CMPUT 615 student reading and in-class presentation

Purpose:

- Learn how to research for information about robotics projects
  – Search web, library books and research papers
  – Skim through many pages. Find a few sources of core in
  – Read core in detail

- Learn how to summarize survey information.
  – What is the common goal of projects/papers?
  – Do they share techniques? Do some use different techniqies for the same goal?
  – Unify notation and make it into a survey.
- Make an interesting presentation for your classmates:
  – Use visuals: images, diagrams, videos (google image and video search)
- Get some practice for the course project and proposal.

CMPUT 615 student reading and in-class presentation

- Presentations of vision literature or readings projects from web pages.
- The presentation is done individually. Each student books a 20 minute slot
- The presentation can focus on a paper, a project web page, or be a summary of a several papers/projects. Some visuals are expected, e.g. images and videos you find on the web.
- Find a title/topic, and list some sources, or give a web link to a web page you make with a source list.
  List not required at signup. You can add the resources as you go along
- Presentations: Feb Wednesdays 13-14 before reading week. Or some in class Tue, Wed

Visual Tracking

Readings:

Szeliski 8.1.3, 4.1

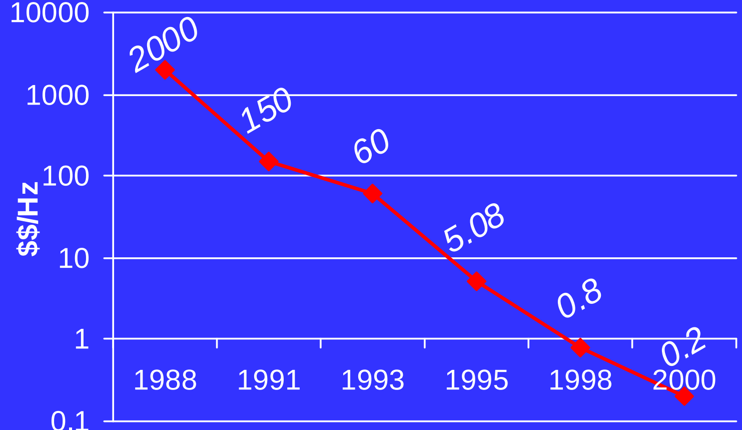
Paper: Baker&Matthews, IJCV

Lukas-Kanade 20 years on

Ma et al Ch 4.

Forsythe and Ponce Ch 17.

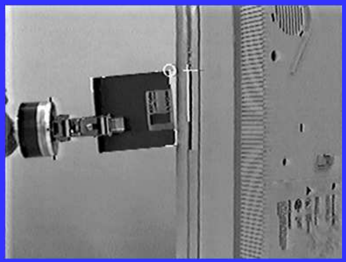
Why Visual Tracking?



Computers are fast enough!

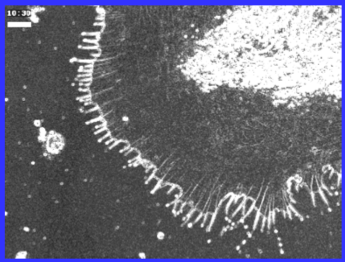
Related technology is cheap!

Applications abound



Motion Control

HCI



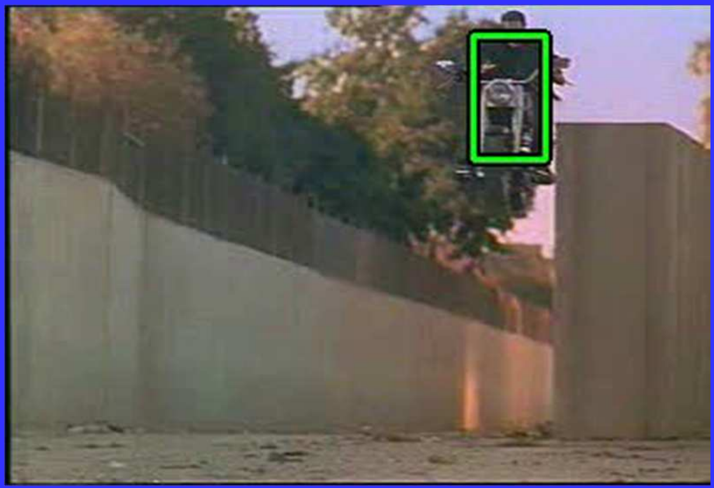
Measurement

Applications: Watching a moving target

- Camera + computer can determine how things move in an scene over time.

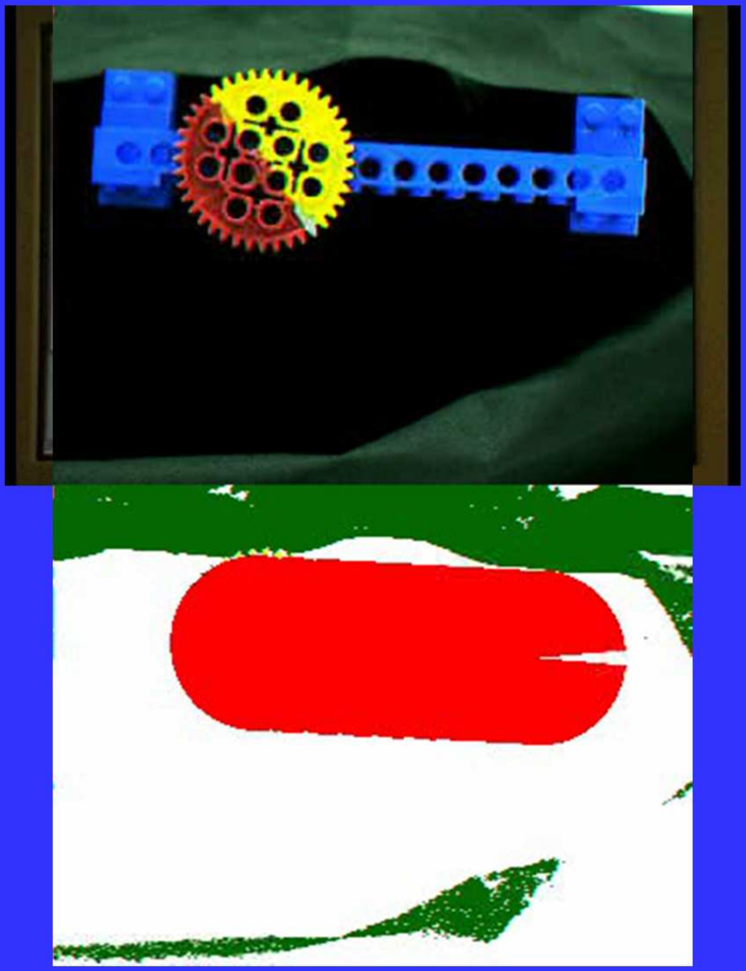
Uses:

- Security: e.g. monitoring people moving in a subway station or store
- Measurement: Speed, alert on colliding trajectories etc.



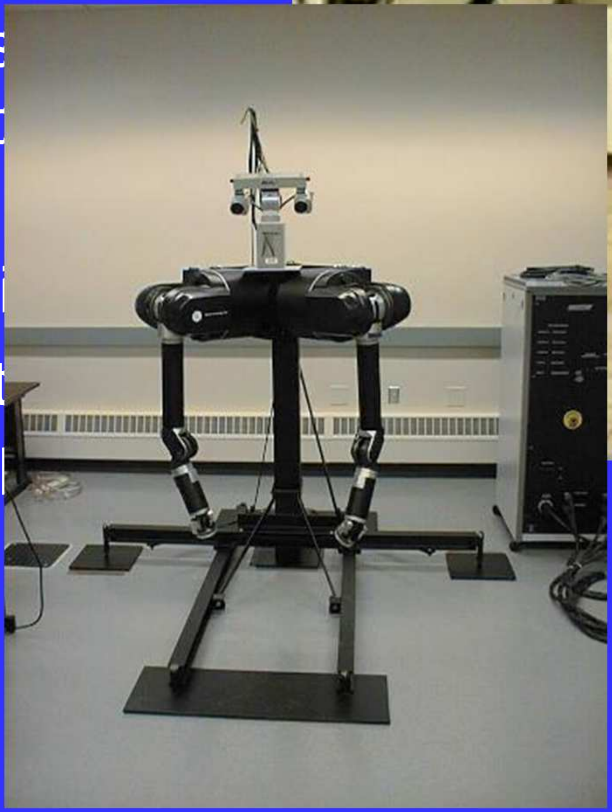
Applications
Human-Computer Interfaces

- Camera + computer tracks motions of human user and interprets this in an on-line interaction.
- Can interact with menus, buttons and e.g. drawing programs using hand movements as mouse movements, and gestures as clicking
- Furthermore, can interpret physical interactions



Applications
Human-Machine Interfaces

- Camera + computer tracks motions of human user, interprets this machine/robo out task.
- Remote mani
- Service robot the handicap elderly



Applications
Human-Human Interfaces

- Camera + computer tracks motions and expressions of human user, interprets, codes and transmits to render at remote location
- Ultra low bandwidth video communication
- Handheld video cell phone

A Modern Digital Camera (Firewire)

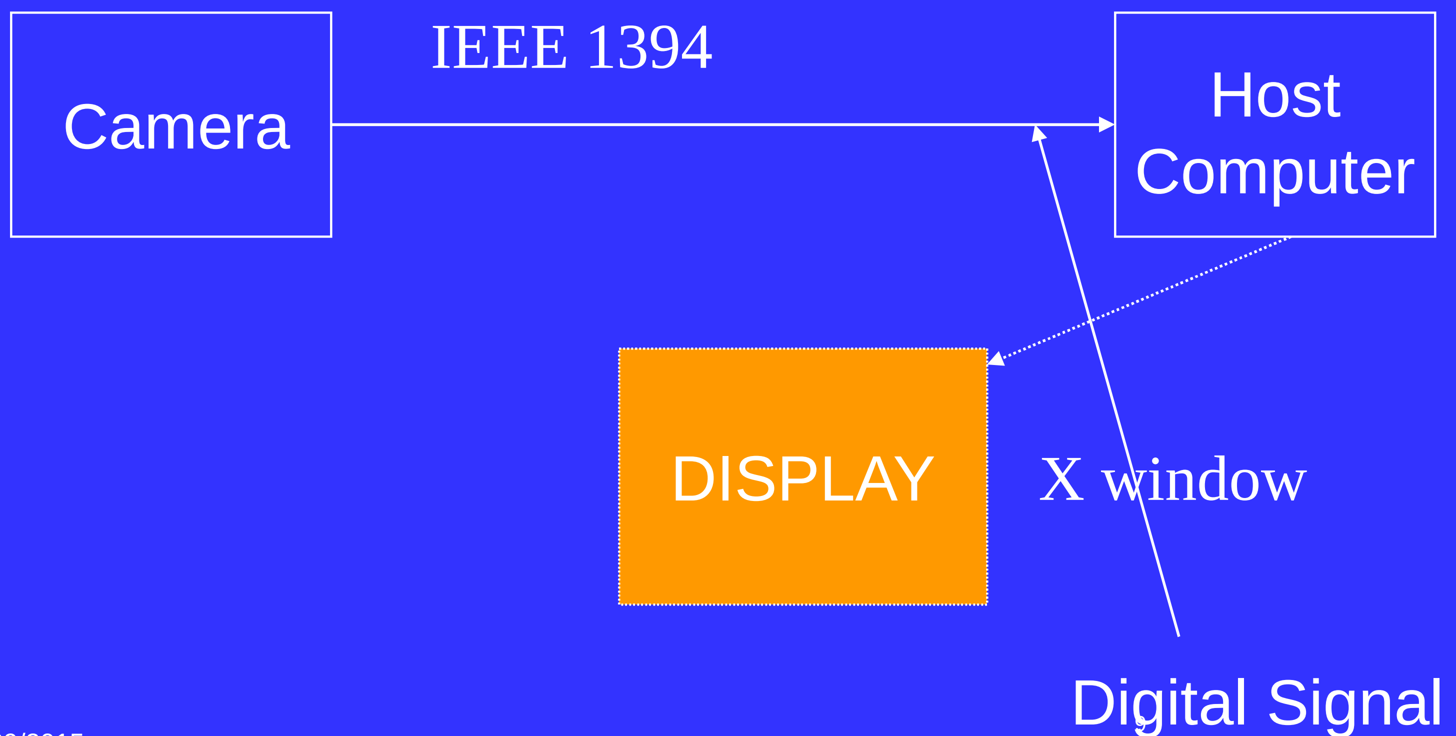
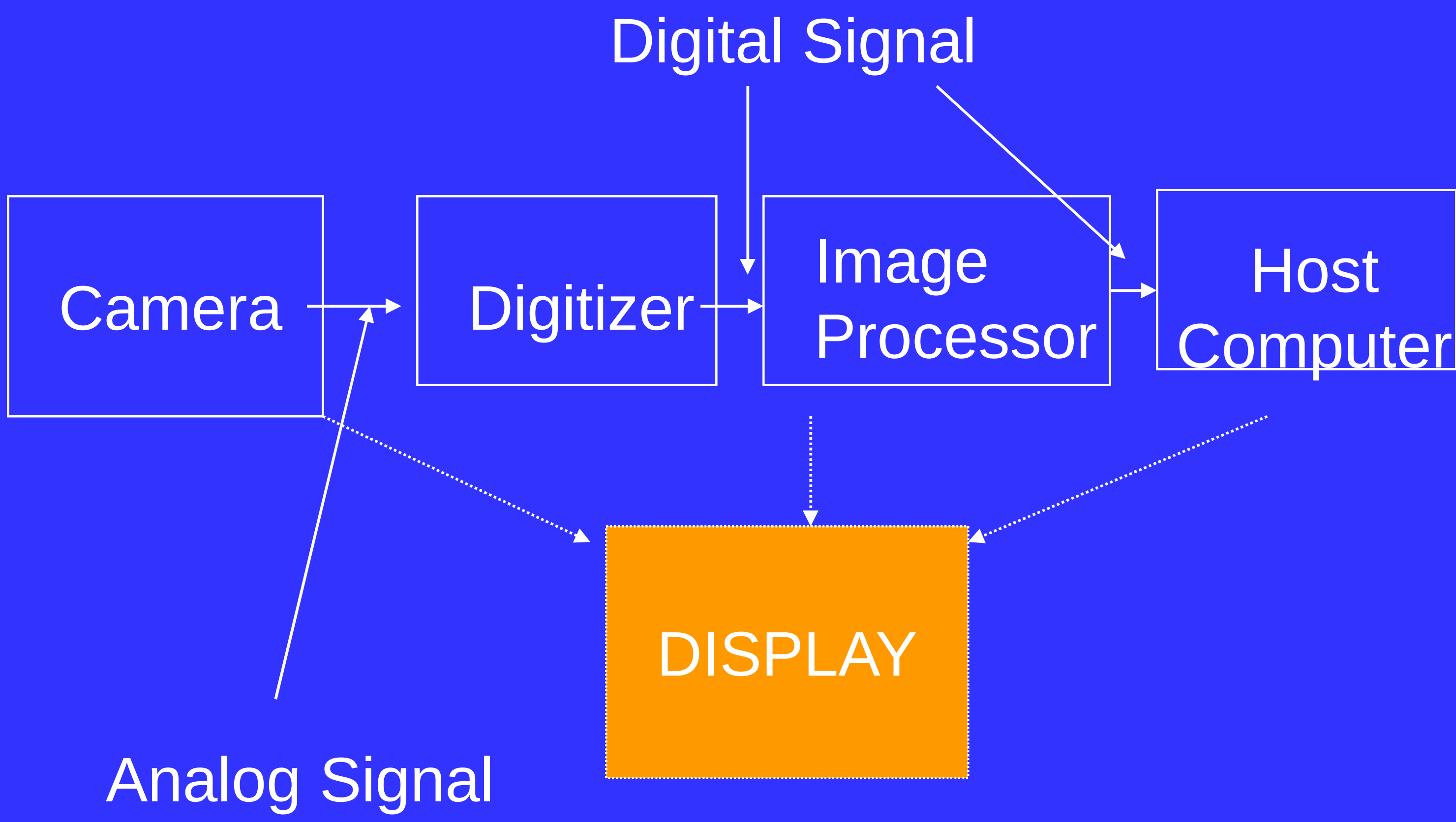


Image streams -> Smartphone, tablet



BANDWIDTH and PROCESSIN REQUIREMENTS: TV camera

Binary
1 bit * 640x480 * 30 = 9.2 Mbits/second

Grey
1 byte * 640x480 * 30 = 9.2 Mbytes/second

Color
3 bytes * 640x480 * 30 = 27.6 Mbytes/second (actually about 37 mbytes/sec)

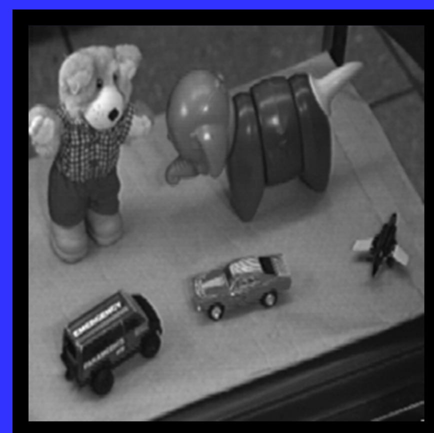
Typical operation: 8x8 convolution on grey image
64 multiplies + adds → 600 Mflops

Consider 800x600, 1200x1600…

Today's PC's are getting to the point they can process images at frame rate
For real time: process small window in image

Characteristics of Video Processing

abs(Image 1 - Image 2) = ?



Note: Almost all pixels change!

