathbf{z}_t|\mathbf{z}_{1:t-1})}$$

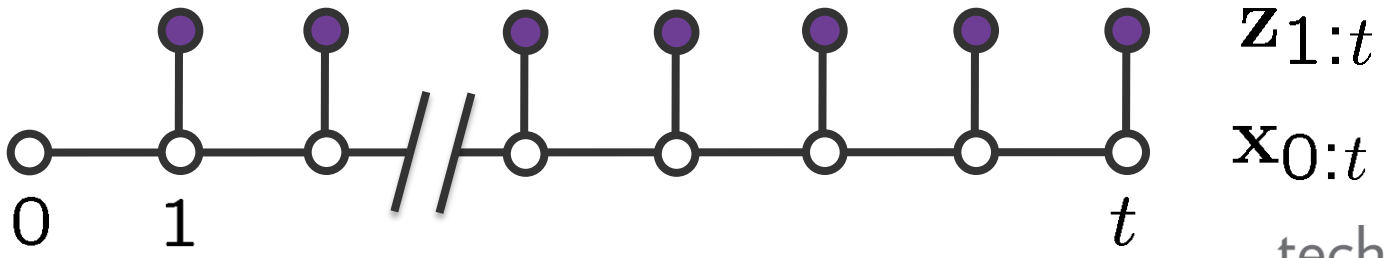
  - One step prediction

$$p(\mathbf{x}_t|\mathbf{z}_{1:t-1}) = \int p(\mathbf{x}_t|\mathbf{x}_{t-1})p(\mathbf{x}_{t-1}|\mathbf{z}_{1:t-1})d\mathbf{x}_{t-1}$$

  - Predictive likelihood

$$p(\mathbf{z}_t|\mathbf{z}_{1:t-1}) = \int p(\mathbf{z}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{z}_{1:t-1})d\mathbf{x}_t$$

- At each step: two integrals or summations (depends on state-space)



technicolor

Bayesian filtering

- Finite state space: matrix vector products classic in Markov chains
- Linear Gaussian model: close-formed solution (Kalman Filter)
- Continuous state space with mono-modal pdf: Gaussian approximations (extended Kalman Filter [EKF],unscented Kalman Filter [UKF]) propagating the two first moments
- General continuous case
  - Still Gaussian approximation (e.g, PDAF)
  - Monte Carlo approximation: particle filter

Limitation of KF and variants

- Strong limitations on observations model
  - Measurements must be of same nature as (part of) state, e.g. detected object position
  - Measurement of interest must be identified (data association problem)
- In visual tracking, especially difficult
  - State specifies which part of data is concerned (actual measurement depends on hypothesized state)
  - Clutter is frequent
- Variants of KF (extended KF, unscented KF) can help, to some extent

Particle filtering

- Monte Carlo based on *sequential importance sampling* (SIS)

- History
  - Gordon 1993, *Novel approach to non-linear/non-Gaussian Bayesian state estimation*
  - Kitagawa 1996, *Monte Carlo filter and smoother for non-Gaussian nonlinear state space models*
  - Isard et Blake 1996, *CONDENSATION: CONditional DENSity propagATION for visual tracking*

- Reasons of success in CV
  - Visual tracking often implies multimodal filtering distributions
  - PF maintains multiple hypotheses: good for robustness
  - Easy to implement and little restrictions on model ingredients

Particle filtering

- Aim: approximate posterior pdfs $p(\mathbf{x}_{0:t}|\mathbf{z}_{1:t})$ with weighted samples ('particles')

$$\{(\mathbf{x}_{0:t}^{(m)}, \pi_t^{(m)})\}_{m=1\cdots M}, \ \sum_m \pi_t^{(m)} = 1$$

- Use: for any function f on Λ^{t+1}

$$\int p(\mathbf{x}_{0:t}|\mathbf{z}_{1:t}) f(\mathbf{x}_{0:t}) d\mathbf{x}_{0:t} \approx \sum_{m=1}^{M} \pi_t^{(m)} f(\mathbf{x}_{0:t}^{(m)})$$

- In particular, approximate filtering distributions and its expectation

$$p(\mathbf{x}_t|\mathbf{z}_{1:t}) \approx \sum_m \pi_t^{(m)} \delta_{\mathbf{x}_t^{(m)}}$$

$$\int p(\mathbf{x}_t|\mathbf{z}_{1:t}) \mathbf{x}_t d\mathbf{x}_t \approx \sum_m \pi_t^{(m)} \mathbf{x}_t^{(m)}$$

Importance sampling

- Problem: sampling target pdf $p(\mathbf{x}_{0:t}|\mathbf{z}_{1:t})$ not possible
- One tool: importance sampling
  - Target distribution $p(\mathbf{x}) \propto \phi(\mathbf{x})$
  - Instrumental proposal distribution $q(\mathbf{x})$ (supp(p) ⊂ supp(q))

$$\mathbb{E}_p[f] = \frac{\mathbb{E}_q[\frac{\phi f}{q}]}{\mathbb{E}_q[\frac{\phi}{q}]}$$

  - Importance weighted samples

$$\mathbf{x}^{(m)} \sim q(\mathbf{x}),\ m = 1 \cdots M$$

$$\pi^{(m)} \propto \frac{\phi(\mathbf{x}^{(m)})}{q(\mathbf{x}^{(m)})} \text{ with } \sum_{m=1}^{M} \pi^{(m)} = 1$$

Sequential importance sampling

- Target distribution

$$p(\mathbf{x}_{0:t}|\mathbf{z}_{1:t}) \propto p(\mathbf{x}_0) \prod_{i=1}^{t} p(\mathbf{x}_i|\mathbf{x}_{i-1}) p(\mathbf{z}_i|\mathbf{x}_i)$$

- Factored proposal

$$q(\mathbf{x}_{0:t}|\mathbf{z}_{1:t}) = q(\mathbf{x}_0) \prod_{i=1}^{t} q(\mathbf{x}_i|\mathbf{x}_{i-1}, \mathbf{z}_i)$$

- Sequential sampling and weighting

$$\mathbf{x}_t^{(m)} \sim q(\mathbf{x}_t|\mathbf{x}_{t-1}^{(m)}, \mathbf{z}_t), \; m = 1 \cdots M$$

$$\pi_t^{(m)} \propto \pi_{t-1}^{(m)} \frac{p(\mathbf{z}_t|\mathbf{x}_t^{(m)}) p(\mathbf{x}_t^{(m)}|\mathbf{x}_{t-1}^{(m)})}{q(\mathbf{x}_t^{(m)}|\mathbf{x}_{t-1}^{(m)}, \mathbf{z}_t)} \text{ with } \sum_m \pi_t^{(m)} = 1$$

Resampling

- But sample pool degenerates
- Re-sampling
  - Selection mechanism (weakest samples are eliminated, strongest are duplicated) with reweighting, which preserves asymptotic properties
  - A simple method: sampling discrete distribution $\{\pi_t^{(m)}\}_{m=1\cdots M}$
- When?
  - Systematic resampling
  - Adaptive resampling based on “efficient” size as degeneracy measure

$$\frac{1}{\sum_{m=1}^{M}[\pi_t^{(m)}]^2} \leq M$$

Proposal density

- Optimal density (rarely accessible)

$$q(\mathbf{x}_t|\mathbf{x}_{t-1},\mathbf{z}_t) = p(\mathbf{x}_t|\mathbf{x}_{t-1},\mathbf{z}_t) = \frac{p(\mathbf{z}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{x}_{t-1})}{p(\mathbf{z}_t|\mathbf{x}_{t-1})}$$
$$\Rightarrow \pi_t^{(m)} \propto \pi_{t-1}^{(m)}\, p(\mathbf{z}_t|\mathbf{x}_{t-1}^{(m)}) \text{ with } \sum_m \pi_t^{(m)} = 1$$

- Bootstrap filter: classic for its simplicity

$$q(\mathbf{x}_t|\mathbf{x}_{t-1},\mathbf{z}_t) = p(\mathbf{x}_t|\mathbf{x}_{t-1})$$
$$\Rightarrow \pi_t^{(m)} \propto \pi_{t-1}^{(m)}\, p(\mathbf{z}_t|\mathbf{x}_t^{(m)}) \text{ with } \sum_m \pi_t^{(m)} = 1$$

- In-between: try and use current data for better efficiency

Generic synopsis

- Given $\{(\mathbf{x}_{0:t-1}^{(m)}, \pi_{t-1}^{(m)})\}_{m=1\cdots M}$
- One step proposal

$$\tilde{\mathbf{x}}_t^{(m)} \sim q(\mathbf{x}_t|\mathbf{x}_{t-1}^{(m)}, \mathbf{z}_t),\ m = 1 \cdots M$$

- Weights update

$$\tilde{\pi}_t^{(m)} \propto \pi_{t-1}^{(m)} \frac{p(\mathbf{z}_t|\mathbf{x}_t^{(m)})p(\mathbf{x}_t^{(m)}|\mathbf{x}_{t-1}^{(m)})}{q(\mathbf{x}_t^{(m)}|\mathbf{x}_{t-1}^{(m)}, \mathbf{z}_t)} \text{ with } \sum_{m=1}^{M} \tilde{\pi}_t^{(m)} = 1$$

- Resampling
  - If $\sum_{m=1}^{M} \tilde{\pi}_t^{(m)2} > M_{\text{seuil}}^{-1}$

$$\forall m,\ a_m \sim \sum_{k=1}^{M} \tilde{\pi}_t^{(k)} \delta_k,\ \mathbf{x}_{1:t}^{(m)} = (\mathbf{x}_{1:t-1}^{(a_m)}, \tilde{\mathbf{x}}_t^{(a_m)}) \text{ and } \pi_t^{(m)} = \frac{1}{M}$$