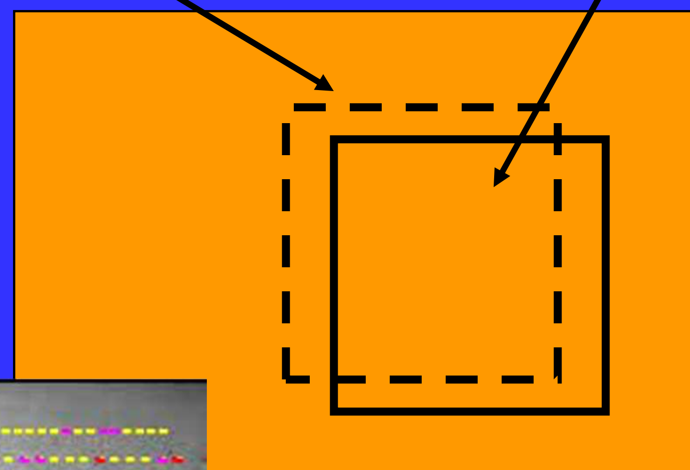
Constancy: The physical scene is the same

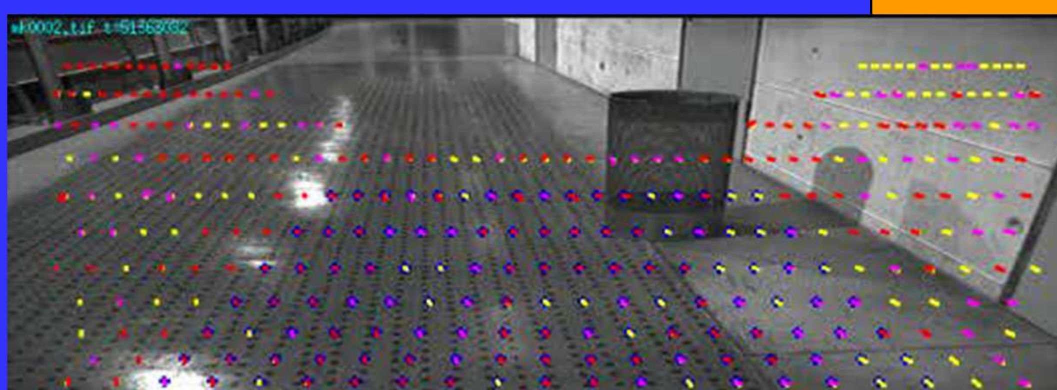
How do we make use of this?

Fundamental types of video processing
“Visual motion detecton”

- Relating two adjacent frames: (small differences):

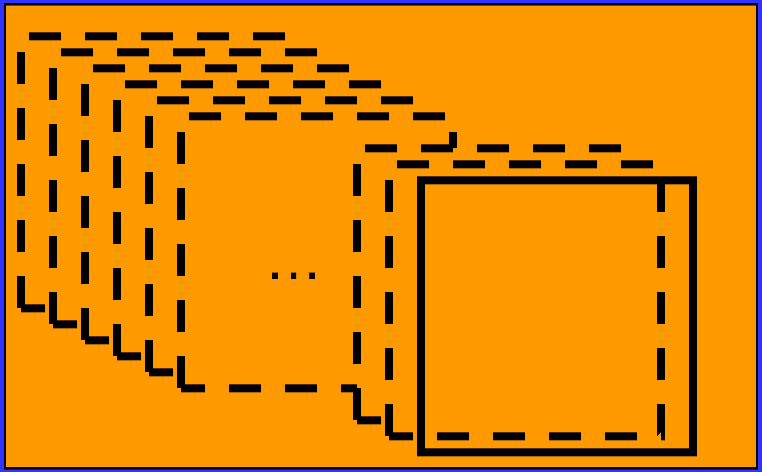
$$\mathrm{Im}(x+\delta x,\ \ y+\delta y,\ \ t+\delta t)\ \ =\ \ \mathrm{Im}(x,\ \ y,\ \ t)$$

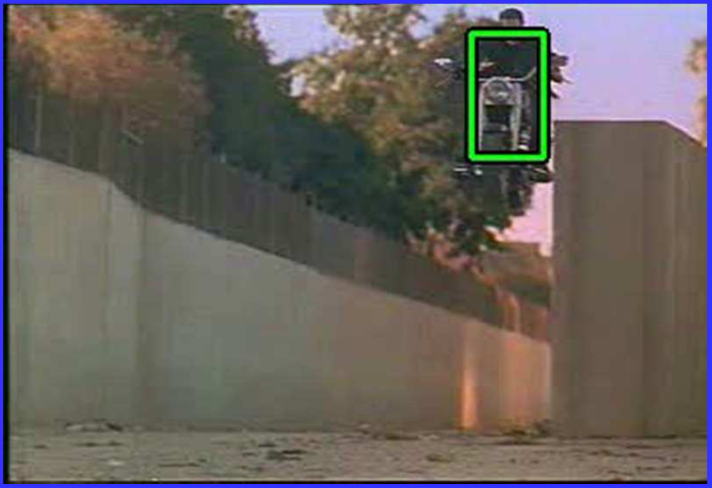




Fundamental types of video processing "Visual Tracking" / Stabilization

- Globally relating all frames: (large differences):





Types of tracking

- **Point tracking**

  Extract the point (pixel location) of a particular image feature in each image.

- **Segmentation based tracking**

  Define an identifying region property (e.g. color, texture statistics). Repeatedly extract this region in each image.

- **Stabilization based tracking**

  Formulate image geometry and appearance equations and use these acquire image transform parameters for each image.

Point tracking

- Simplest technique. Already commercialized in motion capture systems.
- Features commonly LED’s (visible or IR), special markers (reflective, patterns)
- Detection: e.g. pick brightest pixel(s), cross correlate image to find best match for known pattern.

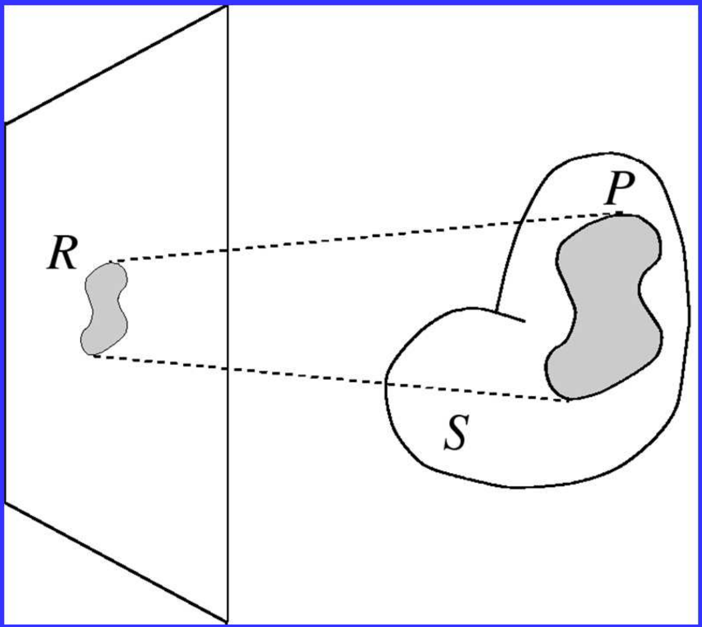
Region Tracking (Segmentation-Based)

- Select a "cue:" $\gamma(\mathbf{I}(x, y))$
  - foreground enhancement
  - background subtraction
- Segment
  - threshold
  - "clean up"
- Compute region geometry
  - centroid (first moment)
  - orientation (second moment)
  - scale (second moment)

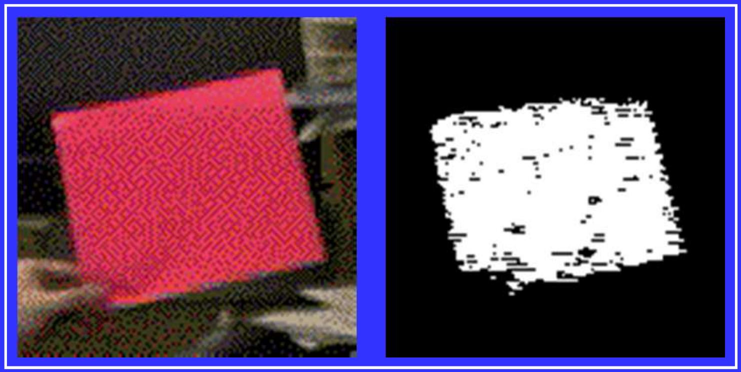
$$u = (x, y)'$$
$$m_i = \sum_i S(u) u^i$$
$$c = m_1 / m_0$$
$$\Lambda = m_2 / m_0 - m_1^2$$

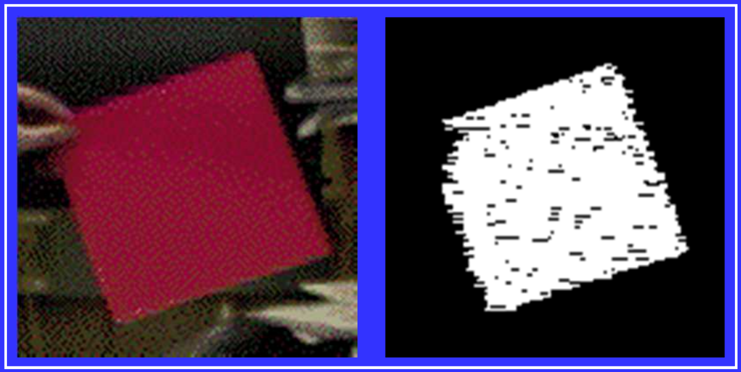
Regions

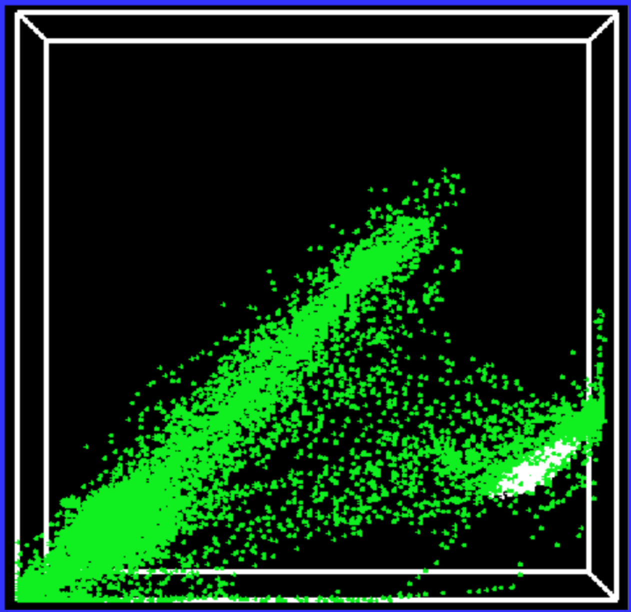
- $c_P = P \rightarrow \Re^3$ (color space): intrinsic coloration
  - Homogeneous region: $c_P(P)$ is roughly constant
  - Textured region: $c_P(P)$ has significant intensity gradients horizontally and vertically
  - Contour: (local) rapid change in contrast

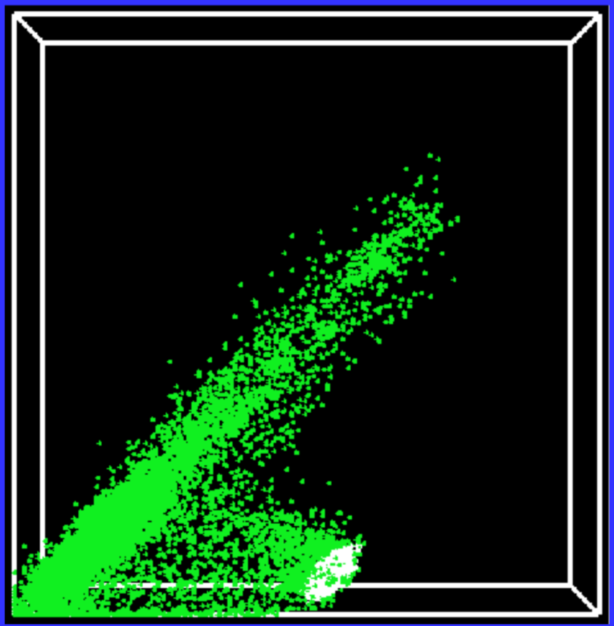


Homogeneous Color Region: Photometry









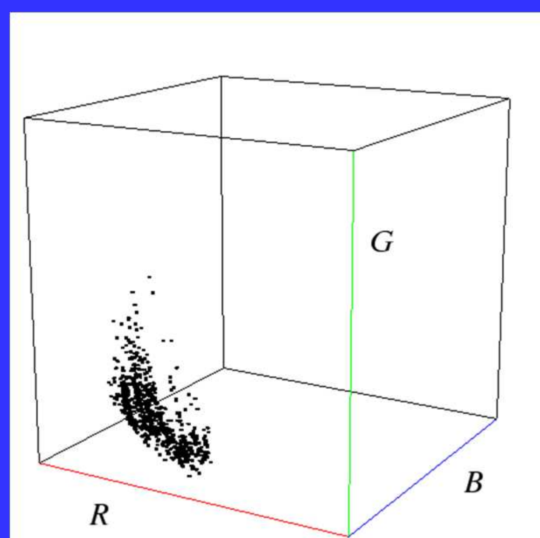
Color Representation

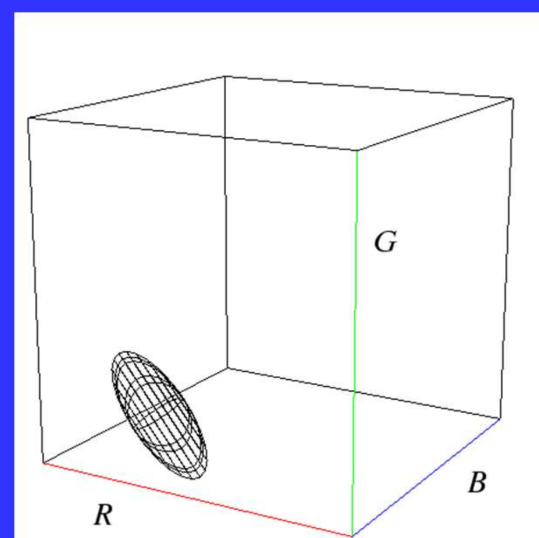
- $c_R = R \rightarrow \Re^3$ is irradiance of region *R*
- Color representation
  - DRM [Klinker et al., 1990]: if P is Lambertian, has *matte* line and *highlight* line
  - User selects *matte* pixels in *R*
  - PCA fits ellipsoid $(\mathbf{S}, \mathbf{R}^T, \mathbf{T})$ to matte cluster
  - Color similarity $\gamma(\mathbf{I}(x, y))$ is defined by Mahalanobis distance

$$|\mathbf{S}^{-1}\mathbf{R}^T(\mathbf{I}(x, y) - \mathbf{T})| < 1 inside$$

Homogeneous Region: Photometry

Sample





PCA-fitted ellipsoid

Color Extension (Contd.)

Color Histograms (Swain et al.)

- H_i = number of pixels in class i
- Histograms are vectors of bin counts
- Given histograms H and G, compare by

$$H \cdot G = \frac{\sum H_i G_i}{\sqrt{\sum H_i^2}\sqrt{\sum G_i^2}}$$

  - dense, stable signature
  - relies on segmentation
  - relative color and feature locations lost
  - affine transformations preserve area ratios of planar objects

Principles of Stabilization Tracking

p_t

I_0 $\qquad$ I_t

Variability model: $I_t = g(I_0, p_t)$

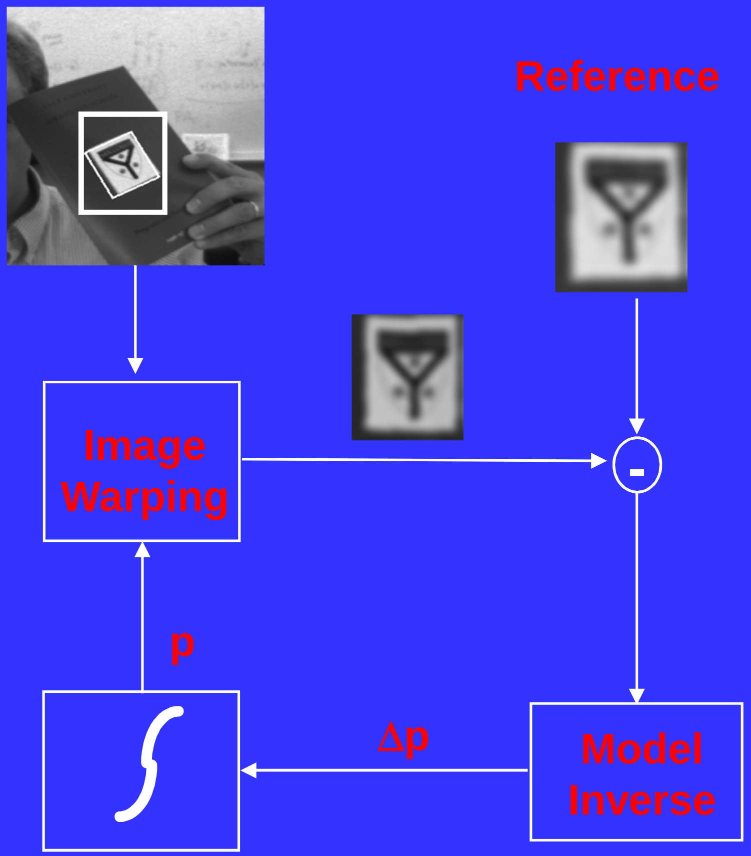
Incremental Estimation: From I_0, I_{t+1} and p_t compute Δp_{t+1}

$$\boxed{\| I_0 - g(I_{t+1}, p_{t+1}) \|^2 \implies \min}$$

Visual Tracking = Visual Stabilization

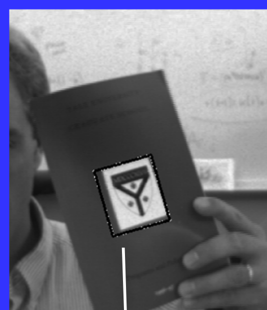
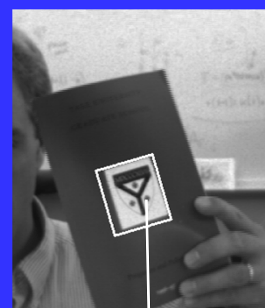
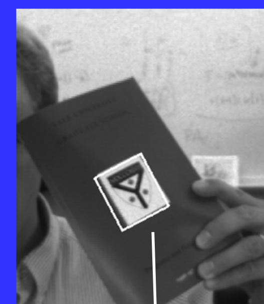
Tracking Cycle

- Prediction
  - Prior states predict new appearance
- Image warping
  - Generate a "normalized view"
- Model inverse
  - Compute error from nominal
- State integration
  - Apply correction to state



Stabilization tracking: Planar Case

Planar Object +
Linear camera => Affine motion model: $u'_i = A\, u_i + d$

Subtract these:  -  -  = nonsense

Warping

But these:  -  -  = small variation

$$I_t = g(p_t, I_0)$$

1/29/2015

State information

- Position
- Orientation
- Scale
- Aspect ratio/Shear

SO(2) + scale

Affine (SL(2) x R(2))

- Pose (S0(3); subgroup of Affine x R(2))

- Kinematic configuration (chains in SO(2) or SO(3))

- Non-rigid information (eigenmodes)

- Photometric information (parametric illumination models)

Mathematical Formulation

- Define a “warped image”
  - $f(p,x) = x'$ (warping function)
  - $I(x,t)$ (image a location x at time t)
  - $g(p,I_t) = (I(f(p,x_1),t), I(f(p,x_2),t), \dots I(f(p,x_N),t))'$
- Define the Jacobian of warping function
  - $M(p,t) = \left[\frac{\partial g}{\partial p}\right]$
- Consider “Incremental Least Squares” formulation
  - $O(\Delta p, t+\Delta t) = \| g(p_t, I_{t+\Delta t}) - g(0, I_0) \|^2$

Stabilization Formulation

- Model
  - $I_0 = g(p_t, I_t)$ (image I, variation model g, parameters p)
  - $\Delta I = \mathbf{M}(p_t, I_t)\,\Delta p$ (local linearization **M**)
- Define an error
  - $e_{t+1} = g(p_t, I_{t+1}) - I_0$
- Close the loop
  - $p_{t+1} = p_t - (\mathbf{M}^T \mathbf{M})^{-1} \mathbf{M}^T e_{t+1}$ where $\mathbf{M} = \mathbf{M}(p_t, I_t)$

M is N x m and
is time varying!