  - Otherwise

$$\forall m,\ \mathbf{x}_{1:t}^{(m)} = (\mathbf{x}_{1:t-1}^{(m)}, \tilde{\mathbf{x}}_t^{(m)}) \text{ and } \pi_t^{(m)} = \tilde{\pi}_t^{(m)}$$

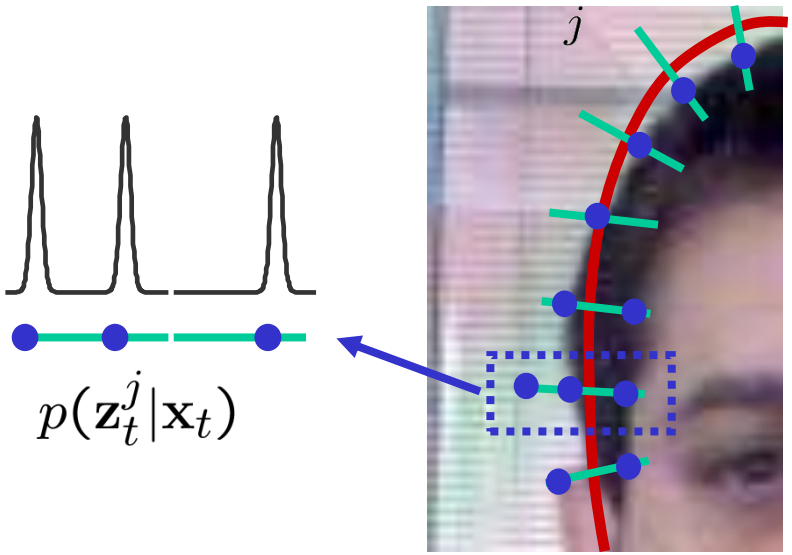
- Monte Carlo approximation

$$\mathbb{E}[f(\mathbf{x}_t)|\mathbf{z}_{1:t}] \approx \sum_{m=1}^{M} \pi_t^{(m)} f(\mathbf{x}_t^{(m)})$$

“CONDENSATION”

- State: active shape model (ASM) with autoregressive dynamics
- Observation model: based on edgels near hypothesized silhouette
- Bootstrap filter: proposal and dynamics coincide

$$p(\mathbf{z}_t|\mathbf{x}_t) = \prod_j p(\mathbf{z}_t^j|\mathbf{x}_t)$$

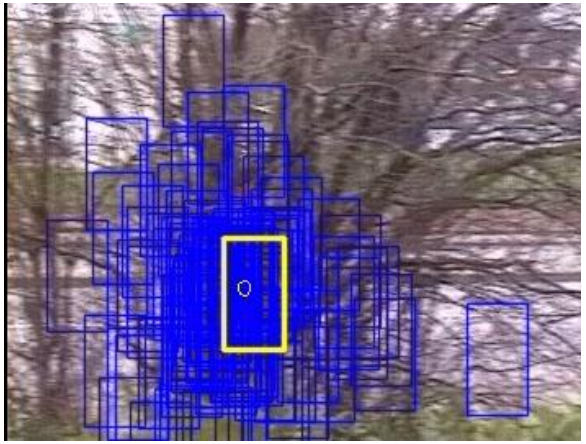


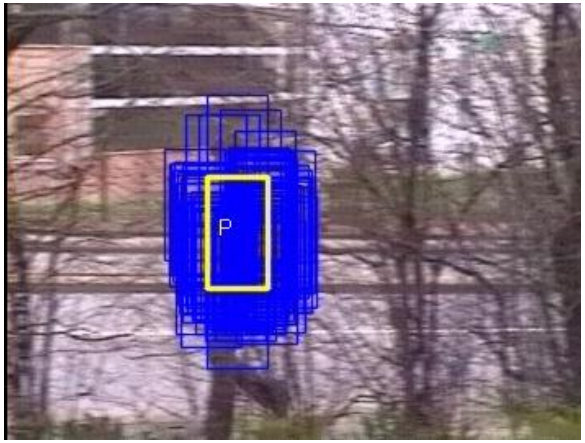
[Isard and Blake, ECCV 1996]

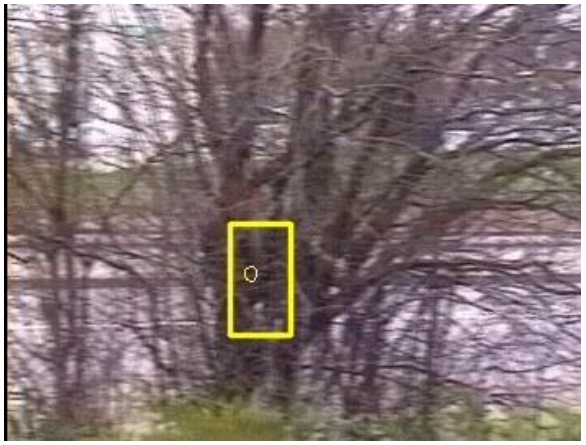
technicolor

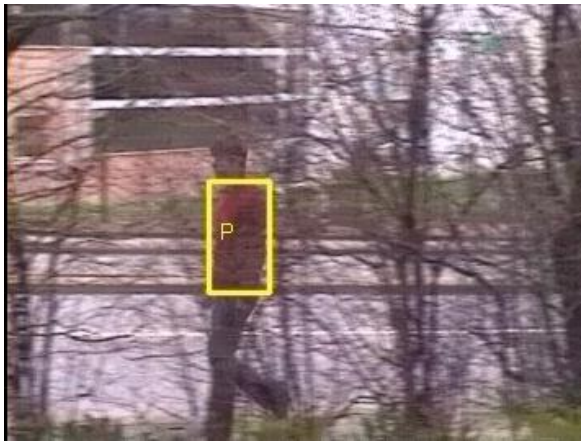
Color-based PF

- Based on color histogram similarities
- Bootstrap filter and data model $p(\mathbf{z}_t|\mathbf{x}_t) \propto \exp \lambda\rho[\mathbf{q}(\mathbf{x}_t), \mathbf{q}^*]$





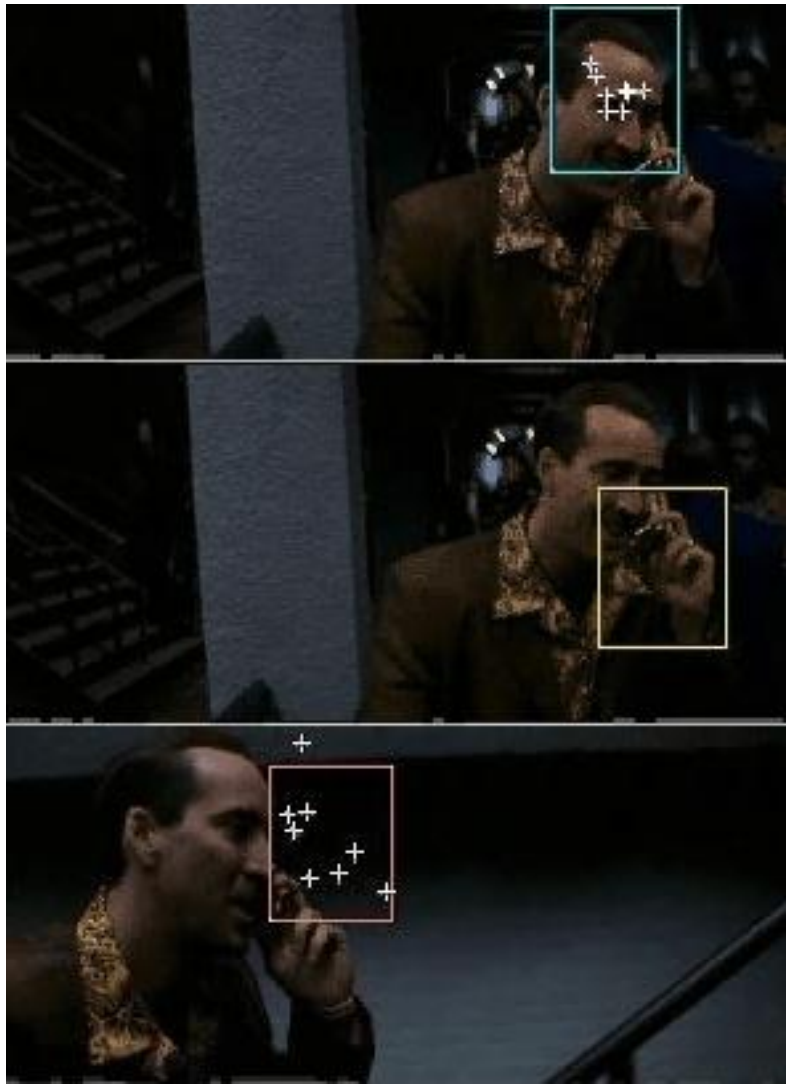
[Pérez et al. ECCV'02]

technicolor

PF with multiple cues

[Wu and Huang, ICCV'01]

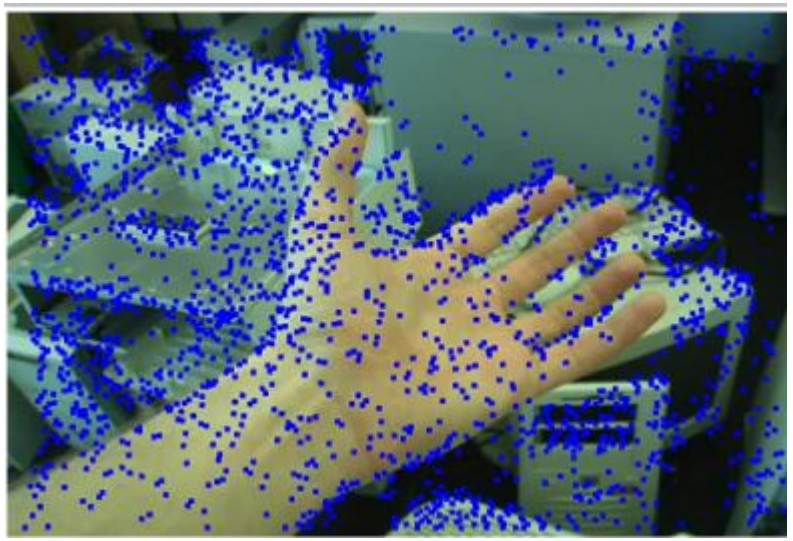
[Gatica-Perez et al., 2003]



[Badrinarayanan et al. ICCV'07]

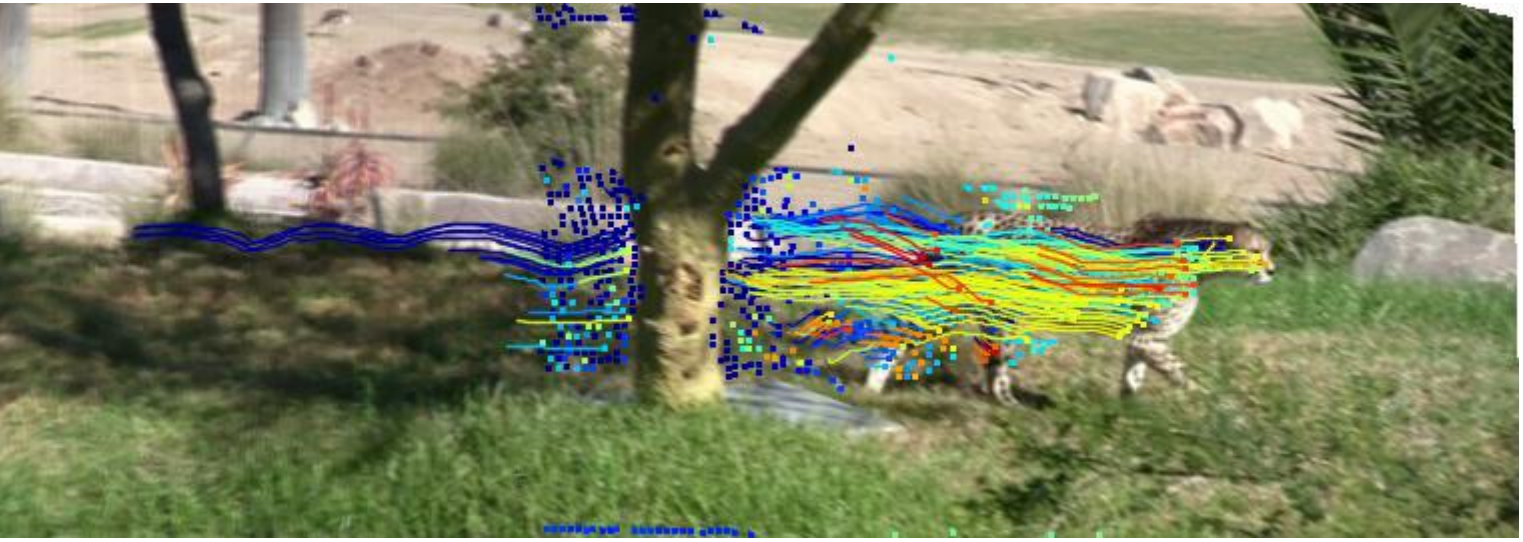
Tracking (small) fragments

- Track “key points” (Harris and the like), or random patches, as long as possible
  - Input: detected/sampled/chosen patches
  - Output: *tracklets* of various life-spans



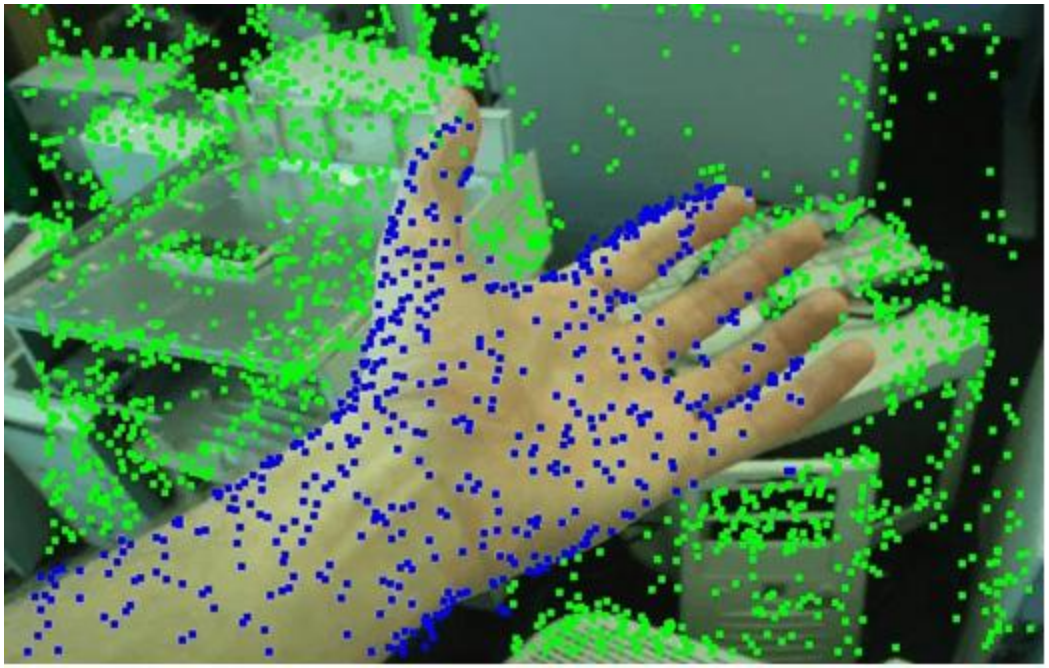
[Sand and Teller CVPR 2006]

[Rubinstein et al. BMVC12]



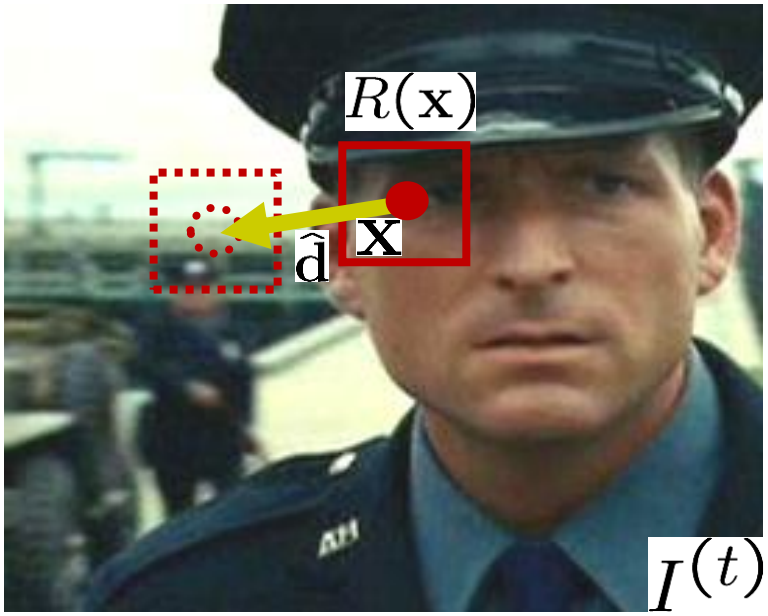
Use of tracklets

- Structure-from-motion and camera pose tracking
- Video segmentation into objects
- Video indexing and copy detection
- Action synchronization and recognition
- Fragment-based object grouping and tracking



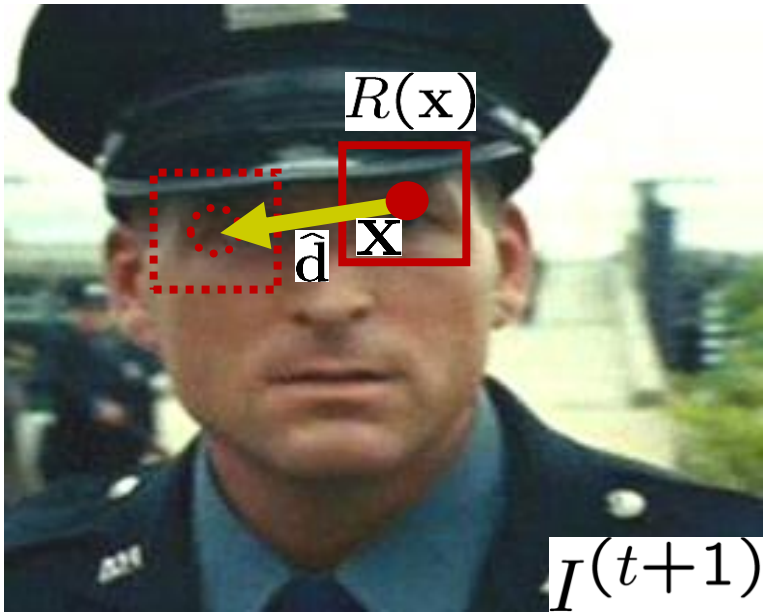
[Fradet *et al*. CVMP'09]

Point tracking



$$\hat{\mathbf{d}} = \arg\min_{\mathbf{d}} \underbrace{\sum_{\mathbf{p}\in R(\mathbf{x})} |I^{(t+1)}(\mathbf{p}+\mathbf{d}) - I^{(t)}(\mathbf{p})|^2}_{\text{SSD}}$$