A Factoring Result

(Hager & Belhumeur 1998)

Suppose I = g(I_t, p) at pixel location u is defined as

$$I(u) = I(f(p,u),t)$$

and $\left(\frac{\partial f}{\partial u}\right)^{-1} \frac{\partial f}{\partial p}$ = **L**(u)**S**(p)

Then

$$\mathbf{M}(p, I_t) = \mathbf{M}_0\, S(p) \quad \text{where } \mathbf{M}_0 = \mathbf{M}(0, I_0)$$

Alternative: Compositional method:

"**Lucas-Kanade 20 Years On: A Unifying Framework**

By Simon Baker and Iain Matthews

International Journal of Computer Vision Volume 56, Number 3, p 221-255,

Stabilization Revisited

- In general, solve
  - $[\mathbf{S}^T\ \mathbf{G}\ \mathbf{S}]\ \Delta p = \mathbf{M}_0^T\ e_{t+1}$ where $\mathbf{G} = \mathbf{M}_0^T\mathbf{M}_0$ constant!
  - $p_{t+1} = p_t + \Delta p$
- If S is invertible, then
  - $p_{t+1} = p_t - \mathbf{S}^{-T}\ \mathbf{G}\ \ e_{t+1}$ where $\mathbf{G} = (\mathbf{M}_0^T\mathbf{M}_0)^{-1}\mathbf{M}_0^T$

G is m x N,
e is N x 1
S is m x m

→ O(mN) operations

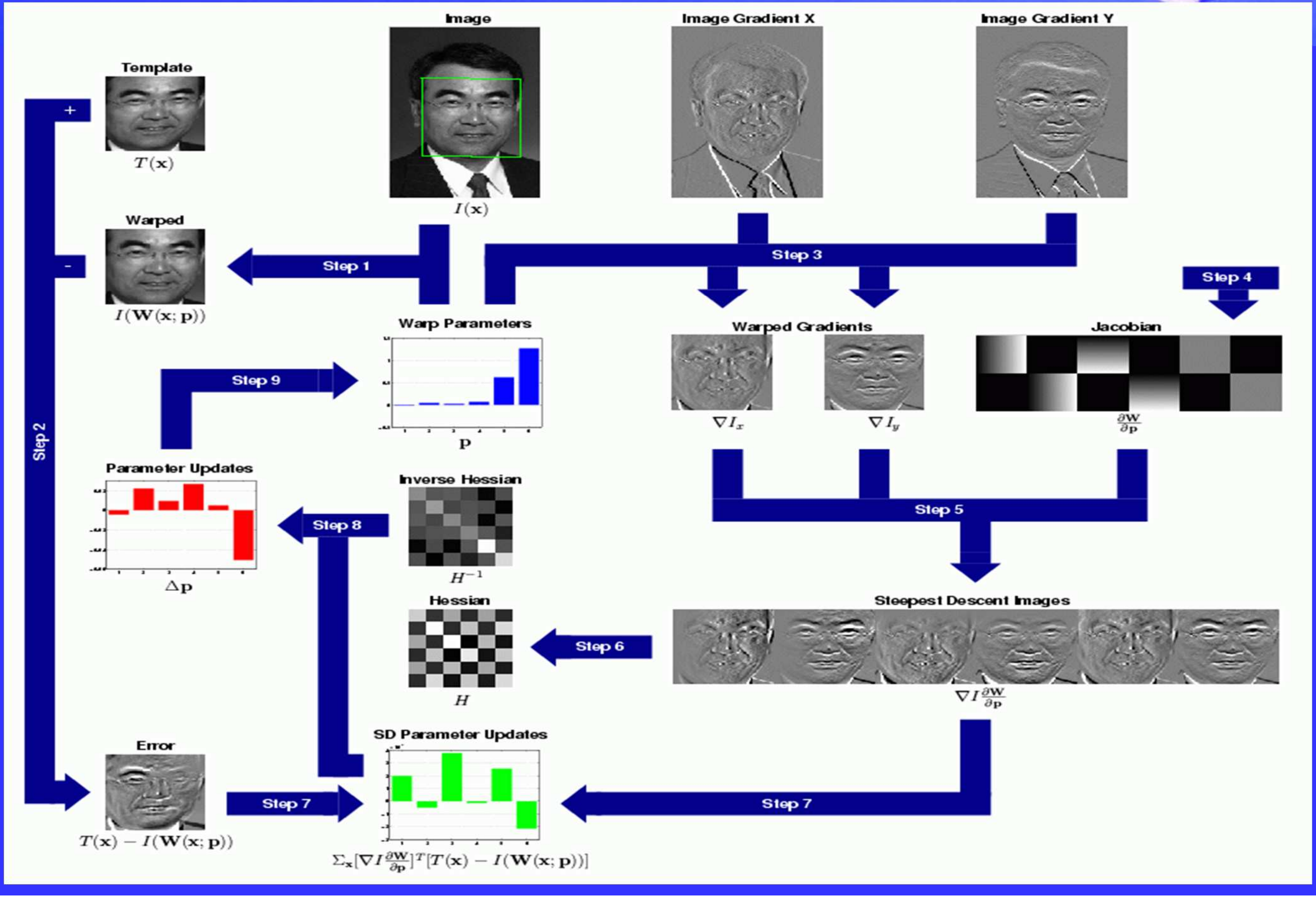
On The Structure of M

Planar Object -> Affine motion model: $u'_i = \mathbf{A}\,u_i + d$

X **Y** **Rotation** **Scale** **Aspect** **Shear**

$$M(p) = \partial g / \partial p$$

Putting all together



Video

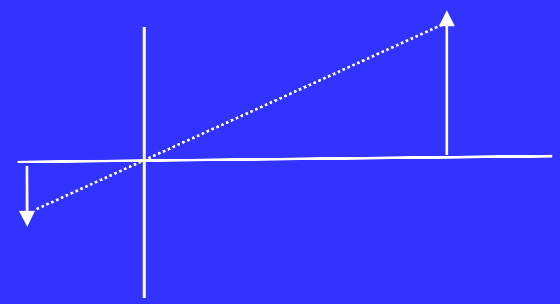
Tracking 3D Objects

What is the set of all images of a 3D object?

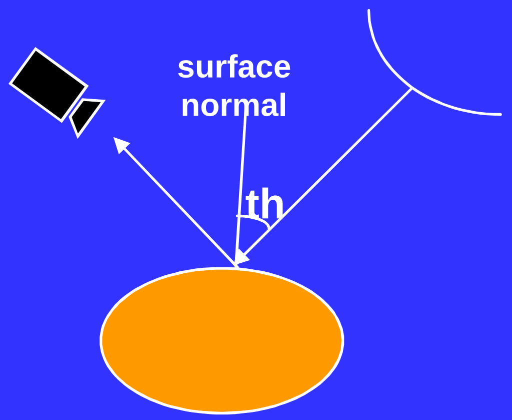
Motion **Illumination** **Occlusion**

Some Background

- Perspective (pinhole) camera
  - X' = x/z
  - Y' = y/z



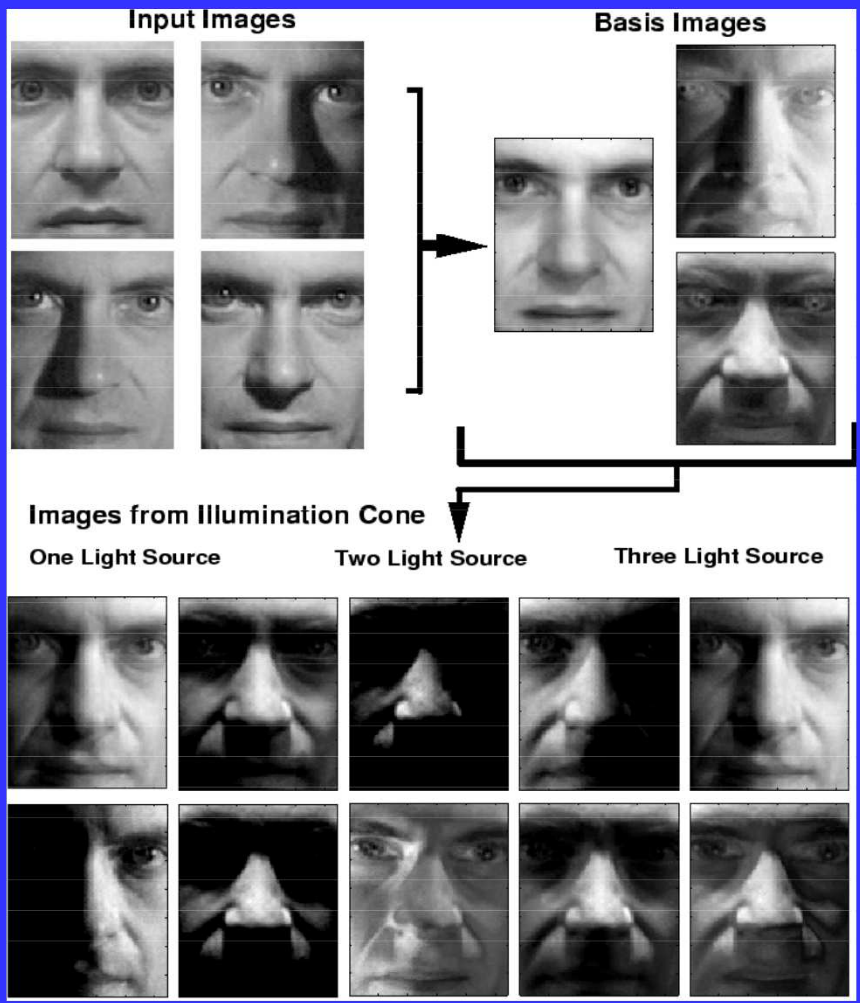
- Weak or para-perspective
  - X' = s x
  - Y' = s y
- Lambert's law
  - B = a cos(th)



3D Case: Local Geometry

Non-Planar Object: $u_i = \mathbf{A} u_i + b z_i + d$

3D Case: Illumination Modeling



Illumination basis:

$$I_t = \mathbf{B}\,\alpha + I_0$$

Observations:

- Lambertian object, single source, no cast shadows => 3D image space
- With shadows => a cone
- Empirical evidence suggests 5 to 6 basis images suffices

Putting It Together

- Variability model
  - $I_0 = g(p_t, I_t) + \mathbf{B}\,\alpha$ (variation model g, parameters p, basis **B**)
  - $\Delta I = \mathbf{M}(p)\,\Delta p + \mathbf{B}\,\alpha$ (local linearization **M**)
- Combine motion and illumination
  - $e_{t+1} = g(p_t, I_{t+1}) - I_0 - \mathbf{B}\,\alpha_t$
  - $p_{t+1} = p_t + (\mathbf{J}^T \mathbf{J})^{-1} \mathbf{J}^T\, e_{t+1}$ where $\mathbf{J} = [\, \mathbf{M}, \mathbf{B}\,]$
- Or, project into illumination kernel
  - $p_{t+1} = p_t + (\mathbf{M}^T \mathbf{N}\, \mathbf{M})^{-1} \mathbf{M}^T\, \mathbf{N}\, e_{t+1}$ where $\mathbf{N} = 1 - \mathbf{B}\,(\mathbf{B}^T\mathbf{B})^{-1}\, \mathbf{B}^T$

Handling Occlusion

- Robust error metric

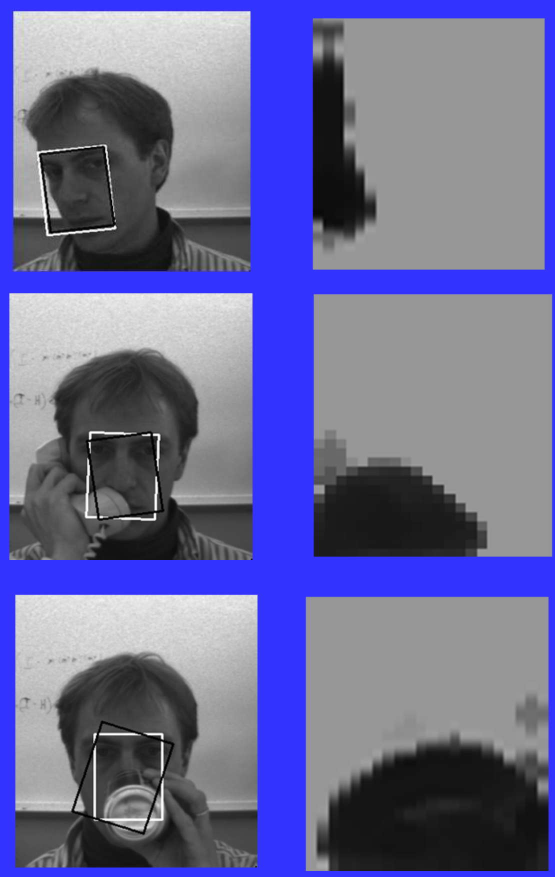
$$\Delta p = \arg\min \rho(e)$$

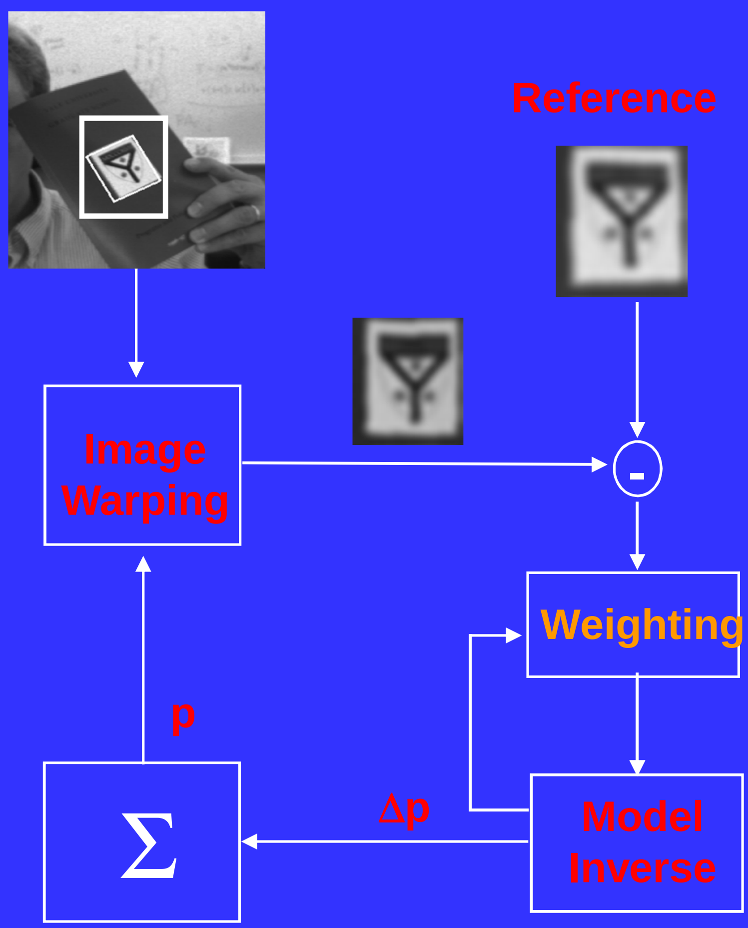
- Huber & Dutter 1981 --- modified IRLS estimation

$$\Delta p^k = (\mathbf{M}^T\mathbf{M})^{-1}\,\mathbf{M}^T\;\mathbf{W}(\Delta p^k)\;e$$

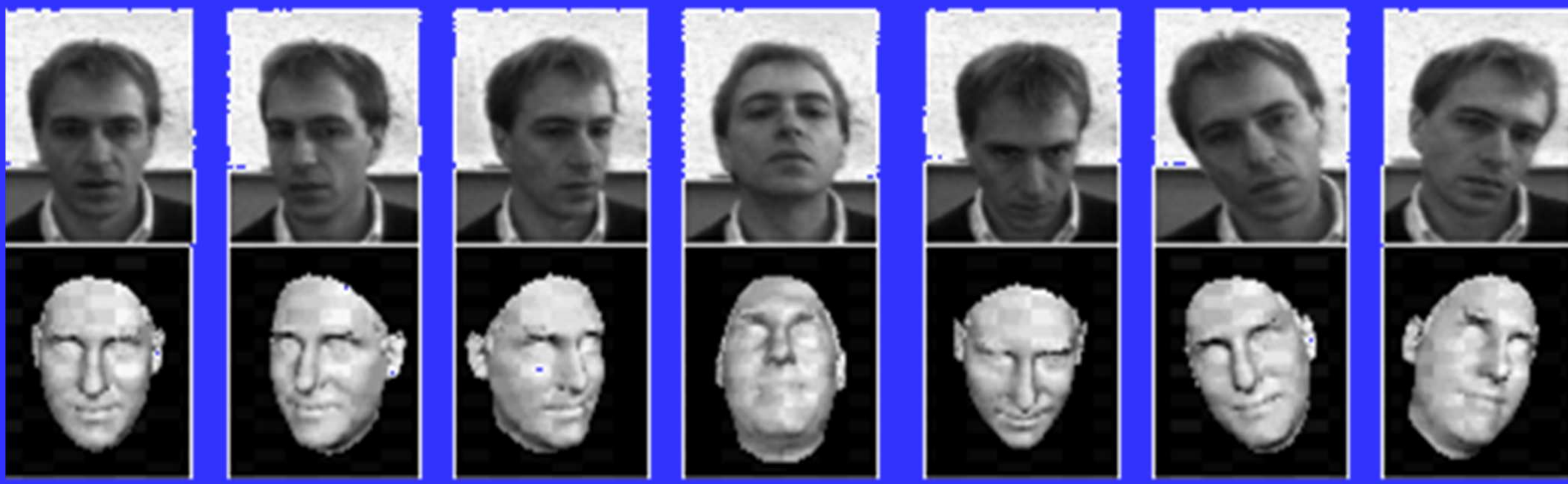
- Temporal propagation of weighting
  - $\mathbf{W}_{t+1} = F\,(\mathbf{W}_t)$

Handling Occlusion

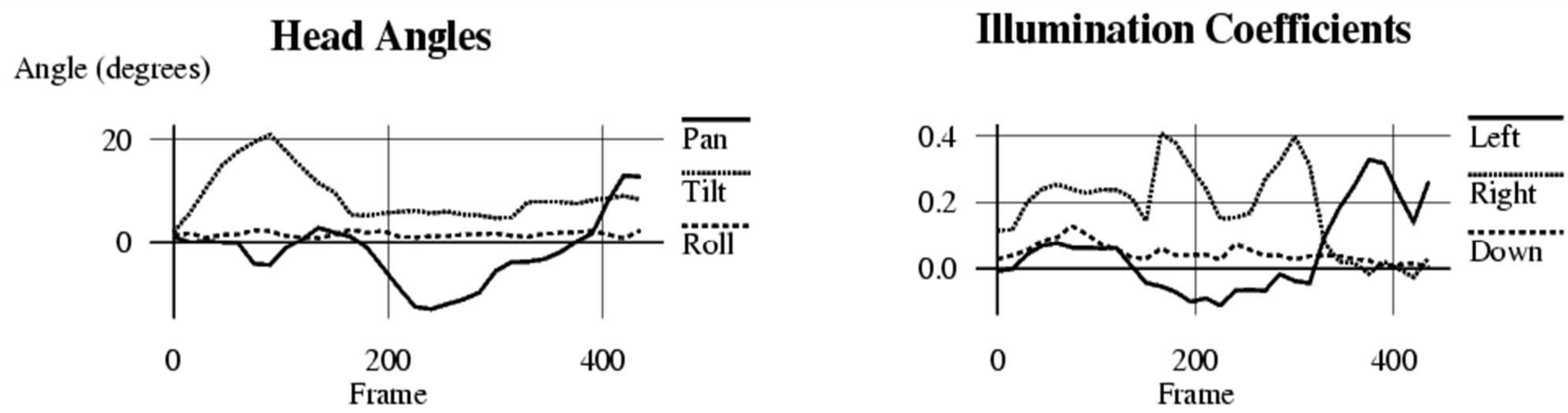




Pose



Pose and Illumination

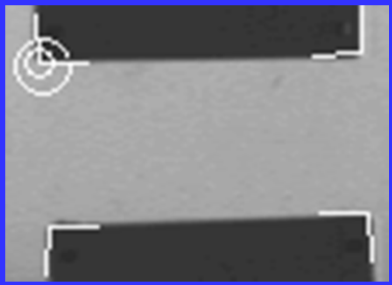


Related Work: Modalities

- Color
  - Histogram [Birchfield, 1998; Bradski, 1998]
  - Volume [Wren *et al*., 1995; Bregler, 1997; Darrell, 1998]
- Shape
  - Deformable curve [Kass *et al*.1988]
  - Template [Blake *et al*., 1993; Birchfield, 1998]
  - Example-based [Cootes *et al*., 1993; Baumberg & Hogg, 1994]
- Appearance
  - Correlation [Lucas & Kanade, 1981; Shi & Tomasi, 1994]
  - Photometric variation [Hager & Belhumeur, 1998]
  - Outliers [Black *et al*., 1998; Hager & Belhumeur, 1998]
  - Nonrigidity [Black *et al*., 1998; Sclaroff & Isidoro, 1998]

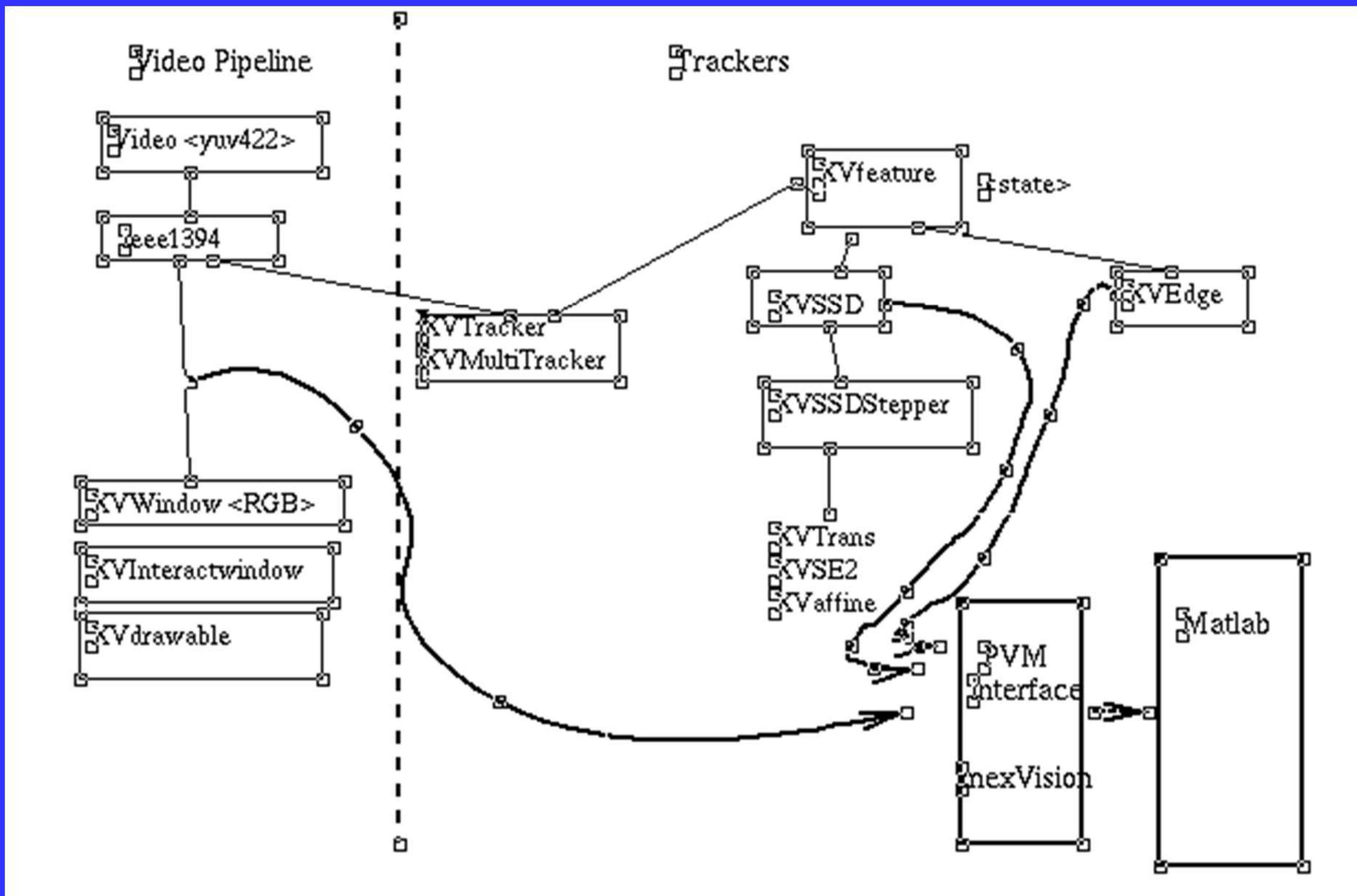
XVision: Desktop Feature Tracking

- Graphics-like system
  - Primitive features
  - Geometric constraints



- Fast local image processing
  - Perturbation-based algorithms
- Easily reconfigurable
  - Set of C++ classes
  - State-based conceptual model of information propagation
- Goal
  - Flexible, fast, easy-to-use substrate

Xvision/ mexVision structure



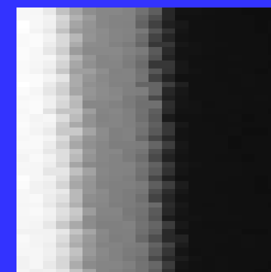
Edges: An Illustrative Example

- Classical approach to feature-based vision
  - Feature extraction (e.g. Canny edge detector)
  - Feature grouping (edges →edgels)
  - Feature characterization (length, orientation, magnitude)
  - Feature matching (combinatorial problem!)

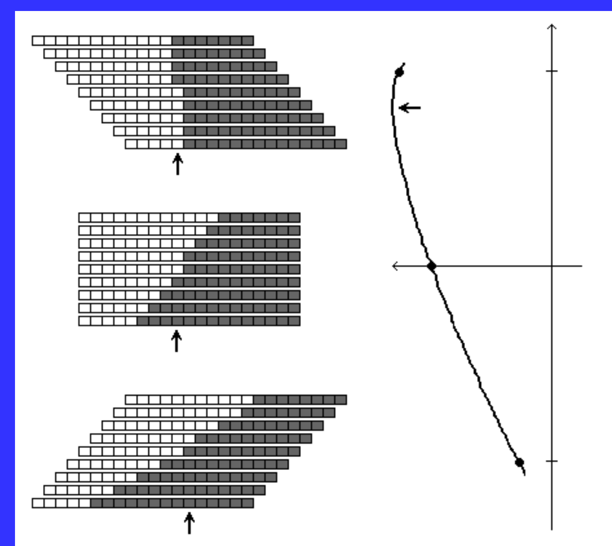
- XVision approach
  - Develop a “canonical” edge with fixed state
    - Vertical step edge with position, orientation, strength
  - Assume prior information and use warping to acquire candidate region
  - Optimize detection algorithm to compute from/to state

Edges (XVision Approach)

Rotational warp



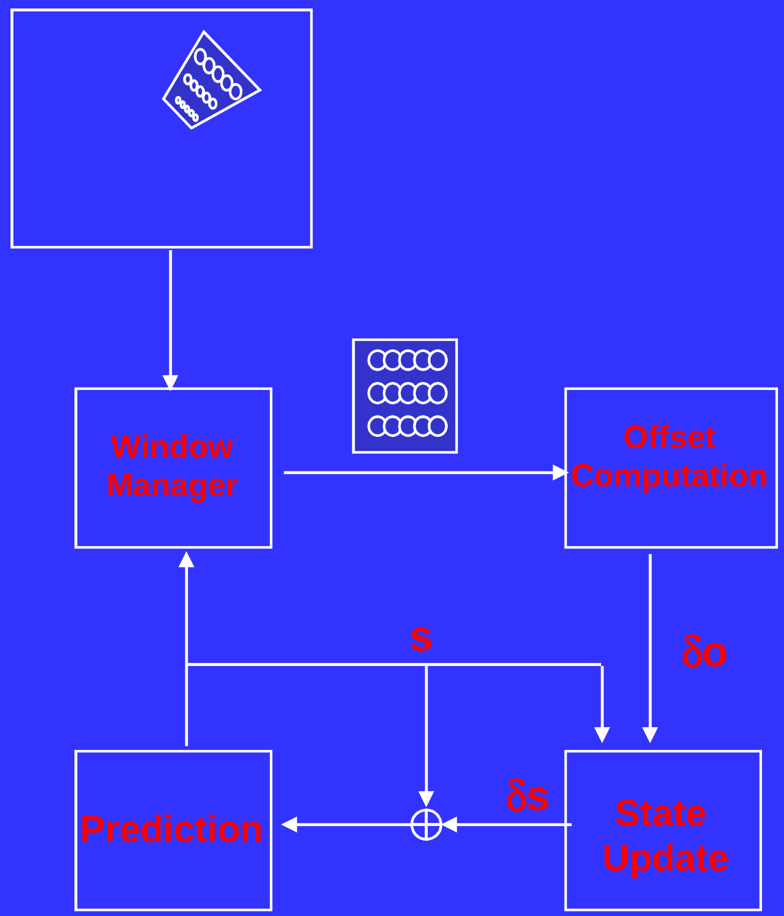
Apply a derivative of a triangle (IR filter) across rows



Sum and interpolate to get position and orientation

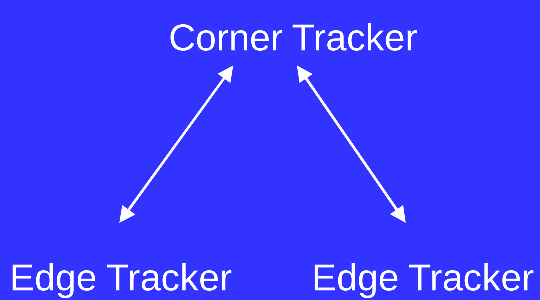
Abstraction: Feature Tracking Cycle

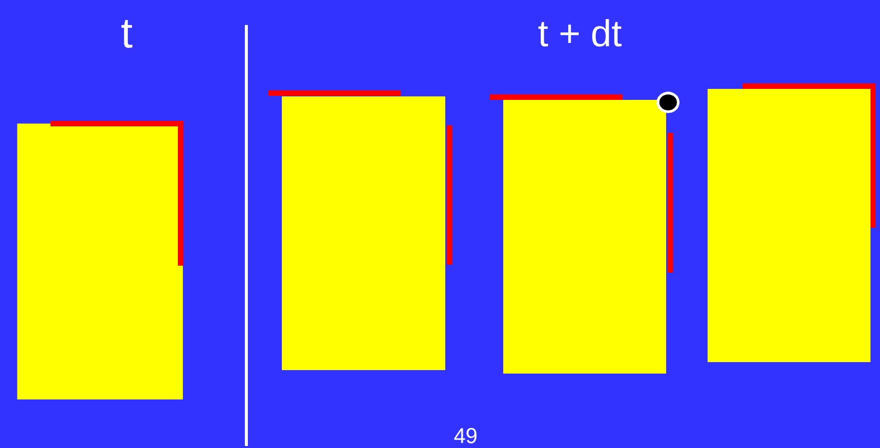
- Prediction
  - prior states predict new appearance
- Image rectification
  - generate a “normalized view”
- Offset computation
  - compute error from nominal
- State update
  - apply correction to fundamental state



Abstraction: Feature Composition

- Features related through a projection-embedding pair
  - f: $R^n \rightarrow R^m$, and g: $R^m \rightarrow R^n$, with $m \le n$ s.t. $f \circ g$ = identity
- Example: corner composed of two edges
  - each edge provides one positional parameter and one orientation.
  - two edges define a corner with position and 2 orientations.





Example Tools

Primitives

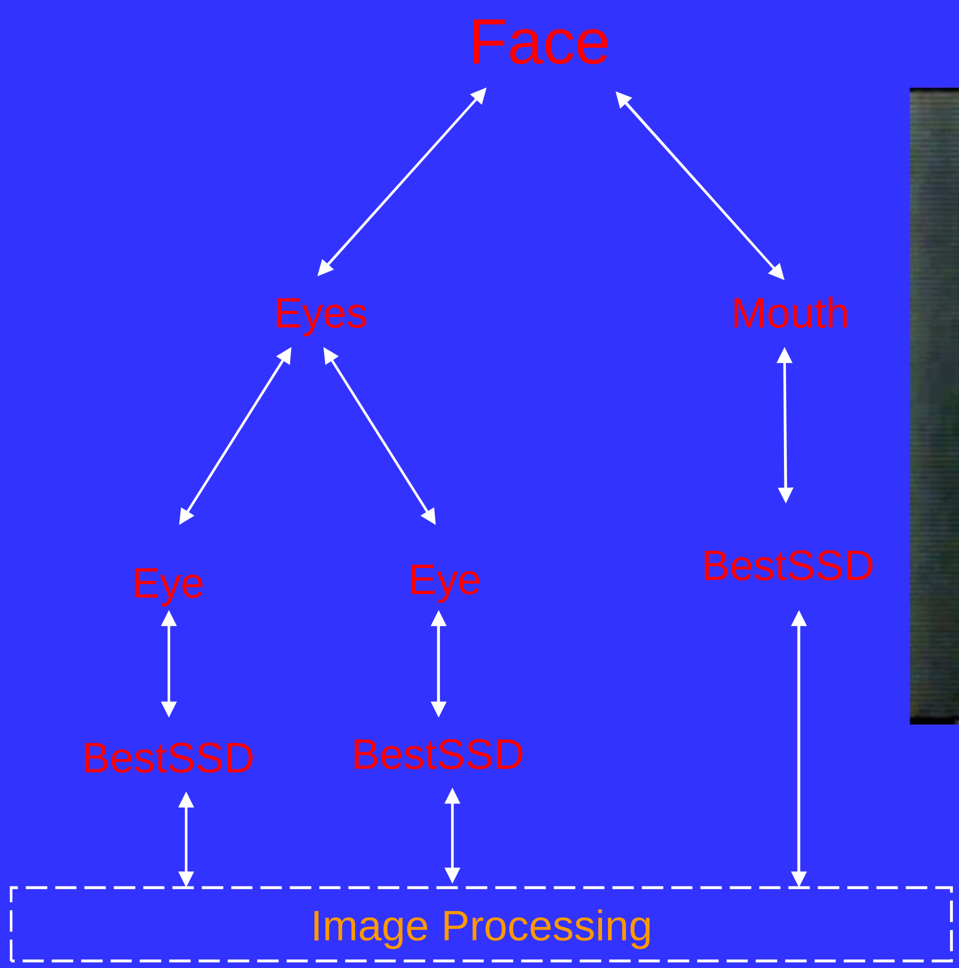
- Blobs
  - position/orientation
- Lines
  - position/orientation
- Correlation
  - position/orientation
  - +scale
  - full affine

Composed

- Intersecting Lines
  - corners
  - tee
  - cross
- Objects
  - diskette
  - screwdriver
- Snakes