Point tracking



$$\hat{\mathbf{d}} = \arg\min_{\mathbf{d}} \underbrace{\sum_{\mathbf{p}\in R(\mathbf{x})} |I^{(t+1)}(\mathbf{p}+\mathbf{d}) - I^{(t)}(\mathbf{p})|^2}_{\text{SSD}}$$

KLT (Kanade-Lucas-Tomasi)

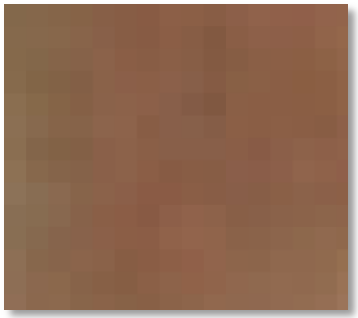
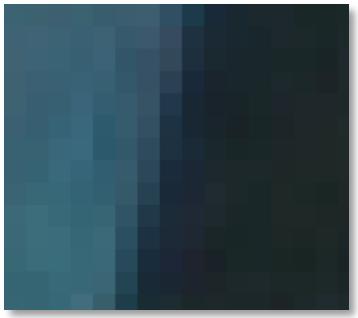
- Assuming small displacement: 1st-order Taylor expansion *inside SSD*

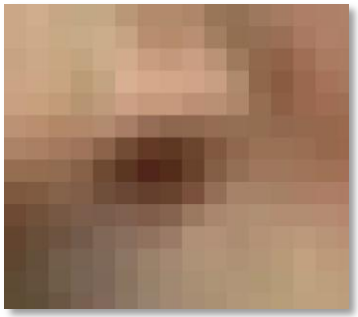
$$\hat{\mathbf{d}} = \arg\min_{\mathbf{d}} \sum_{\mathbf{p}\in R(\mathbf{x})} |I^{(t+1)}(\mathbf{p}) + \nabla I^{(t+1)}(\mathbf{p})^{\mathrm{T}}\mathbf{d} - I^{(t)}(\mathbf{p})|^2$$

$$\hat{\mathbf{d}} = -\left(\sum_{\mathbf{p}\in R(\mathbf{x})} \nabla I(\mathbf{p})\nabla I(\mathbf{p})^{\mathrm{T}}\right)^{-1} \sum_{\mathbf{p}\in R(\mathbf{x})} \nabla I(\mathbf{p}) I_t(\mathbf{p})$$

For good conditioning, patch must be textured/structured enough:

- Uniform patch: no information
- Contour element: aperture problem (one dimensional information)
- Corners, blobs and texture: best estimate



[Lucas and Kanade 1981][Tomasi and Shi, CVPR'94]

Monitoring quality

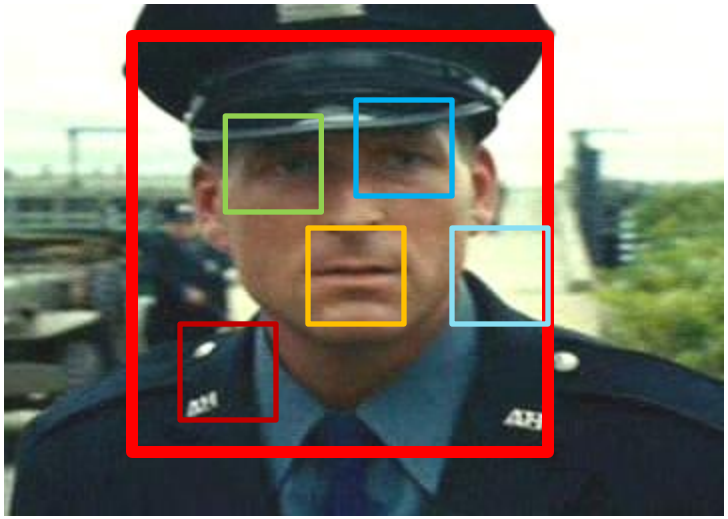
- Translation is usually sufficient for small fragments, but:
  - Perspective transforms and occlusions cause drift and loss
- Two complementary options
  - Kill tracklets when minimum SSD too large
  - Compare as well with *initial patch under affine transform (warp)* assumption

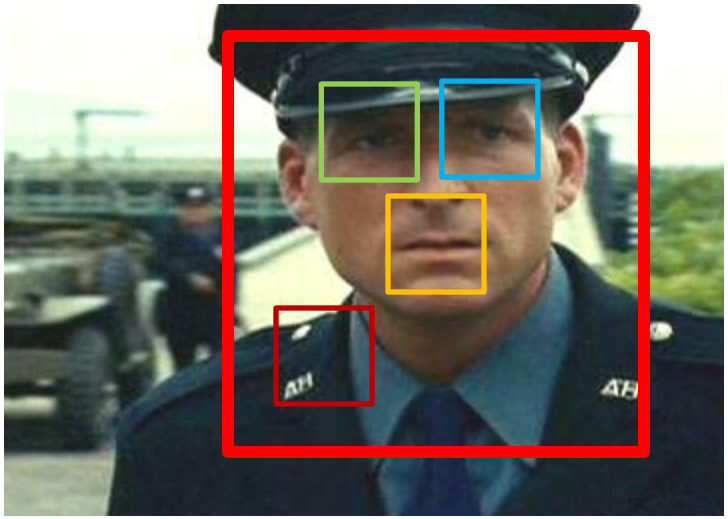
$$\hat{\mathbf{d}} = \arg\min_{\mathbf{d}} \sum_{\mathbf{p} \in R(\mathbf{x}_t)} |I^{(t+1)}(\mathbf{p}+\mathbf{d}) - I^{(t)}(\mathbf{p})|^2$$

$$\hat{\mathbf{w}} = \arg\min_{\mathbf{w}} \sum_{\mathbf{p} \in R(\mathbf{x}_0)} |I^{(t+1)}(\mathbf{w}[\mathbf{p}]) - I^{(0)}(\mathbf{p})|^2$$

Larger fragment as collection

- Track in next frame fragments from current *bounding box*
- Terminate weak tracklets
- Infer global motion of bounding box
- Select new points if necessary
- In effect: *part-based adaptive appearance model*





$$\hat{\mathbf{x}}_{t+1} = \hat{\mathbf{x}}_t + \text{robust average}(\mathbf{d}_1 \cdots \mathbf{d}_{n_t})$$

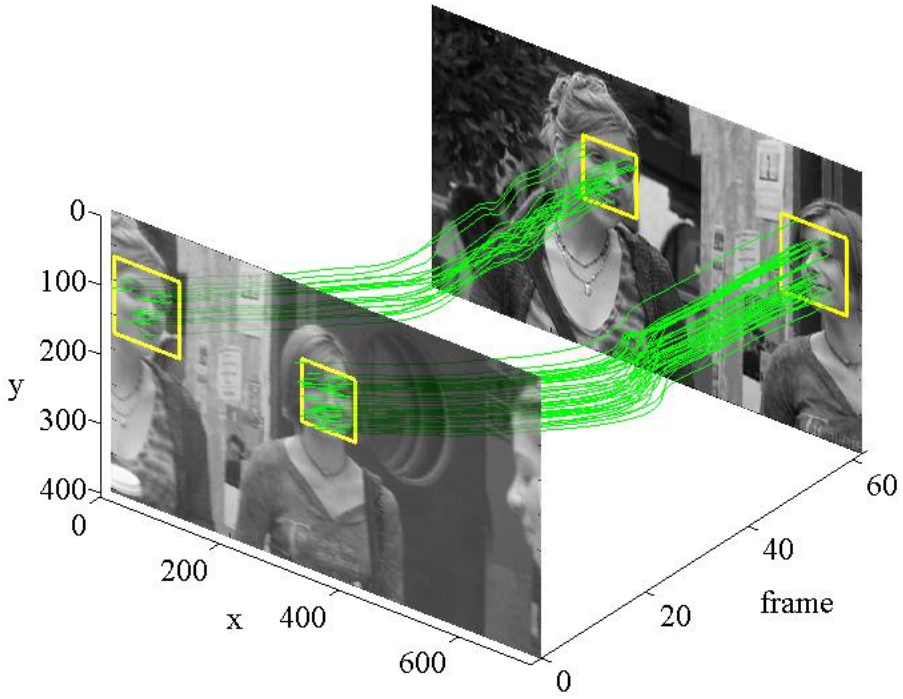
Larger fragment as collection

- Can work really well and fast


- Until
  - It drifts (due to partial occlusion, out-of-plane rotation)
  - It breaks down (diverging drift, total occlusion)


Linking detections with tracklets

- *Detect* objects of interest in each frame
- Connect instances traversed by sufficient fraction of tracklets
- Yet another detect-before-track approach



http://www.robots.ox.ac.uk/~vgg/research/nface/

technicolor

Holistic tracking of arbitrary “objects”

- Extend point tracking to whole region
- Assume a *reference image template* is available $[I^*(\mathbf{p})]_{\mathbf{p}\in R^*}$
- Search for best wrap of *reference image template*

$$\hat{\mathbf{x}}_{t+1} = \arg\min_{\mathbf{w}} \sum_{\mathbf{p}\in R^*} |I^{(t+1)}(\mathbf{w}[\mathbf{p}]) - I^*(\mathbf{p})|^2$$

  - Multi-scale Gauss-Newton around previous wrap

technicolor

Reference template

- Two extreme choices
  - Short term memory: reference = last object instance

$$I^*(\mathbf{p}) := I^{(t)}(\hat{\mathbf{x}}_t[\mathbf{p}]),\ \mathbf{p} \in R^*$$