Over 800 downloads since 1995

Regions + XVision Composition

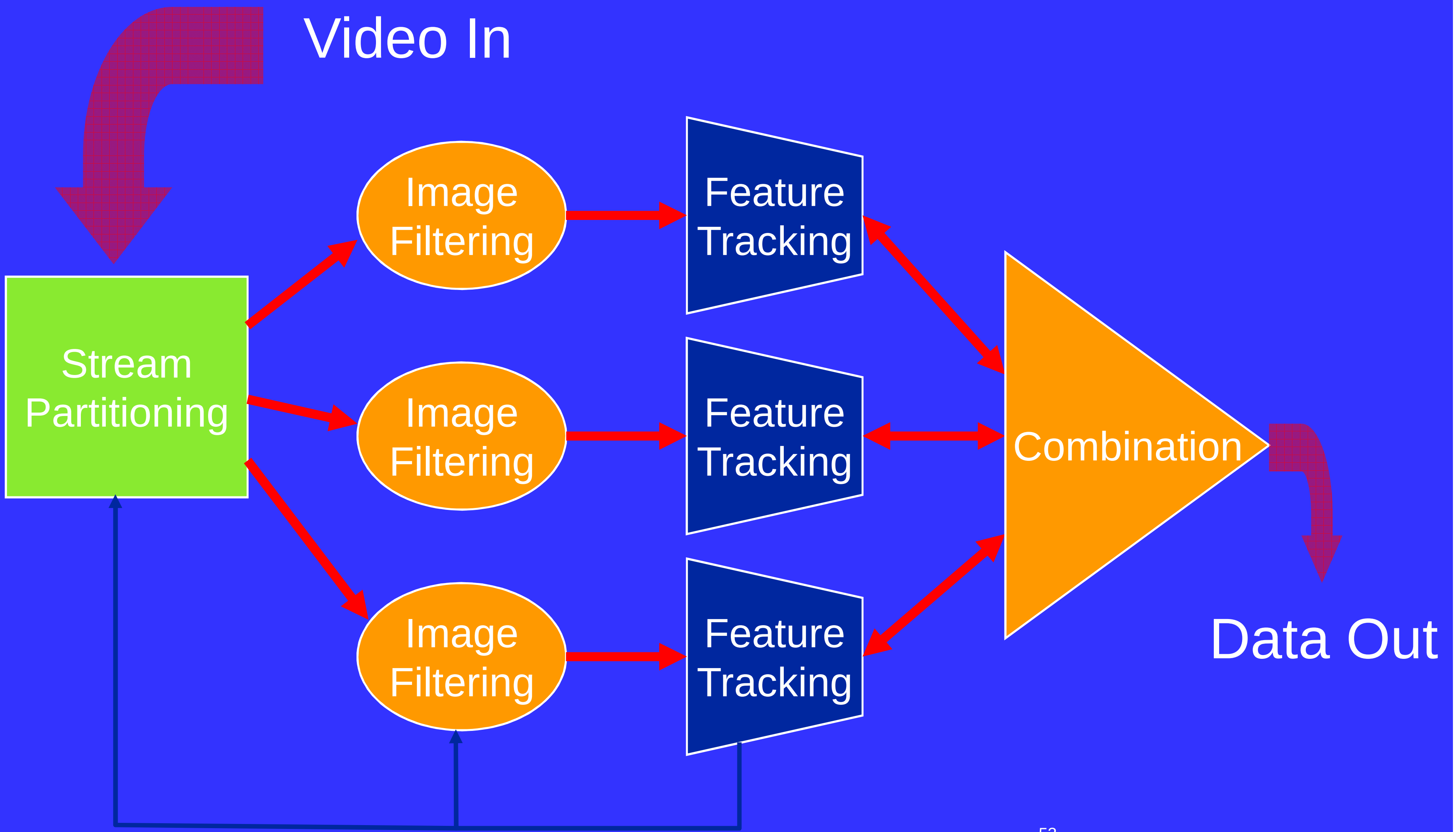


Software: Limitations

- Integration leads to recurring implementation chores
  - Writing loops to step forward discretely in time
  - Time slicing time-varying components that operate in parallel
- Code reuse
  - Two pieces of code need to do *almost* the same thing, but not quite
- What's correct?
  - The design doesn't look at all like the code
  - Hard to tell if its a bug in the code, or a bug in the design

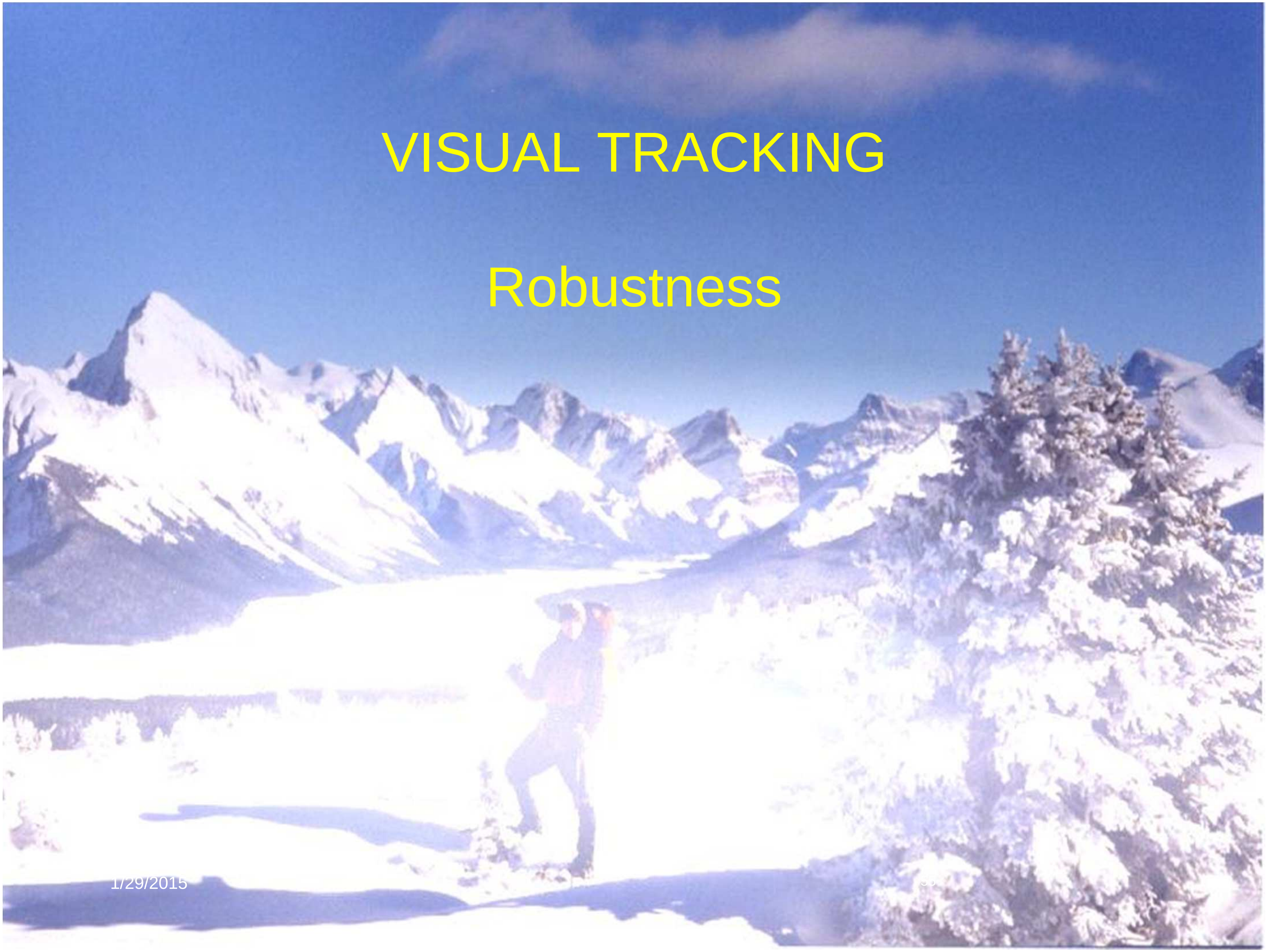
Programs should describe *what* to do not *how* to do it

New XVision Programming Model



Programming Dynamical Systems

```
trackMouth v = bestSSD mouthIms (newsrcI v (sizeof mouthIms))
trackLEye v  = bestSSD leyeIms  (newsrcI v (sizeof leyeIms))
trackREye v  = bestSSD reyeIms  (newsrcI v (sizeof reyeIms))
trackEyes v = composite2 (split, join) (trackLEye v) (trackREye v)
   where
      split = segToOrientedPts    --- some geometry
      join  = orientedPtsToSeg    --- some more geometry
trackClown v = composite2 concat2 (trackEyes v) (trackMouth v)
```

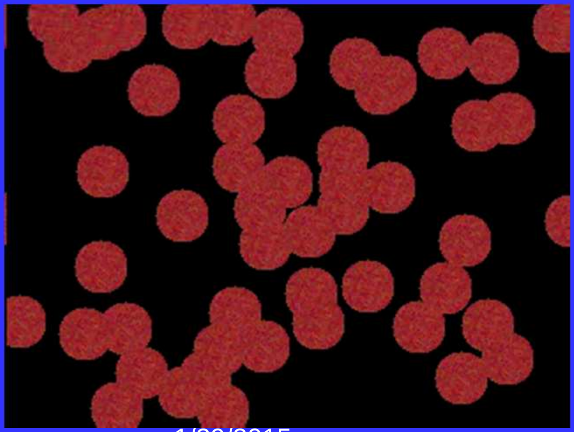

VISUAL TRACKING
Robustness
1/29/2015

Visual Disruptions

Distraction
Scene element similar in image appearance to target

Occlusion
Scene element interposed between camera and target

Agile motion
Target movement that exceeds prediction abilities of tracker



Related Work: Single Object Tracking

- Sampling: Condensation [Isard & Blake, 1996]
- Resolve after overlap [Rosales & Sclaroff, 1999; Stauffer & Grimson, 1999]
- Analyze overlap [Koller *et al*., 1994; Rehg & Kanade, 1995; Beymer & Konolige, 1999; MacCormick & Blake, 1999]

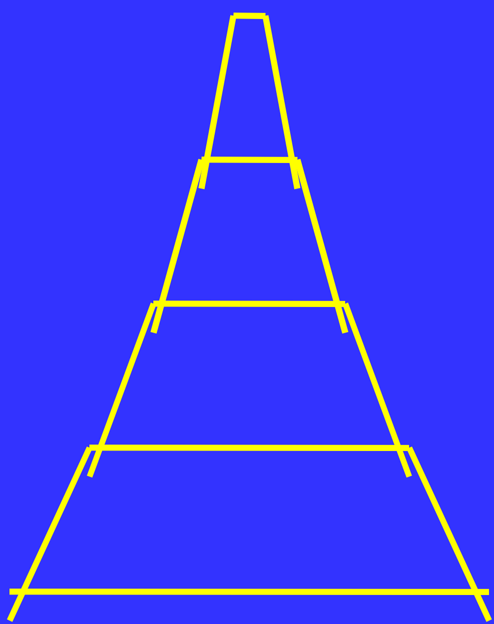
Related Work: Joint Tracking

- Resolve after overlap [Rosales & Sclaroff, 1999; Stauffer & Grimson, 1999]
- Analyze overlap [Koller *et al*., 1994; Rehg & Kanade, 1995; Beymer & Konolige, 1999; MacCormick & Blake, 1999]
- Multi-part tracking
  - 3-D [Rehg & Kanade, 1995; Gavrila & Davis, 1996; Bregler & Malik, 1997]
  - 2.5-D [Wren *et al*., 1995; Jojic *et al*., 1999]
  - 2-D [Reynard *et al*., 1996; Ju *et al*., 1996; Morris & Rehg, 1998]
- Multi-attribute tracking
  - Sequential [Kahn *et al*., 1996; Toyama, 1997]
  - Simultaneous [Darrell *et al*., 1998; Birchfield, 1998]

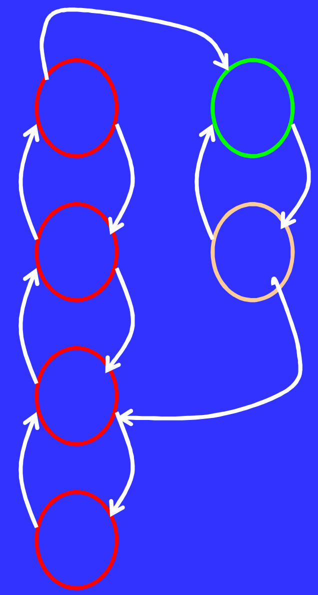
IFA: Architecture

(Kentaro Toyama, Microsoft)

target state



full configuration space



search track

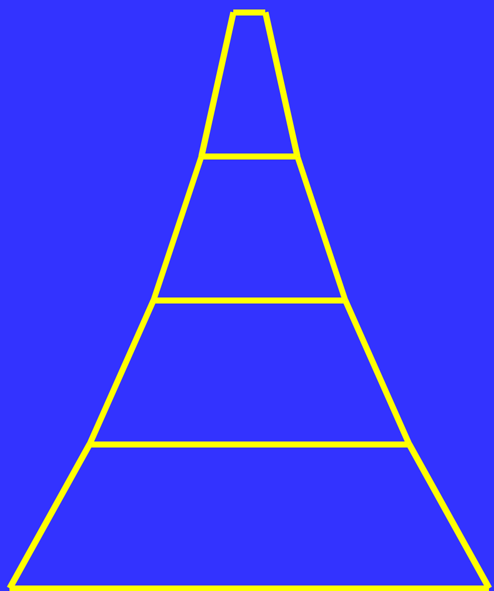
algorithmic layers **internal state**

Basic idea: layer complementary modalities into a hybrid systems architecture

IFA: Layers

IFA is based on a search in the *CONFIGURATION SPACE of the target*

target state



full configuration space

algorithmic layers

- Layered tracking and searching algorithms
- Sorted by precision
- Execution one layer at a time

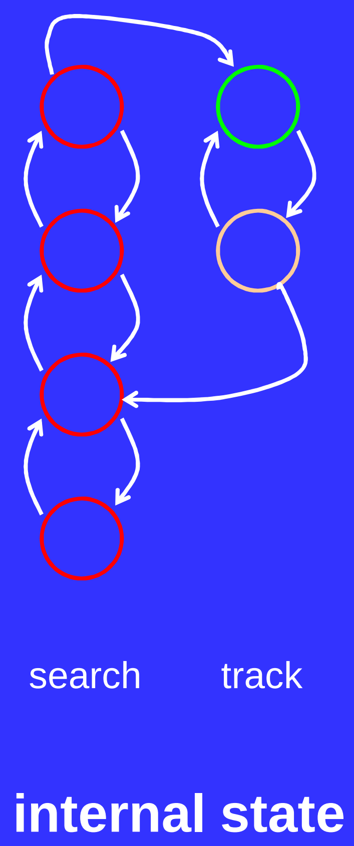
IFA: Layers

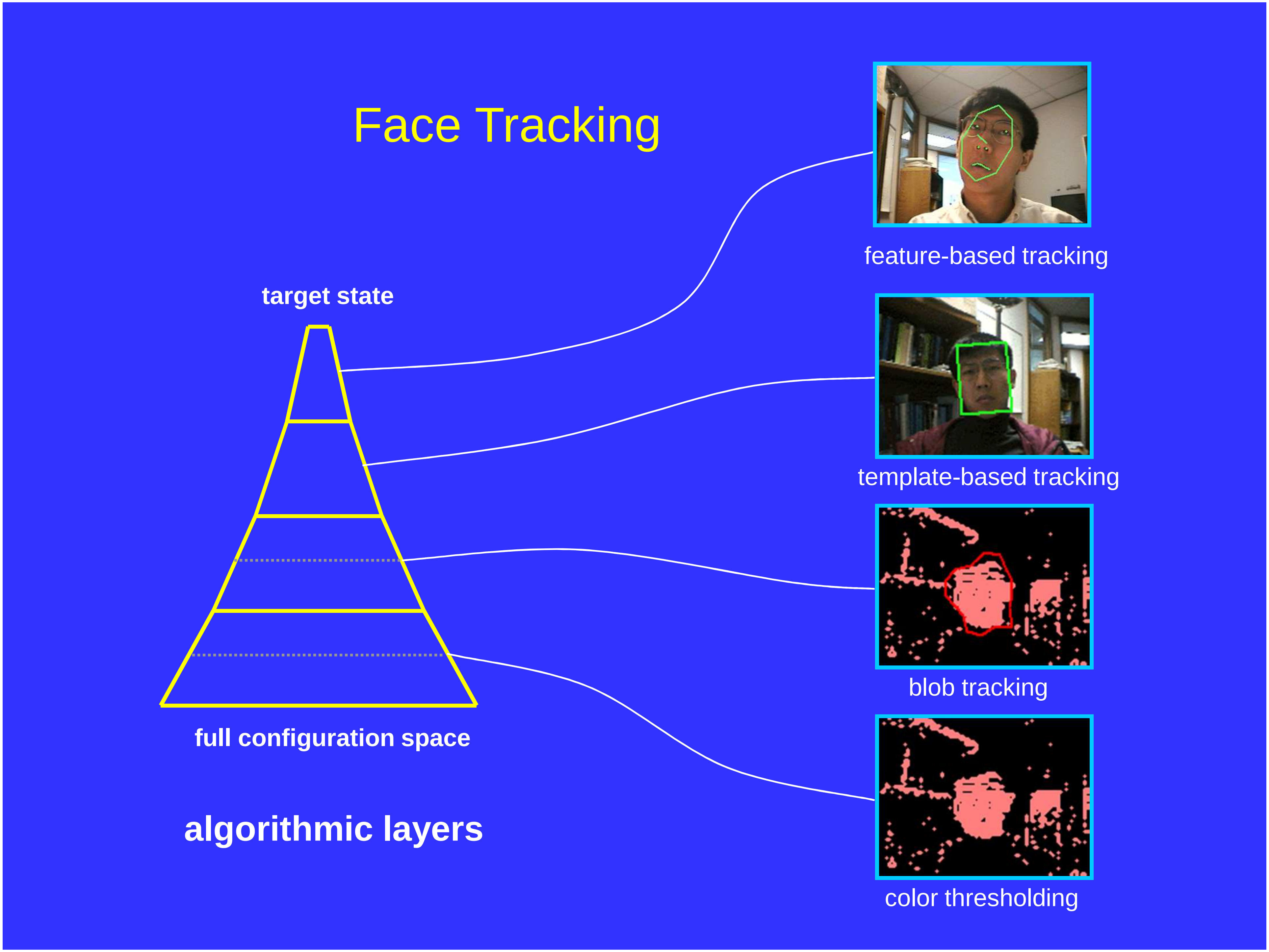
- Selectors
  - $P((x^* \text{ in } X^{out}) \mid x^* \text{ in } X^{in})$ high, but not 1
  - Failure after repeated failure in higher layers
  - Partition search space
  - Heuristic focus of attention

- Trackers
  - $P((x^* \text{ in } X^{out}) \text{ or } (X^{out} = 0) \mid x^* \text{ in } X^{in}) = 1$
  - Failure when $X^{out} = 0$, or when trackability low
  - Provide partial configuration information
  - Confirm existence of object in search space

IFA: State Transitions

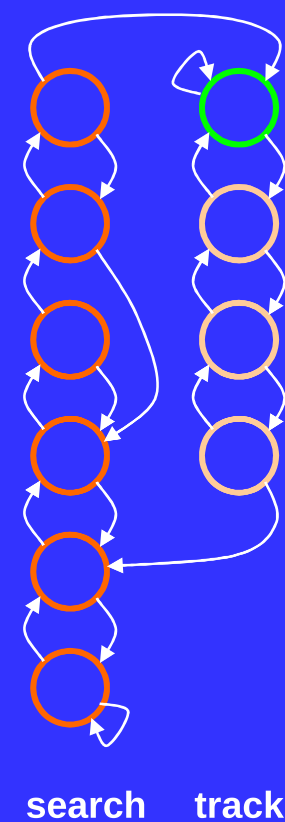
- Layer successes determine transitions
- Search and track modes
- Symbolic representation of success and precision




Face Tracking
target state
full configuration space
algorithmic layers
feature-based tracking
template-based tracking
blob tracking
color thresholding

Face Tracking

Layer 6	features & 3D geometry	(precise x y z r_x r_y r_z)
Layer 5	template	(precise x y r_z)
Layer 4	blob w/orientation	(approximate x y r_z)
Layer 3	blob	(approximate x y)
Layer 2	color and motion	(candidate x y)
Layer 1	color	(candidate x y)



Face Tracking: Color and Motion

- Region of interests found.
- Liberal color model for skin colors

$$k_1 < R/G < k_2$$
$$k_3 < R/B < k_4$$
$$k_5 < (R+G+B)/3 < k_6$$

- Threshold image differences

color classification

Face Tracking: Color Blob

- Approximate 2D position, in-plane orientation tracked.