    Same pros and cons as point tracking
  - Long term memory: reference = initial object instance

$$I^*(\mathbf{p}) := I^{(0)}(\mathbf{x}_0[\mathbf{p}]),\ \mathbf{p} \in R^*$$

    Even with affine, often not robust enough to illumination/pose changes…

technicolor

Reference template

- Two extreme choices
  - Short term memory: reference = last object instance

$$I^*(\mathbf{p}) := I^{(t)}(\hat{\mathbf{x}}_t[\mathbf{p}]),\ \mathbf{p} \in R^*$$

    Same pros and cons as point tracking
  - Long term memory: reference = initial object instance

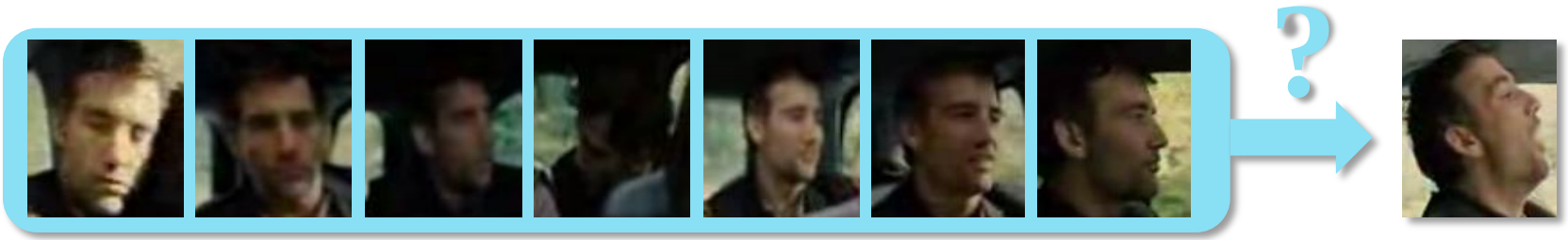
$$I^*(\mathbf{p}) := I^{(0)}(\mathbf{x}_0[\mathbf{p}]),\ \mathbf{p} \in R^*$$

    Even with affine, often not robust enough to illumination/pose changes…

technicolor

Toward improved robustness

- Enrich the holistic model and *update on-line*



- Looser appearance modeling via spatial aggregation
  - No (or loose) layout information
  - Color or texture statistics
  - Adaptation might not be necessary
  - "Mean-shift" tracker [Comaniciu *et al*. 2001]
    - Color histogram
    - Spatial kernel
  - Again: iterative Gauss-Newton descent

technicolor

Color-based tracking

- Global description of tracked region: color histogram
- Reference histogram with *B* bins

$$\mathbf{q}^* = (q_u^*)_{u=1\cdots B}$$

set at track initialization

- Candidate histogram at current instant

$$\mathbf{q}(\mathbf{x}) = (q_u(\mathbf{x}))_{u=1\cdots B}$$

gathered in region $R(\mathbf{x})$ of current image.

- At each instant

$$\hat{\mathbf{x}}_{t+1} = \arg\min_{\mathbf{x}} \text{dist}(\mathbf{q}^*, \mathbf{q}(\mathbf{x}))$$

  - searched around $\hat{\mathbf{x}}_t$
  - iterative search initialized with $\hat{\mathbf{x}}_t$: meanshift-like iteration

technicolor

Color-based tracking

- Global description of tracked region: color histogram
- Reference histogram with *B* bins

$$\mathbf{q}^* = (q_u^*)_{u=1\cdots B}$$

set at track initialization

- Candidate histogram at current instant

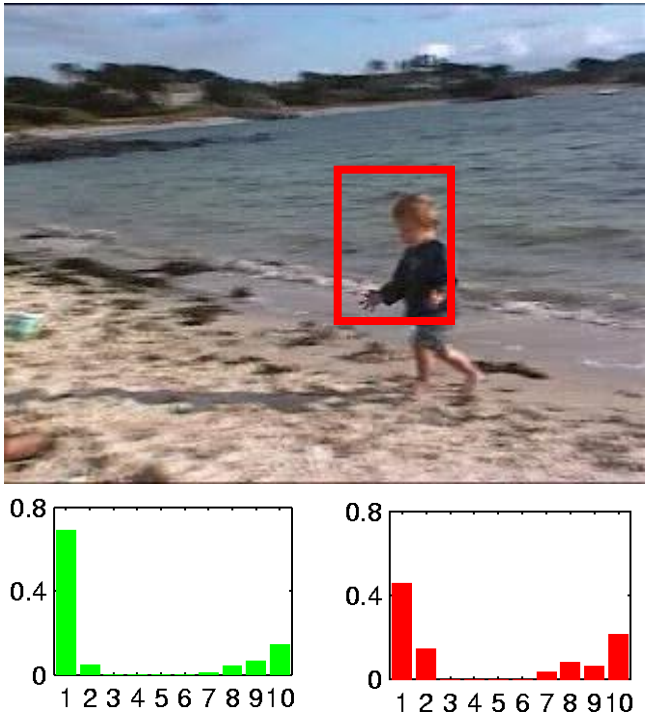
$$\mathbf{q}(\mathbf{x}) = (q_u(\mathbf{x}))_{u=1\cdots B}$$

gathered in region $R(\mathbf{x})$ of current image.

- At each instant

$$\hat{\mathbf{x}}_{t+1} = \arg\min_{\mathbf{x}} \text{dist}(\mathbf{q}^*, \mathbf{q}(\mathbf{x}))$$

  - searched around $\hat{\mathbf{x}}_t$
  - iterative search initialized with $\hat{\mathbf{x}}_t$: meanshift-like iteration

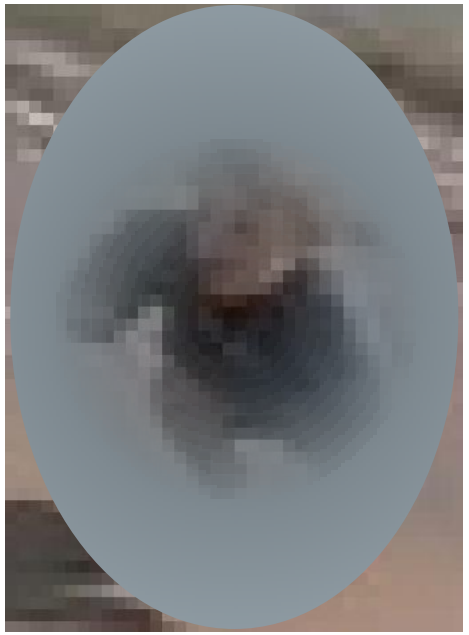


technicolor

Color distributions and similarity

- Color histogram weighted by a kernel
  - Kernel elliptic support sits on the object
  - Central pixels contribute more
  - Makes differentiation possible

$$q_u(\mathbf{x}) \propto \sum_{\mathbf{p}_i \in R(\mathbf{x})} k\left(\|\mathbf{p}_i - \mathbf{x}\|^2_{H^{-1}}\right) \mathbf{1}[I(\mathbf{p}_i) \in b_u]$$



  - *H*: "bandwidth" sym. def. pos. matrix, related to bounding box dimensions
  - *k*: "profile" of kernel (Gaussian or Epanechnikov)
- Histogram dissimilarity measure
  - Battacharyya measure $\text{dist}(\mathbf{q}^*, \mathbf{q}(\mathbf{x}))^2 = 1 - \sum_u \sqrt{q^*_u q_u(\mathbf{x})} = 1 - \rho[\mathbf{q}^*, \mathbf{q}(\mathbf{x})]$
  - Symmetric, bounded, null only for equality
  - 1 - dot product on positive quadrant of unitary hyper-sphere

Iterative ascent

$$\hat{\mathbf{x}}_{t+1} = \arg\max_{\mathbf{x}} \sum_u \sqrt{q_u^* q_u(\mathbf{x})}$$

$$q_u(\mathbf{x}) \propto \sum_{\mathbf{p}_i} k\left(\|\mathbf{p}_i - \mathbf{x}\|^2_{H^{-1}}\right) \mathbf{1}[I(\mathbf{p}_i) \in b_u]$$

- Non quadratic minimization: iterative ascent with linearizations

  u_i bin index of pixel i: $I(\mathbf{p}_i) \in b_{u_i}$

$$\nabla \sum_u \sqrt{q_u^* q_u(\mathbf{x})} \propto H^{-1} \sum_{\mathbf{p}_i} \sqrt{\frac{q_{u_i}^*}{q_{u_i}(\mathbf{x})}} k'\left(\|\mathbf{p}_i - \mathbf{x}\|^2_{H^{-1}}\right)(\mathbf{x}-\mathbf{p}_i)$$

- Setting move to (*g*=-*h*')

$$\frac{\sum_{\mathbf{p}_i} \sqrt{\frac{q_{u_i}^*}{q_{u_i}(\mathbf{x})}} g\left(\|\mathbf{p}_i - \mathbf{x}\|^2_{H^{-1}}\right)(\mathbf{p}_i - \mathbf{x})}{\sum_{\mathbf{p}_i} \sqrt{\frac{q_{u_i}^*}{q_{u_i}(\mathbf{x})}} g\left(\|\mathbf{p}_i - \mathbf{x}\|^2_{H^{-1}}\right)} = \text{MeanShift}(\mathbf{x}) - \mathbf{x}$$

  yields a simple algorithm…

Meanshift tracker

In frame *t*+1

- Start search at $\mathbf{y}^{(0)} = \hat{\mathbf{x}}_t$
- Until stop
  - Compute candidate histogram $\mathbf{q}(\mathbf{y}^{(n)})$
  - Weight pixels inside kernel support

$$\forall \mathbf{p}_i \in R(\mathbf{y}^{(n)}),\ w_i \propto \sqrt{\frac{q^*_{u_i}}{q_{u_i}(\mathbf{y}^{(n)})}} g\left(\|\mathbf{p}_i - \mathbf{y}^{(n)}\|^2_{H^{-1}}\right),\ \sum_i w_i = 1$$