- “Radial spanning” **(Toyama 98)**

  - **k spokes push outward with forces:**
  - $F_i \quad = \quad F_i^{out} + F_i^{in} + F_i^{int}$
  - $F^{out} \quad : \quad$ **k^{out} P(pixel is skin color)**
  - $F^{in} \quad : \quad$ **k^{in} P(pixel is not)**
  - $F_i^{int} \quad : \quad$ **determined by neighbors**

blob tracking

Face Tracking: Template

- 2D pose tracked (position and in-plane orientation).
- Linearized SSD **(Hager & Belhumeur 96)**

$$dx = -(M^T M)^{-1} M^T [I(x, t+\tau) - I(0, t_0)]$$

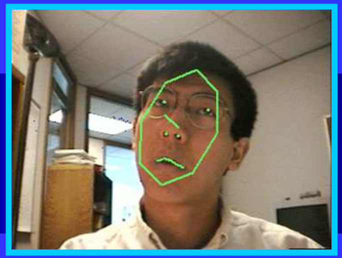
I : **image as vector**

x : **warp parameters, [x y θ]**

M : **Jacobian of** I **w.r.t.** x

template-based tracking

Face Tracking: Features



feature-based tracking

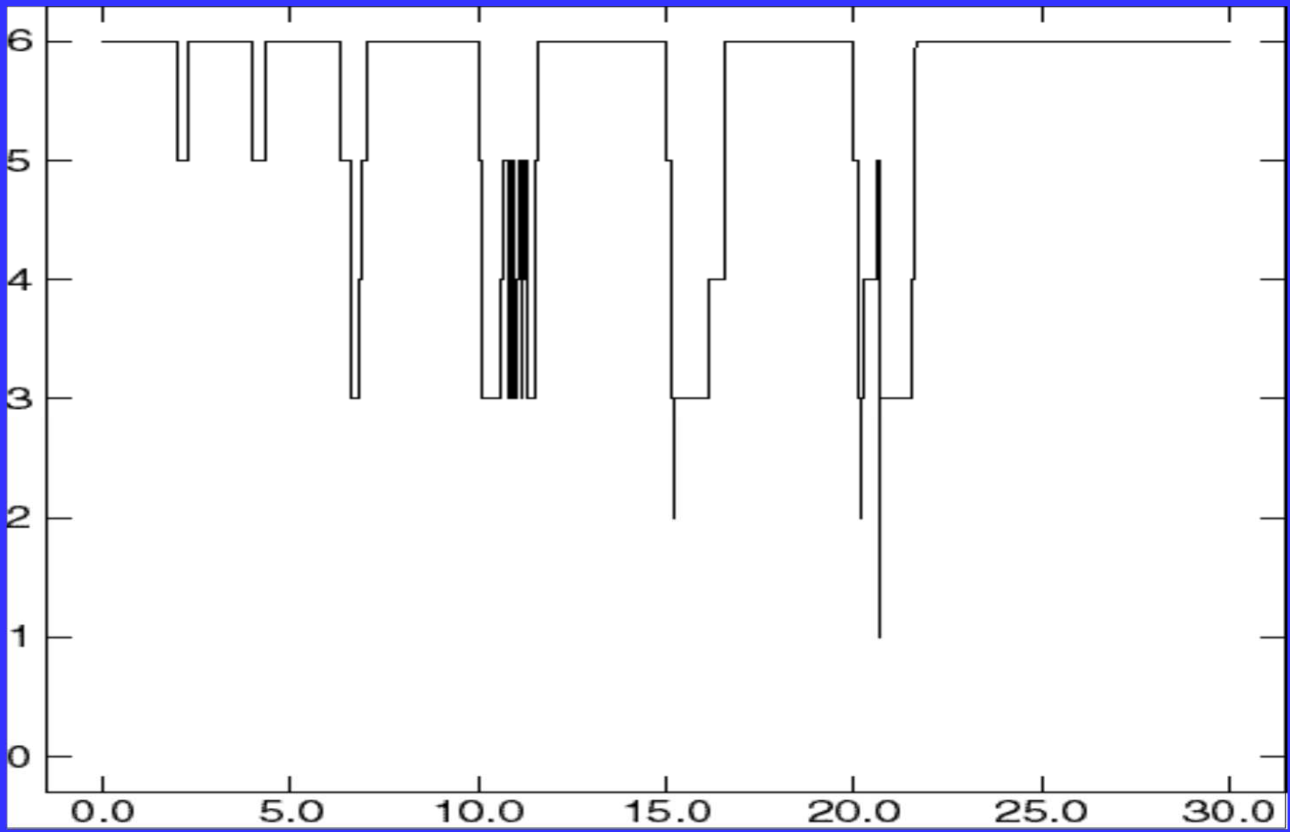
- 3D pose tracked.
- Eyes, nostrils tracked as convex holes
- Mouth tracked by upper lip intensity valley
  - **(Moses et al. 95)**
- Pose estimation using weak perspective
  - **(Gee & Cipolla 96)**

Example

Green: tracking   Red: searching

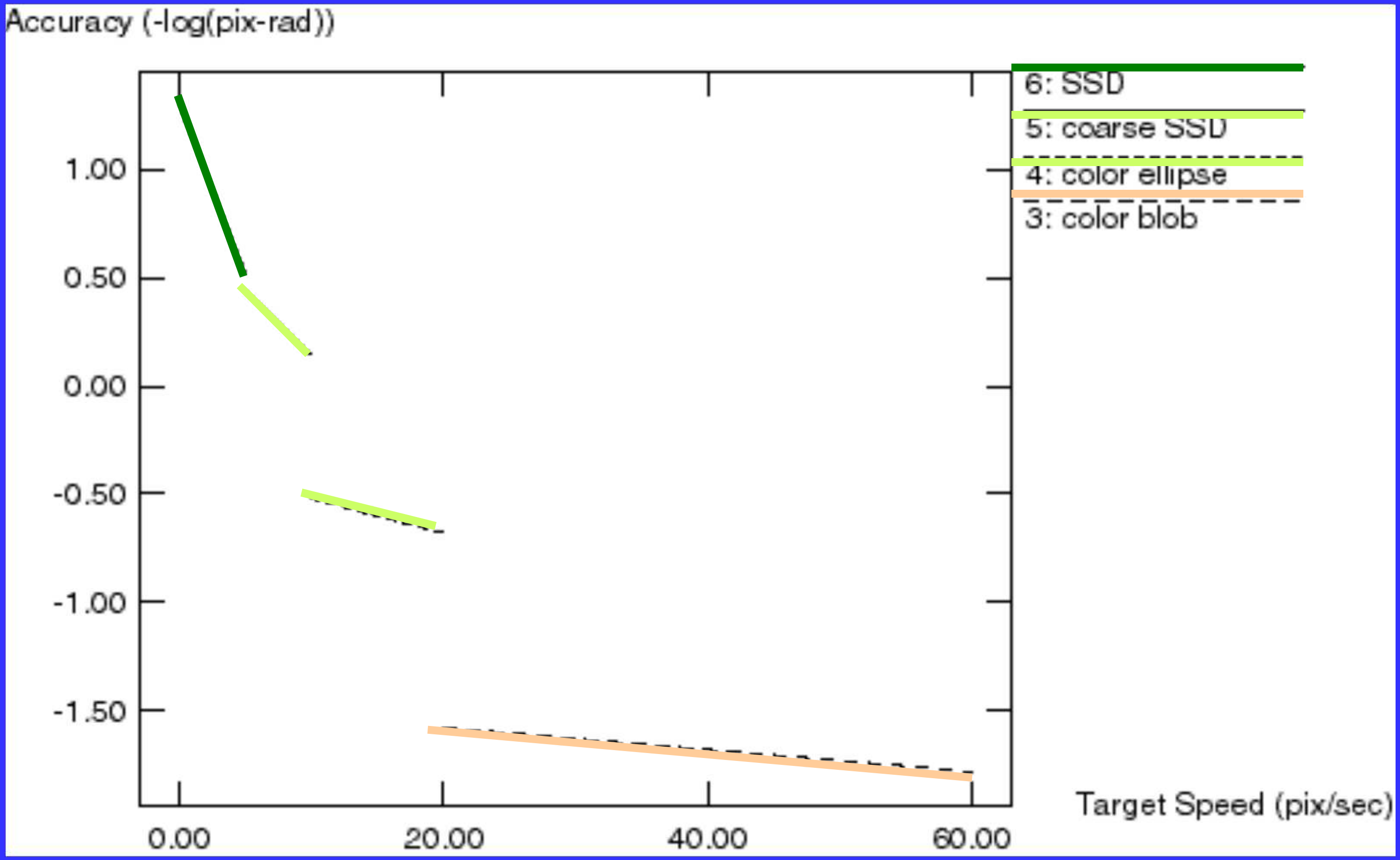
Layer Transitions

Layer

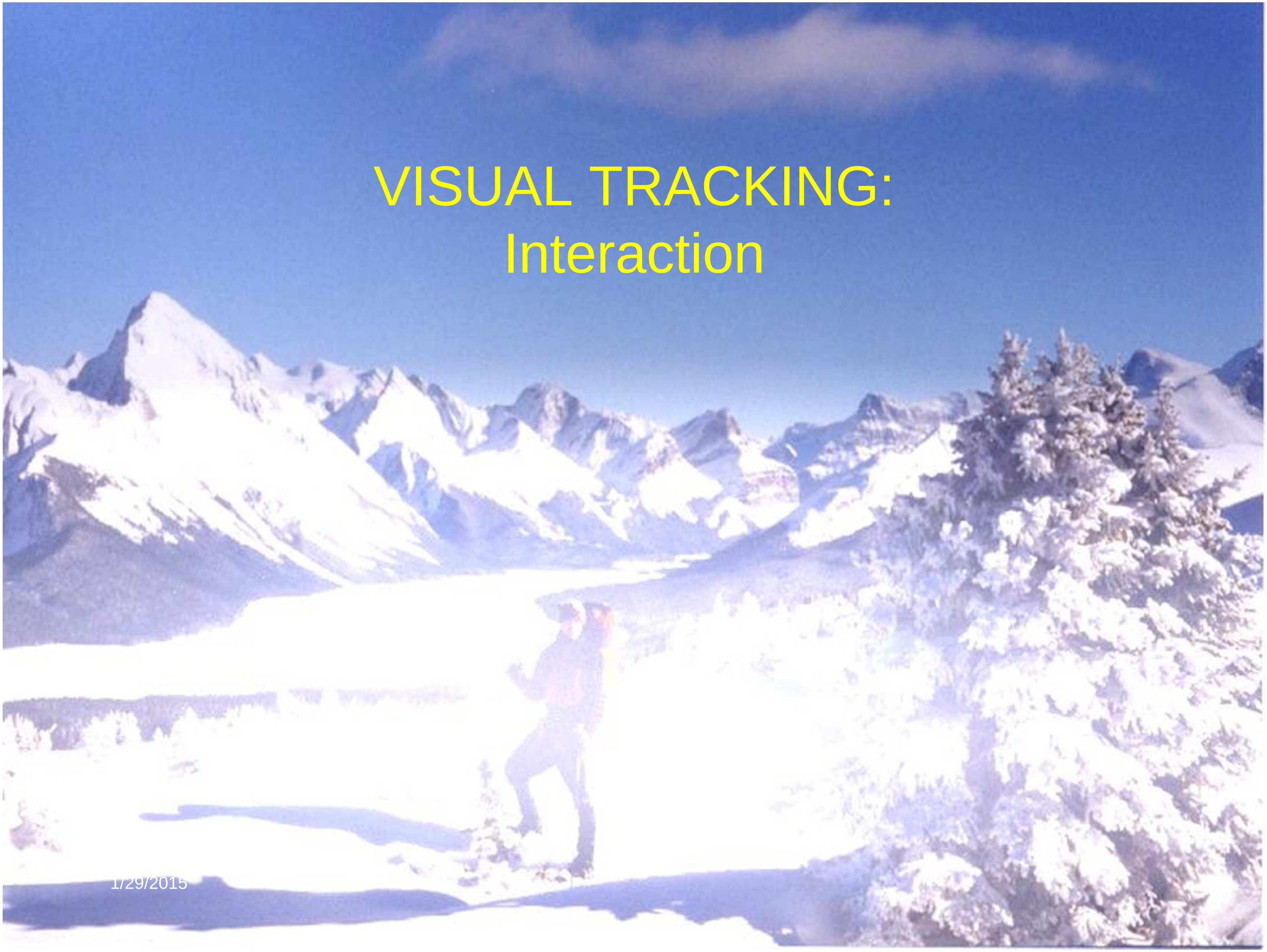


Time (sec)

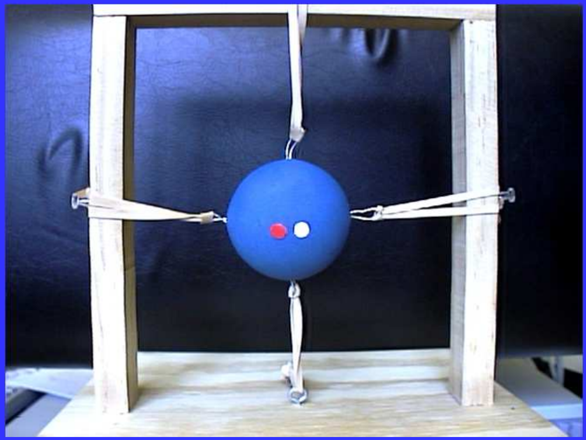
Resource Management

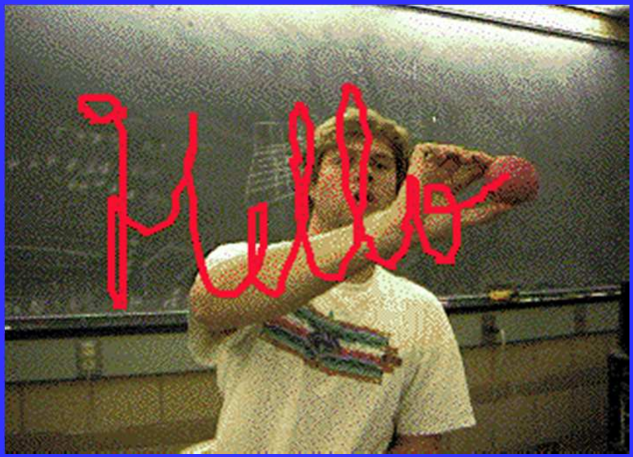


VISUAL TRACKING: Interaction

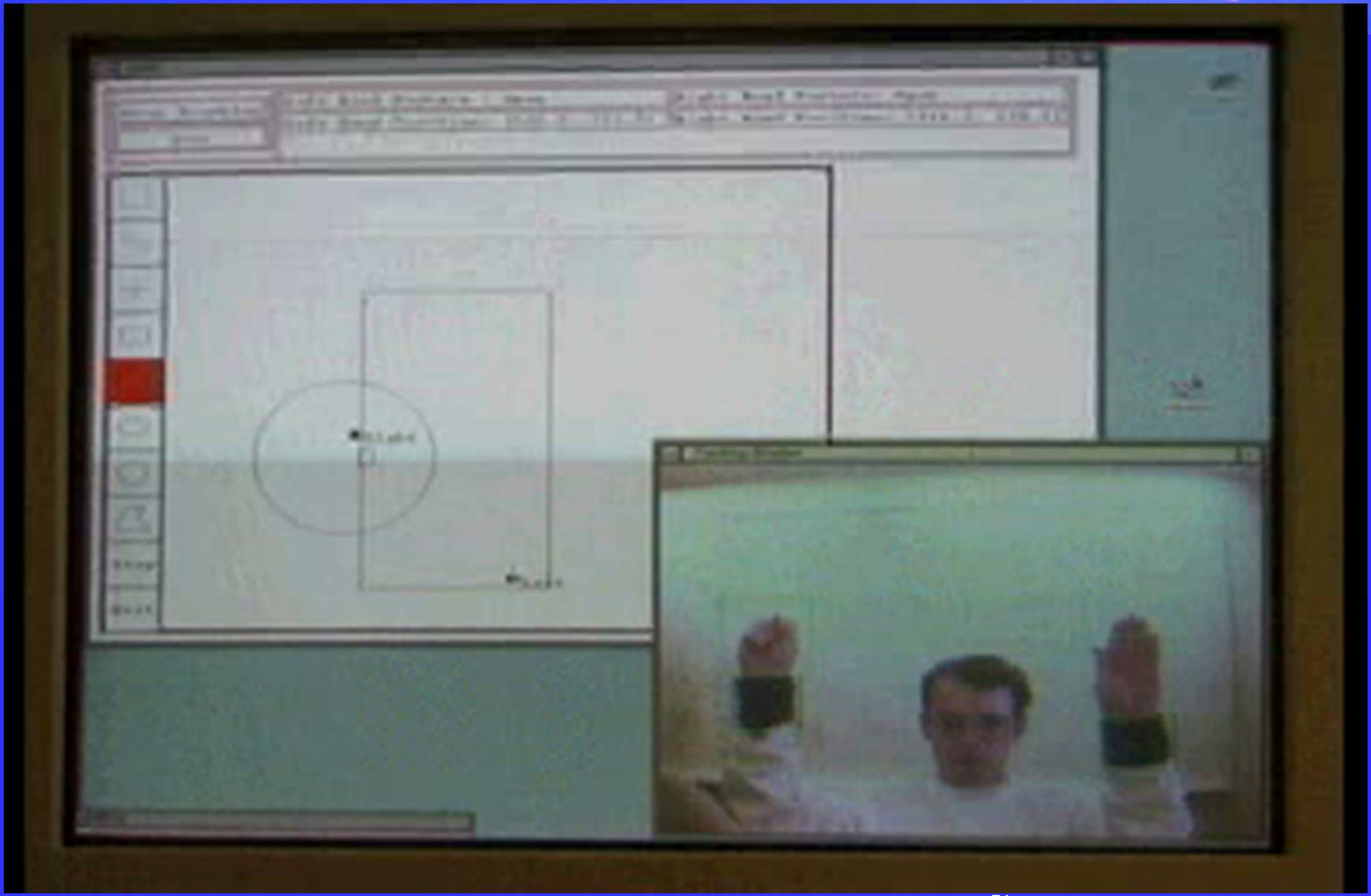


Human-Computer Interaction





Human-Computer Interaction



Human-Computer Interaction

Video Mirroring

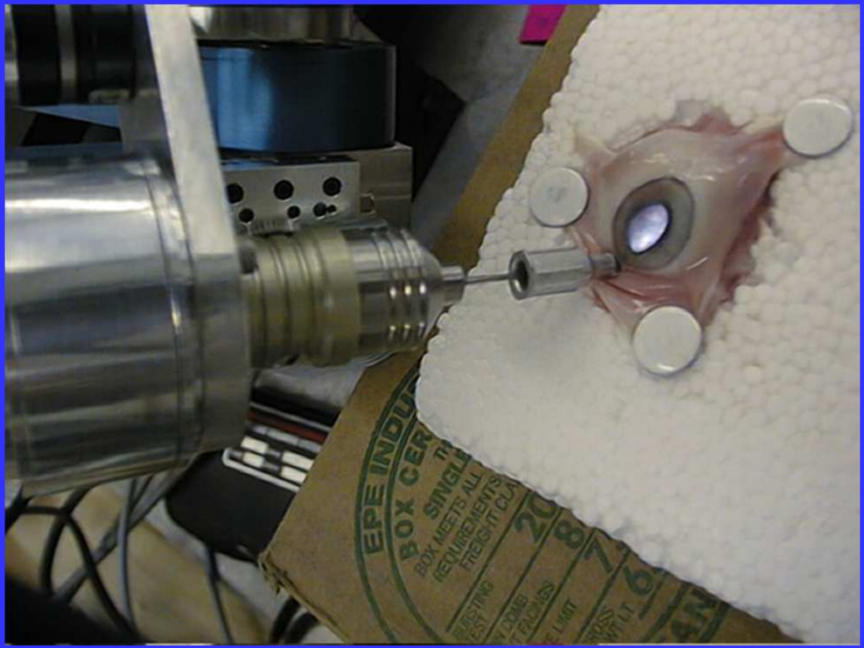
The Future

Hardware

- Firewire
  - 400 mbits/sec (PCI speed)
  - Controllable compression (lossy)
  - OHCI chipsets supported under Linux
- CMOS-based chips
  - Region Saddressing
  - High frame rates
- On chip acceleration
  - IVL gives order of magnitude performance increase

Tracking for Deformable Structures

- Most tracking work is rigid
  - One exception --- snakes/splines
- Biological motion is underlying physical basis
  - Repetition (breathing)
  - Deformation (touching, manipulating, puncturing)



Virtual Visual Objects: An Approach to HCI

- Use a video mirror (one or two cameras)
- Define a set of “interaction icons” that are visible to user
- Detect and parse movements toward and within those regions
- Use vision-based task algebra as a basis for defining geometric interactions.

Conclusion

Interactive Systems = Vision + Control

…we tend to bring any object that attracts our attention into a standard position and orientation so that it varies within as small a range as possible. This does not exhaust the processes which are involved in perceiving the form and meaning of the object, but it certainly facilitates all later processes tending to this end.

Norbert Wiener
1948

Snakes

- Contour C: continuous curve on smooth surface in $\Re^3$
- Snake S: projection of C to image
- Curve types
  - Edge between regions on surface with contrasting properties
  - Line that contrasts with surface properties on both side
  - Silhouette of surface against contrasting background
- General Algorithm:
  - Perform edge detection
  - Fit parametric or non-parametric curve to data

Snakes: Basic Approach

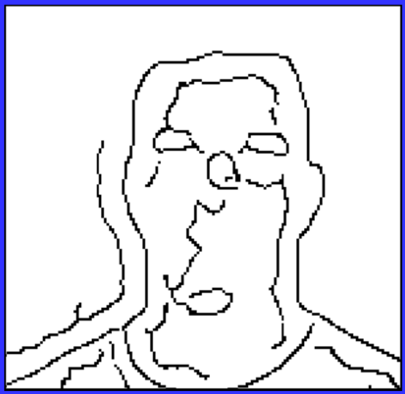
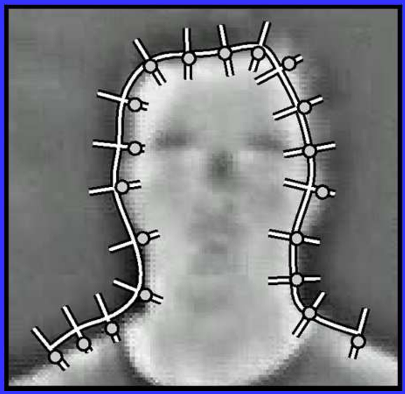
- Parameterize a closed contour

$$\mathbf{Q} = (q_0^x .. q_n^x, q_0^y .. q_n^y)$$

$$\mathbf{U}(s) = \begin{pmatrix} \mathbf{B}(s)^t & 0 \\ 0 & \mathbf{B}(s)^t \end{pmatrix}$$

- $\mathbf{r}(s) = \mathbf{q}^t \mathbf{B}(s)$ or $\mathbf{r}(s) = \mathbf{U}(s)\mathbf{Q}$

- Given a predicted state **q,** search radially for edges
- Solve a least squares problem for new state

Snake: Details

- Constrained deformations of B-spline templates [Blake *et al*., 1993]
  - $\mathbf{z}_i$: nearest edge at point $\mathbf{r}(s_i)$ along the curve with normal $\mathbf{n}(s_i)$
  - X' and S' existing state and weight (covariance) estimate

    1. $\mathbf{Z_0} = 0$, $S_0 = 0$

    2. Iterate i = 1..N

    1. $v_i = (\mathbf{z_i} - \mathbf{r}(s_i)) \,.\, \mathbf{n}(s_i)$
    2. $\mathbf{h}(s_i)^t = \mathbf{n}(s_i)^t\, U(s_i)\, W$
    3. $S_i = S_{i-1} + 1/q_i^2\, \mathbf{h}(s_i)\mathbf{h}(s_i)^t$
    4. $\mathbf{Z}_i = \mathbf{Z}_{i-1} + 1/q_i^2\, \mathbf{h}(s_i) v_i$

    3. Compute

    1. $X = X' + (S' + S_N)^{-1}\mathbf{Z_N}$

optional shape template where $Q = W\mathbf{X} + Q_0$

Snake: Alternative Approach

- Perform gradient ascent on

$$p_{snake}(\mathbf{I} \mid \mathbf{X}) = \exp\left(-\frac{1}{\sigma_{snake}^2} \sum_{i=0}^{N} l(i) \cdot \psi_{snake}(i)\right)$$

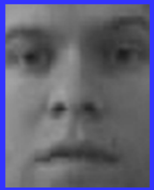
$$\psi_{snake}(i) = \begin{cases} |\mathbf{\Lambda}(i) - \mathbf{z}(i)| & \text{if an edge is found} \\ \xi & \text{otherwise} \end{cases}$$

Textured Region

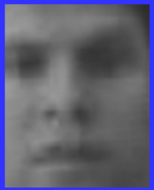
- Mean intensity difference between **I** and affine warp of template image [Shi & Tomasi, 1994]

$$p_{tregion}(\mathbf{I} \mid \mathbf{X}) = \exp\Big(-\frac{1}{\sigma^2_{tregion}} \sum_{x,y \in \mathbf{I}_R} a(x,y) \cdot \psi_{tregion}(x,y)\Big)$$

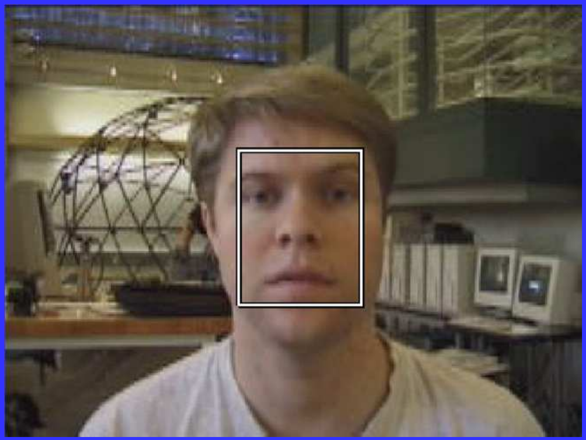
$$\psi_{tregion}(x,y) = (\mathbf{I}_R(x,y) - \mathbf{I}_C(x,y))^2$$



$\mathbf{I}_R$



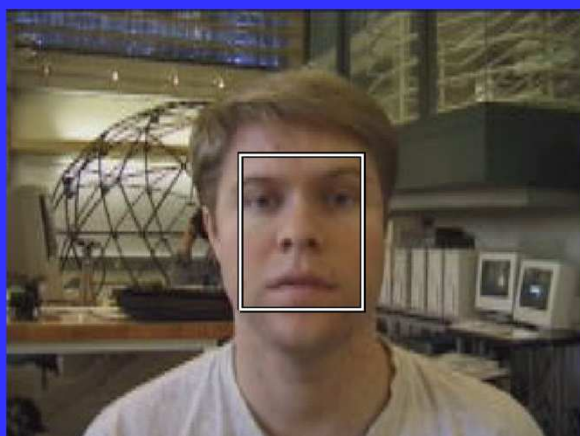
$\mathbf{I}_C$

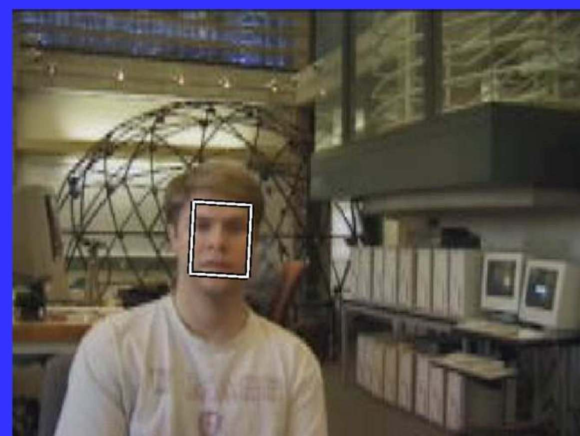


$|\mathbf{I}_R - \mathbf{I}_C|$

Textured Region

- Mean intensity difference between **I** and affine warp of template image [Shi & Tomasi, 1994]

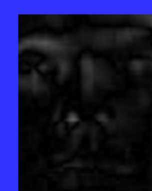




$\mathbf{I}_R$

$\mathbf{I}_C$

$$\psi_{tregion}(x,y) = \sum_{(x,y)'\in W} (\mathbf{I}_R(x,y) - \mathbf{I}_C(x,y))^2$$



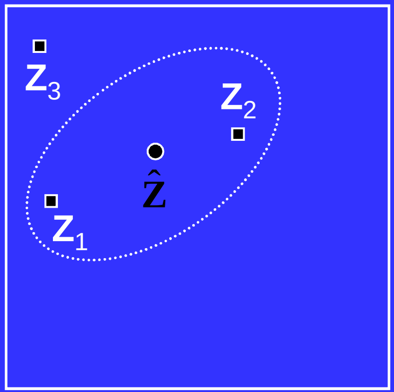
$|\mathbf{I}_R - \mathbf{I}_C|$

Illumination

Probabilistic Data Association Methods

(with Christopher Rasmussen, NIST)

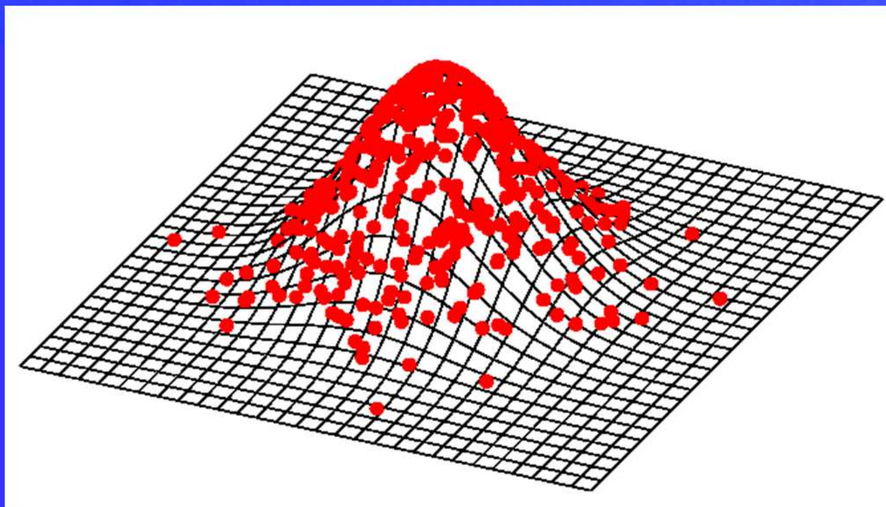
- Developed for tracking aircraft radar blips [Bar-Shalom & Fortmann, 1988]
- Assumes one measurement due to target, others from noise
- Multiple measurements weighted by *association probabilities* proportional to distance from predicted measurement
- Target-derived measurement dominates estimation



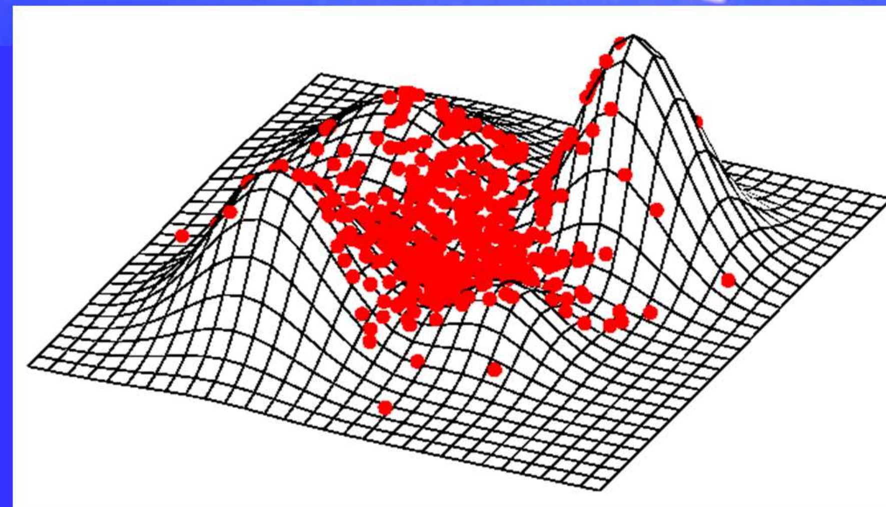
Finding Measurements

- Look for peaks in $p(\mathbf{I} \mid \mathbf{X}); p(\mathbf{X})$ suggests where to search
- Gradient ascent [Shi & Tomasi, 1994; Terzopoulos & Szeliski, 1992]
  - Identifies nearby, good hypothesis
  - May pick incorrectly when there is ambiguity
  - Vulnerable to agile motions
- Random sampling [Isard & Blake, 1996]
  - Approximates local structure of image likelihood
  - Identifies alternatives
  - Resistant to agile motions