  - Move kernel

$$\mathbf{y}^{(n+1)} = \mathbf{y}^{(n)} + [\text{MeanShift}(\mathbf{y}^{(n)}) - \mathbf{y}^{(n)}] = \textstyle\sum_{\mathbf{p}_i \in R(\mathbf{y}^{(n)})} w_i \mathbf{p}_i$$

  - Check overshooting
    until $\rho[\mathbf{q}^*, \mathbf{p}(\mathbf{y}^{(n+1)})] < \rho[\mathbf{q}^*, \mathbf{p}(\mathbf{y}^{(n)})],\ \mathbf{y}^{(n+1)} \leftarrow \frac{\mathbf{y}^{(n)} + \mathbf{y}^{(n+1)}}{2}$

  - If $\|\mathbf{y}^{(n+1)} - \mathbf{y}^{(n)}\|^2 < \varepsilon$ stop, else $n \leftarrow n + 1$
- $\hat{\mathbf{x}}_{t+1} = \mathbf{y}^{(n+1)}$

Examples



http://comaniciu.net/




technicolor

Pros and cons

- Low computational cost (easily faster than real-time)
- Surprisingly robust
  - Invariant to pose and viewpoint
  - Often no need to update reference color model
- Invariance comes at a price
  - Position estimate prone to fluctuation
  - Scale and orientation not well captured
  - Sensitive to color clutter (e.g., teammates in team sports)
- Deterministic local search challenged by
  - abrupt moves
  - occlusions

On-line adaptation

- When tracking arbitrary “objects”, appearance model is key
  - Initialized and kept fixed: requires simple modeling for robustness at cost of discriminative power
  - Obtained at previous instant: works very well until it drifts and fails
  - All sorts of mixes of these two
- Even with strong prior
  - Need for appearance model personalization, esp. for multi-object tracking
- More classic: online parameter estimation of generative model
- More recent trend: *on-line learning* (of appearance)

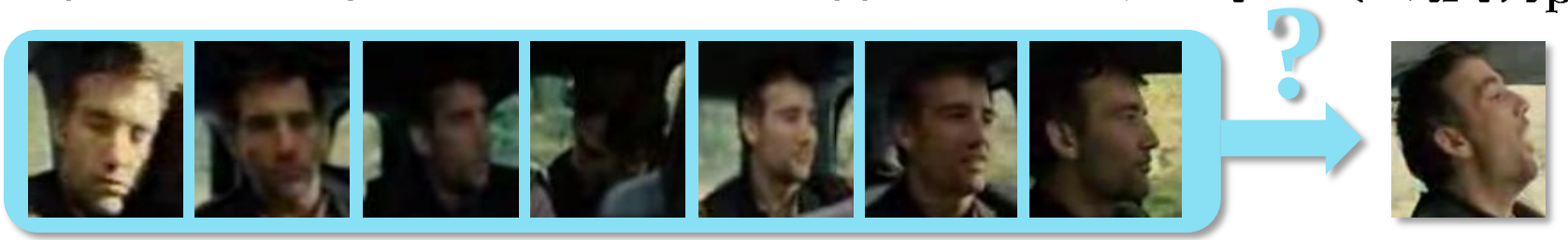
On-line learning

- Use current data to adapt model and infer new position
  - Descriptive modeling: compact model of pixel-wise appearance, plugged into deterministic or probabilistic tracking
  - Discriminative modeling (tracking-by-detection): learn and apply a detector or predictor that discriminates object from background around previous position

- Challenges
  - What are training data? Are they labeled? How?
  - How to avoid drift and to circumvent occlusions?
  - How to control complexity over time?

On-line descriptive learning

- Exploit tracking results to describe appearance $\mathbf{z}_t = [I^{(t)}(\mathbf{x}_t[\mathbf{p}])]_{\mathbf{p}\in R^*}$



- *Marginal* pixel modeling: one intensity pdf per pixel

$$E(\mathbf{x}_t; \mathbf{z}_t, \mathcal{M}_t) = -\sum_{\mathbf{p}\in R^*} \ln p(\mathbf{z}_t(\mathbf{p}); \theta_t(\mathbf{p}))$$

- *Joint* modeling: some compact model (quantized, thin or sparse)

$$E(\mathbf{x}_t; \mathbf{z}_t, \mathcal{M}_t) \propto \|\mathbf{z}_t - \overbrace{q(\mathbf{z}_t; \mathcal{M}_t)}^{\text{approximation}}\| \quad \text{reconst. error}$$

- Update model

$$\{\mathcal{M}_t, \hat{\mathbf{x}}_t, I^{(t)}\} \rightarrow \mathcal{M}_{t+1}$$

technicolor

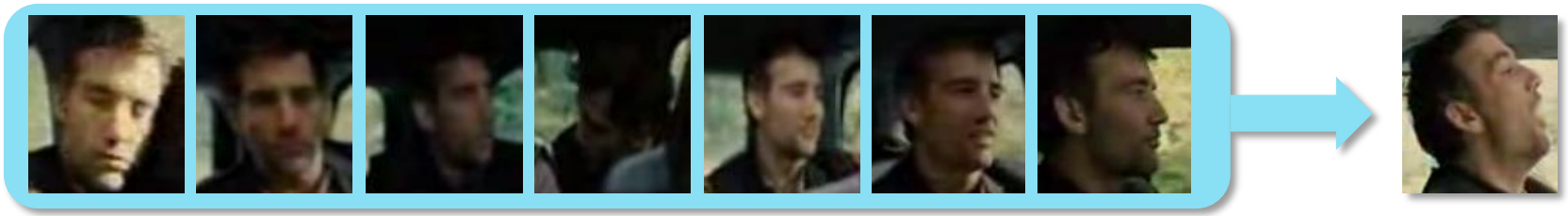
Pixel-wise “RWS” model

- Three-fold mixture per pixel
  - [R]andom component: occlusion, unpredictable changes
  - [W]andering component: rapid changes
  - [S]table component: slow changes
- On-line EM to update mixtures
- Deterministic search for tracking



[Jepson et al. PAMI 25(10), 2003]

On-line joint model



$$E(\mathbf{x}_t; \mathbf{z}_t, \mathcal{M}_t) \propto \|\mathbf{z}_t - q(\mathbf{z}_t; \mathcal{M}_t)\|^2$$

- Match to a catalogue of "exemplars" $C_t = [\mathbf{c}_1 \cdots \mathbf{c}_k],\ k < d$

$$E(\mathbf{x}_t; \mathbf{z}_t, \mathcal{M}_t) = \min_{\mathbf{c}_k \in C_t} \|\mathbf{z}_t - \mathbf{c}_k\|^2$$

- PCA with mean $\boldsymbol{\mu}_t$, basis $U_t = [\mathbf{u}_1 \cdots \mathbf{u}_k],\ k \ll d$

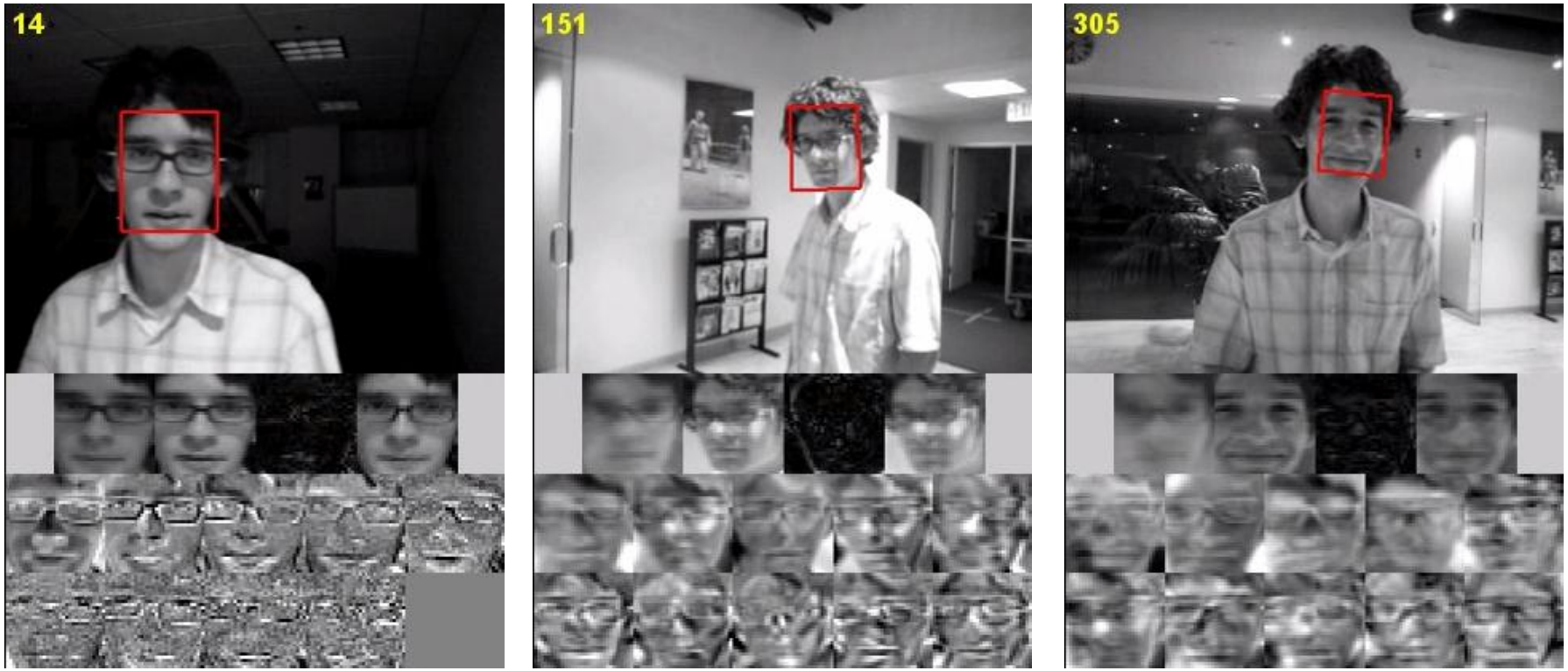
$$E(\mathbf{x}_t; \mathbf{z}_t, \mathcal{M}_t) = \|\mathbf{z}_t - \boldsymbol{\mu}_t - U_t U_t^{\mathrm{T}}(\mathbf{z}_t - \boldsymbol{\mu}_t)\|^2$$

- Sparse coding with dictionary of atoms $D_t = [\mathbf{d}_1 \cdots \mathbf{d}_k],\ k \gg d$

$$E(\mathbf{x}_t; \mathbf{z}_t, \mathcal{M}_t) = \min_{\mathbf{a}} \|\mathbf{z}_t - D_t\mathbf{a}\|^2 + \|\mathbf{a}\|_1$$

On-line subspace learning

- Constant time PCA update with new data, with *learning rate* $\alpha \sim 0.02$
- "Robust" norm to account for background corruption
- Tracking with particle filter

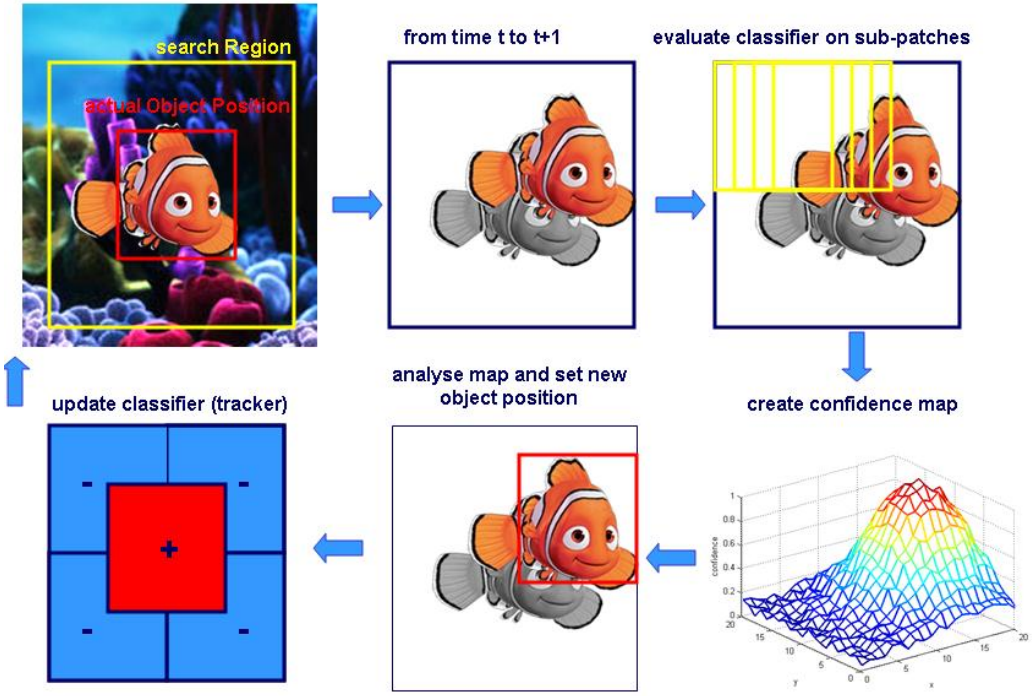


[Ross *et al*. IJCV 2008]

technicolor

On-line discriminative learning

- Instead of learning appearance of object, learn how to discriminate it from the background: *tracking-by-detection*
- Online *supervised* learning



[Grabner and Bischof CVPR 06]

technicolor

On-line supervised learning

- Sub-image descriptor: $\mathbf{z}_t(\mathbf{x}) = \phi\left([I^{(t)}(\mathbf{x}+\mathbf{p})]_{\mathbf{p}\in R^*}\right)$

- Online supervised learning
  - New positive example: $\mathbf{z}_t(\hat{\mathbf{x}}_t)$
  - New negative examples: $\{\mathbf{z}_t(\mathbf{x}), \mathbf{x}$ "around" $\hat{\mathbf{x}}_t\}$
  - Update classifier: $f_t \rightarrow f_{t+1}$

- Next detection:

$$\hat{\mathbf{x}}_{t+1} = \arg\max_{\mathbf{x}\in W(\hat{\mathbf{x}}_t)} f_{t+1}[\mathbf{z}_{t+1}(\mathbf{x})]$$

search window

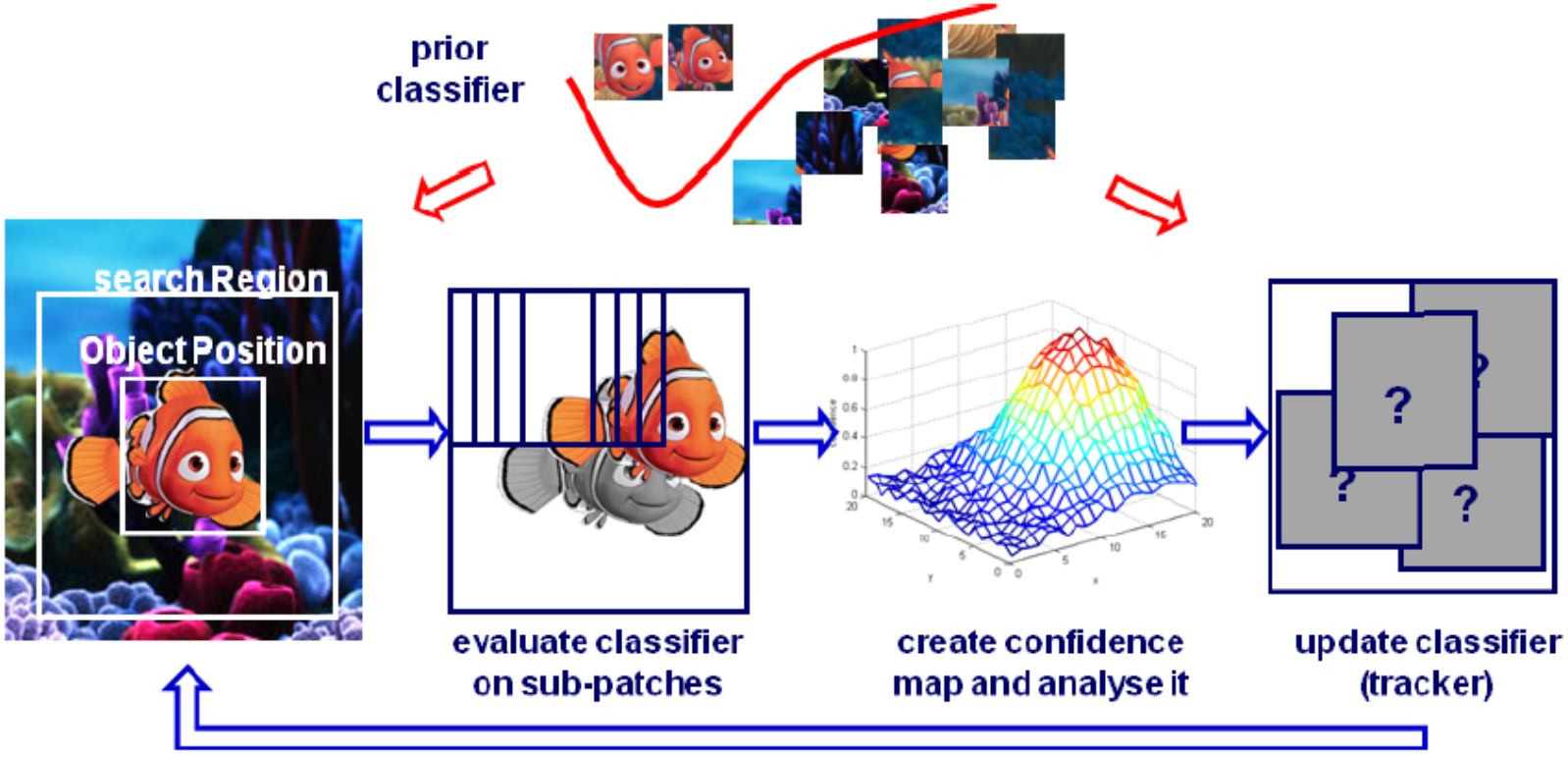
$$\hat{\mathbf{d}}_{t+1} = \arg\max_{\mathbf{d}\in W} f_{t+1}[\mathbf{z}_{t+1}(\hat{\mathbf{x}}_t+\mathbf{d})]$$

range window

- Problem: tracker inaccuracy ⇒ label noise ⇒ tracker drift

On-line semi-supervised boosting

- Only initial examples labeled (‘prior’)
- All other examples, unlabeled



[Gragner et al. ECCV 08]

technicolor

On-line semi-supervised boosting

[Gragner et al. ECCV 08]

technicolor

STRUCK [Hare et al. ICCV 11]

- Extend to tracking [Blascko and Lampert ECCV 08]
- Closer to actual task: learn function $F_t(\mathbf{z}, \mathbf{d})$ such that

$$\hat{\mathbf{d}}_{t+1} = \arg\max_{\mathbf{d} \in W} F_t[\mathbf{z}_{t+1}(\hat{\mathbf{x}}_t), \mathbf{d}]$$

- Kernelized structured output SVM:

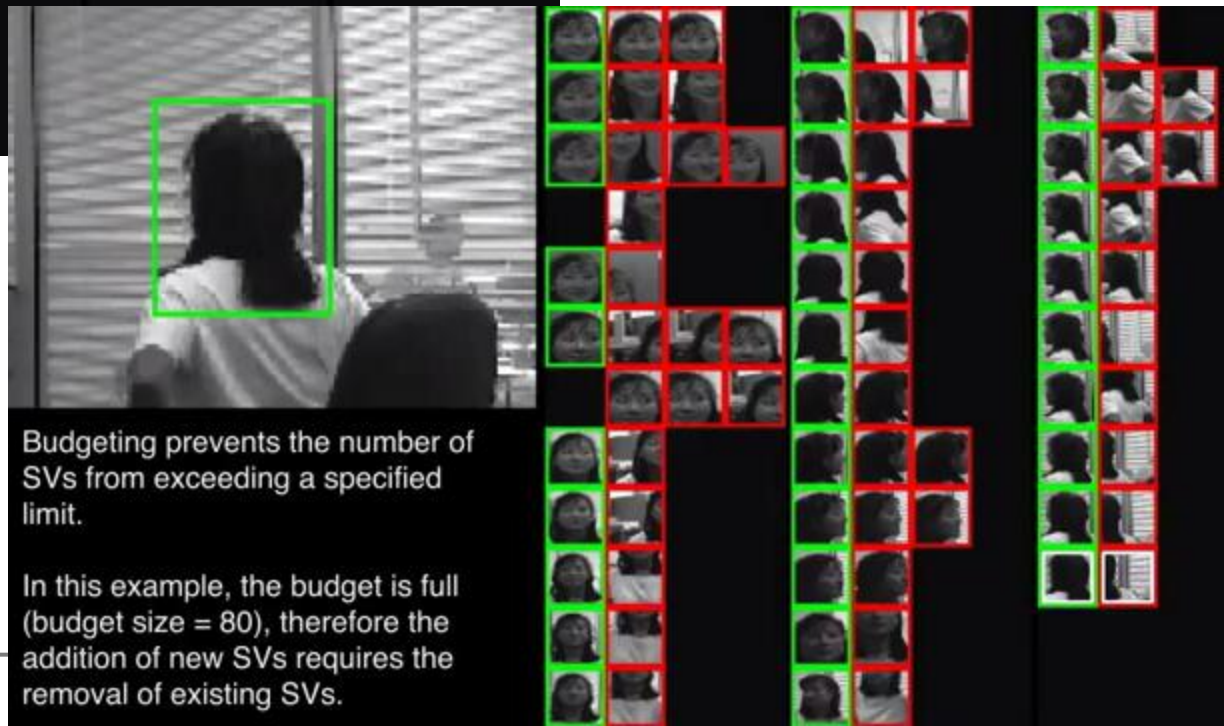
$$F_t(\mathbf{z}, \mathbf{d}) = \mathbf{w}_t^{\mathsf{T}} \mathbf{\Phi}(\mathbf{z}, \mathbf{d}),$$

$$\Delta(\mathbf{d}, \mathbf{d}') = 1 - \frac{|R(\mathbf{d}) \cap R(\mathbf{d}')|}{|R(\mathbf{d}) \cup R(\mathbf{d}')|}$$

- Budgeting support vectors

STRUCK

SVs are not only added from the most recent frame, they can also be added from previous frames for which SVs have been kept.



Budgeting prevents the number of SVs from exceeding a specified limit.

In this example, the budget is full (budget size = 80), therefore the addition of new SVs requires the removal of existing SVs.

Tracking-Learning-Detection

- Hybrid approach: short-term tracking and detection are distinct
- Monitor both to
  - Output new estimated position (or declare loss)
  - Select new samples for detector update

[Kalal et al., PAMI 2010]

technicolor

Current trends

- Leverage cutting-edge ML tools
  - sparse appearance modeling
  - discriminative learning

- Exploitation of *context*
  - “supporters” and “distractors”
  - leveraging scene understanding
    - geometry
    - pixel-wise semantics
    - interaction between scene elements
  - Joint tracking/recognition (action, attributes, etc.)

Some bottlenecks and directions

- Very high-dim tracking
  - Dense MOT
  - Highly articulated and/or deformable
  - Pixel-wise discrete/continuous variables

- Online adaptation/learning
  - Caution: a double side sword
  - Complementary multiple cues:
    - Anchored parameter estimation
    - Co-training

A new resource

Visual Tracker Benchmark (29 trackers, 50 recent sequences)) [Wu et al. CVPR'13]

http://cvlab.hanyang.ac.kr/wordpress/?page_id=14

CPF	P. Pérez, C. Hue, J. Vermaak, and M. Gangnet. Color-Based Probabilistic Tracking. ECCV, 2002.
KMS	D. Comaniciu, V. Ramesh, and P. Meer. Kernel-Based Object Tracking. PAMI, 25(5):564–577, 2003.
SMS	R. Collins. Mean-shift Blob Tracking through Scale Space. CVPR, 2003.
VIVID/VR	R. T. Collins, Y. Liu, and M. Leordeanu. Online Selection of Discriminative Tracking Features. PAMI, 27(10):1631–1643, 2005
Frag	A. Adam, E. Rivlin, and I. Shimshoni. Robust Fragments-based Tracking using the Integral Histogram. CVPR, 2006.
OAB	H. Grabner, M. Grabner, and H. Bischof. Real-Time Tracking via On-line Boosting. BMVC, 2006.
IVT	D. Ross, J. Lim, R.-S. Lin, and M.-H. Yang. Incremental Learning for Robust Visual Tracking. IJCV, 77(1):125–141, 2008.
SBT	H. Grabner, C. Leistner, and H. Bischof. Semi-supervised On-Line Boosting for Robust Tracking. ECCV, 2008.
MIL	B. Babenko, M.-H. Yang, and S. Belongie. Visual Tracking with Online Multiple Instance Learning. CVPR, 2009.
BSBT	S. Stalder, H. Grabner, and L. van Gool. Beyond Semi-Supervised Tracking: Tracking Should Be as Simple as Detection, but not Simpler than Recognition. In ICCV Workshop, 2009.
TLD	Z. Kalal, J. Matas, and K. Mikolajczyk. P-N Learning: Bootstrapping Binary Classifiers by Structural Constraints. CVPR, 2010.
-	J. Kwon and K. M. Lee. Visual Tracking Decomposition. CVPR, 2010.
CXT	T. B. Dinh, N. Vo, and G. Medioni. Context Tracker: Exploring supporters and distracters in unconstrained environments. CVPR, 2011.
LSK	B. Liu, J. Huang, L. Yang, and C. Kulikowsk. Robust Tracking using Local Sparse Appearance Model and K-Selection. CVPR, 2011.
Struck	S. Hare, A. Saffari, and P. H. S. Torr. Struck: Structured Output Tracking with Kernels. ICCV, 2011.
-	J. Kwon and K. M. Lee. Tracking by Sampling Trackers. ICCV, 2011.
ASLA	X. Jia, H. Lu, and M.-H. Yang. Visual Tracking via Adaptive Structural Local Sparse Appearance Model. CVPR, 2012.
DFT	L. Sevilla-Lara and E. Learned-Miller. Distribution Fields for Tracking. CVPR, 2012.
L1APG	C. Bao, Y. Wu, H. Ling, and H. Ji. Real Time Robust L1 Tracker Using Accelerated Proximal Gradient Approach. CVPR, 2012.
LOT	S. Oron, A. Bar-Hillel, D. Levi, and S. Avidan. Locally Orderless Tracking. CVPR, 2012.
MTT	T.Zhang, B. Ghanem,S. Liu,and N. Ahuja. Robust Visual Tracking via Multi-task Sparse Learning. CVPR, 2012.
ORIA	Y. Wu, B. Shen, and H. Ling. Online Robust Image Alignment via Iterative Convex Optimization. CVPR, 2012.
SCM	W. Zhong, H. Lu, and M.-H. Yang. Robust Object Tracking via Sparsity-based Collaborative Model. CVPR, 2012.
CSK	F. Henriques, R. Caseiro, P. Martins, and J. Batista. Exploiting the Circulant Structure of Tracking-by-Detection with Kernels. ECCV, 2012.
CT	K. Zhang, L. Zhang, and M.-H. Yang. Real-time Compressive Tracking. ECCV, 2012.

Reviews, tutorials

Computer vision: a modern approach, Chapter 19, Forsyth and Ponce

Object tracking: a survey, Yilmaz *et al*. 2006

http://vision.eecs.ucf.edu/papers/Object%20Tracking.pdf

A review of visual tracking, Cannons, 2008

http://www.cse.yorku.ca/techreports/2008/CSE-2008-07.pdf

Recent advances and trends in visual tracking: A review, Yang *et al*., 2011

http://210.75.252.83/bitstream/344010/6218/1/110201.pdf

Lucas-Kanade 20 years on: a unifying framework, Barker and Matthews, 2004

http://www.cs.cmu.edu/afs/cs/academic/class/15385-s12/www/lec_slides/Baker&Matthews.pdf

A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking, MS Arulampalam et al., 2002

http://www.dis.uniroma1.it/~visiope/Articoli/ParticleFilterTutorial.pdf

On sequential Monte Carlo sampling methods for Bayesian filtering, Doucet *et al*. 2000

http://www-sigproc.eng.cam.ac.uk/~sjg/papers/99/statcomp_final.ps