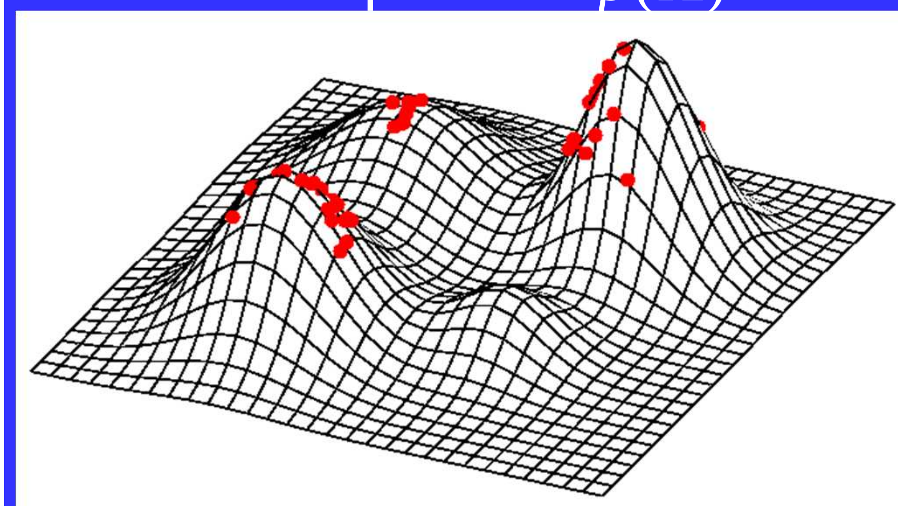
Measurement Generation



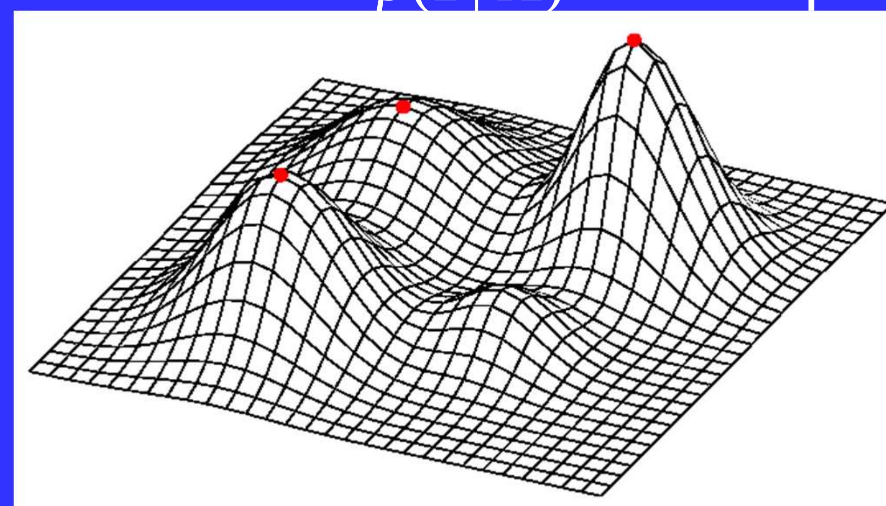
Sample from $p(\mathbf{X})$



Evaluate $p(\mathbf{I} \mid \mathbf{X})$ at samples



Keep high-scoring samples



Ascend gradient & pick exemplar

Measuring: Textured Regions

Predicted state

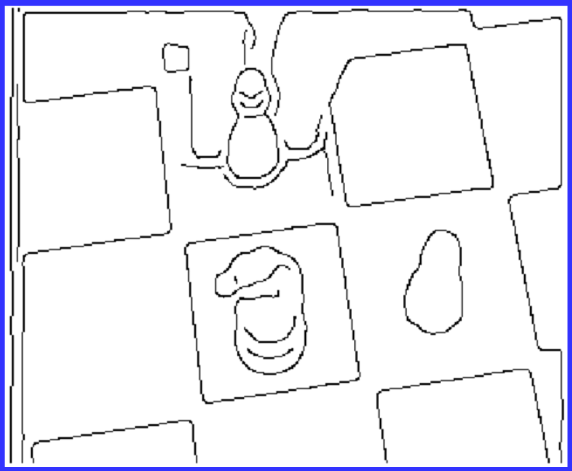
Initial samples

Top fraction

Hill-climbed

Measuring: Snakes

Predicted state

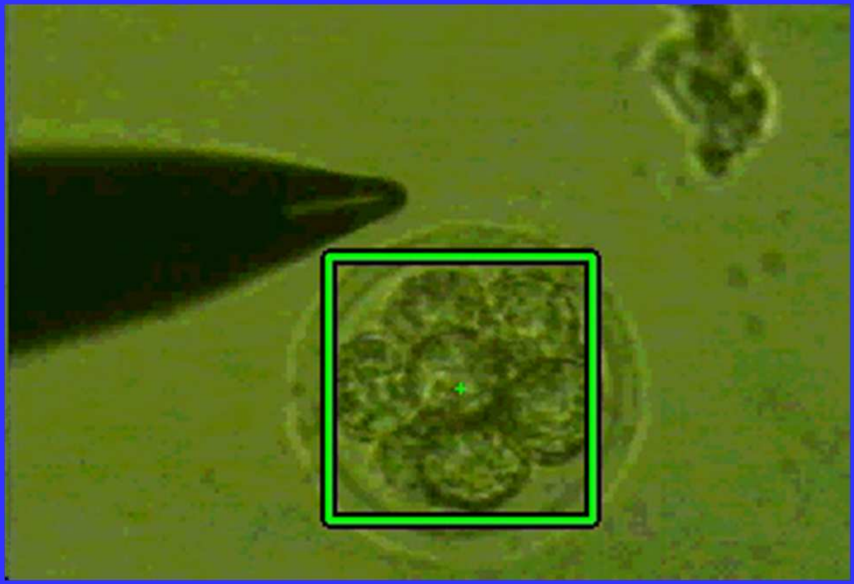


Canny edges

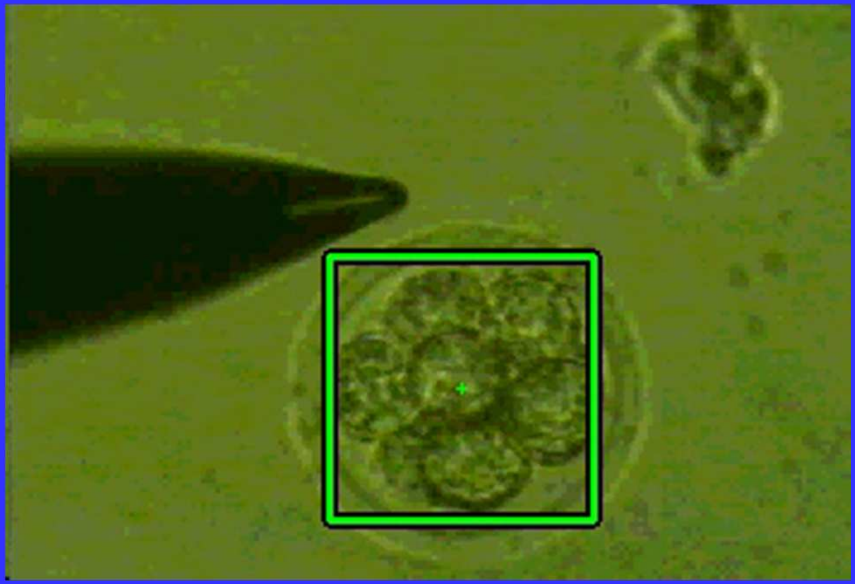
Initial samples

Top fraction

PDAF: Agile motion with a textured region



Gradient ascent



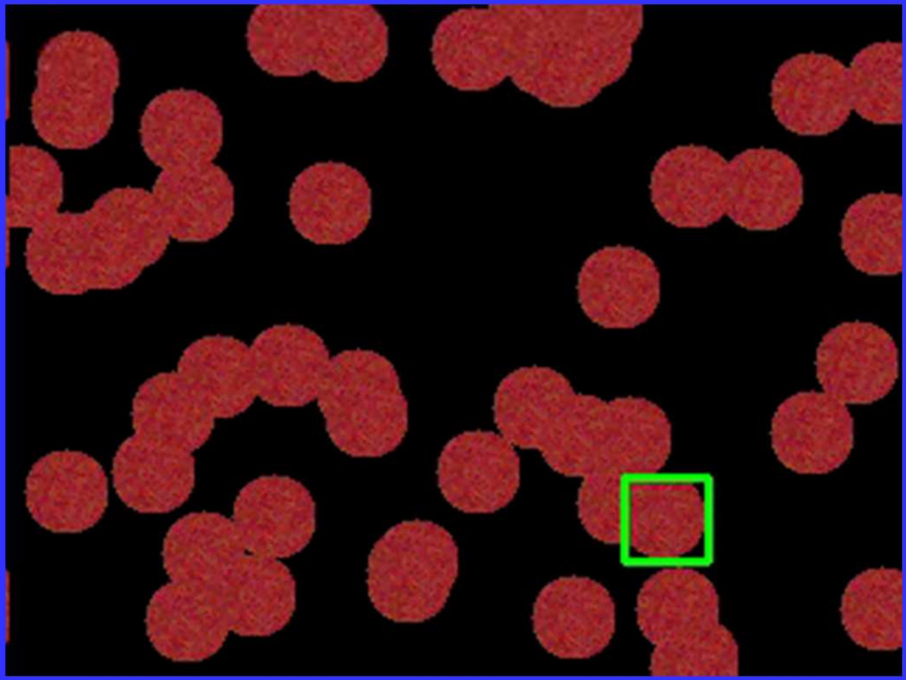
Random sampling

PDAF: Details

- Kalman filter innovation $\boldsymbol{\nu} = \mathbf{z} - \hat{\mathbf{z}}$, where $\hat{\mathbf{z}} = \mathbf{H}(\mathbf{X})$, becomes $\boldsymbol{\nu} = \sum_{i=1}^{n} \beta_i \boldsymbol{\nu}_i$ for n measurements
- Each association probability β_i is proportional to $\exp(-\frac{1}{2}\boldsymbol{\nu}_i'\mathbf{S}^{-1}\boldsymbol{\nu}_i)$
- $\beta_0 = 1 - \sum_{i=1}^{n} \beta_i$ is the probability that none of the measurements are due to the target
- Validation gate: ellipsoid in measurement space defined by $\{\mathbf{z} : \boldsymbol{\nu}_i'\mathbf{S}^{-1}\boldsymbol{\nu}_i \leq \alpha\}$

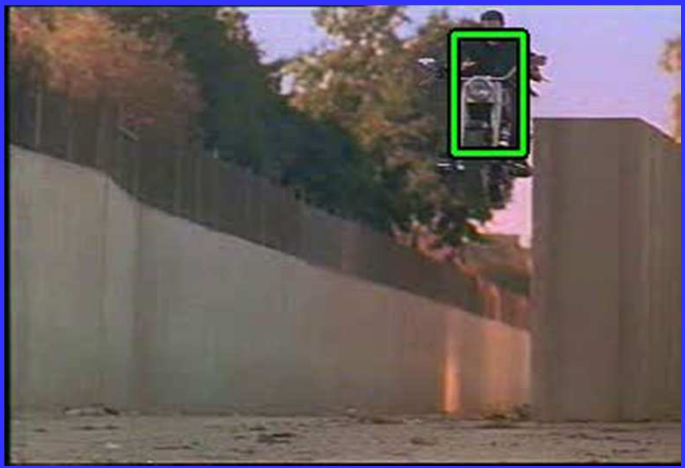
PDAF: Multiple measurements counteract noise

Tracking an orbit (50 distractors)
1 measurement: 5/20 successes
10 measurements: 17/20



Other PDAF results

Homogeneous region (measurements)



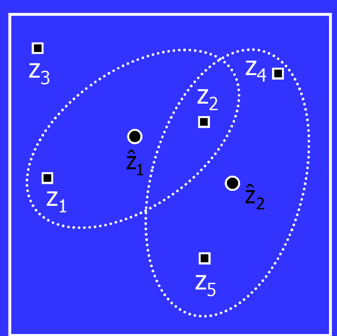
Textured region

Two snakes

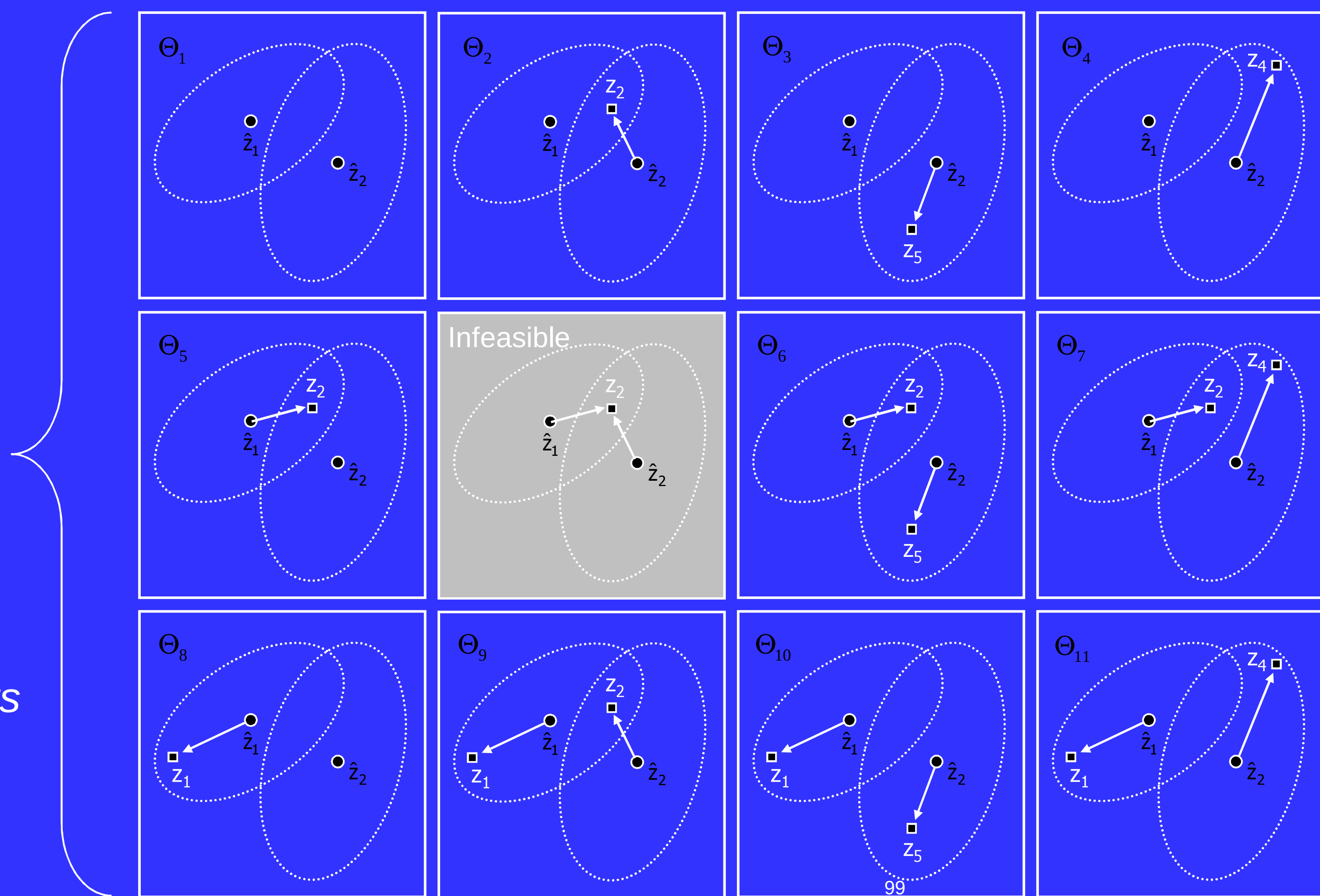
Joint Probabilistic Data Association Filter (JPDAF)

- Extension of PDAF to multiple objects
  [Bar-Shalom & Fortmann, 1988]
- With *N* persistent targets and noise, compute association probabilities jointly
- Enforcing *feasibility* avoids double-counting
  - Each measurement has exactly one source
  - Each target gives rise to at most one measurement

JPDAF: Feasible Joint Events



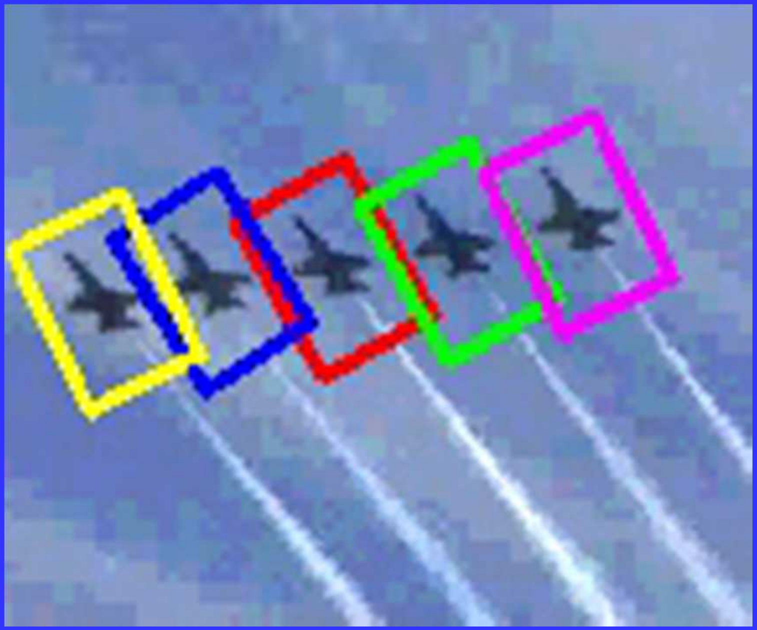
2 targets,
5 measurements



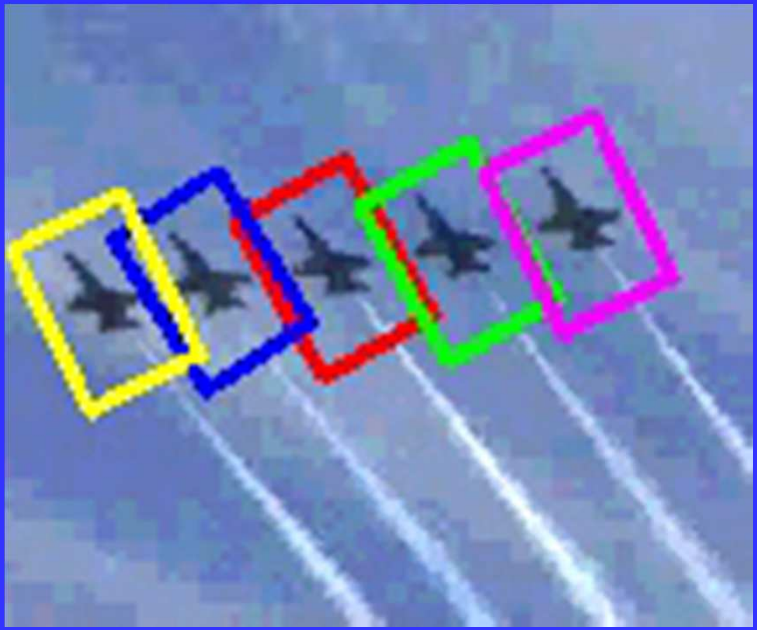
JPDAF algorithm

- ❶ Hypothesize *feasible* joint association events Θ
- ❷ Compute joint event probabilities $P(\Theta \mid \mathbf{Z})$
- ❸ Calculate association probabilities
  $\beta_{jt} = \sum_{\Theta} P(\Theta \mid \mathbf{Z}) \omega_{jt}(\Theta)$, where $\omega_{jt}(\Theta) = 1$
  if measurement *j* is associated with target *t* in Θ and
  0 otherwise

JPDAF: Nearby textured regions

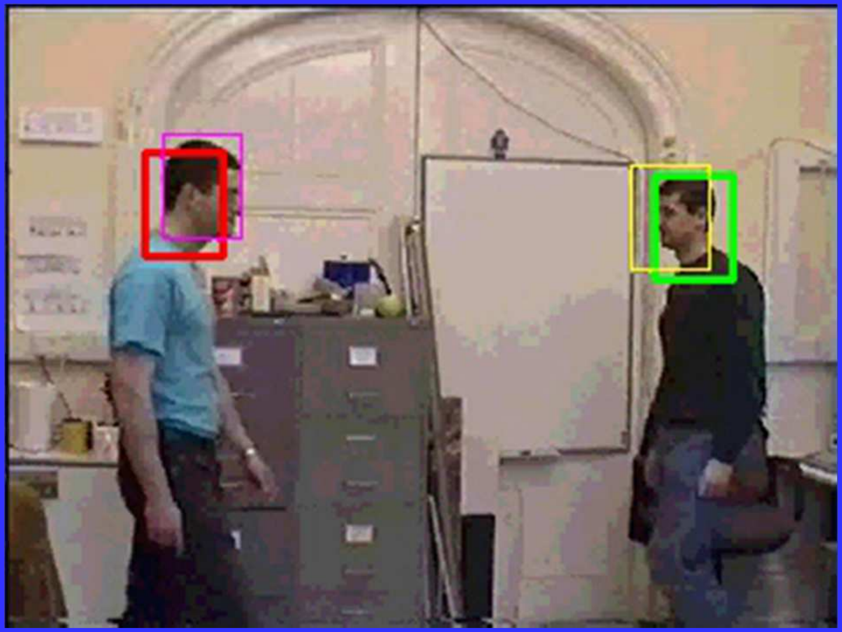


PDAF

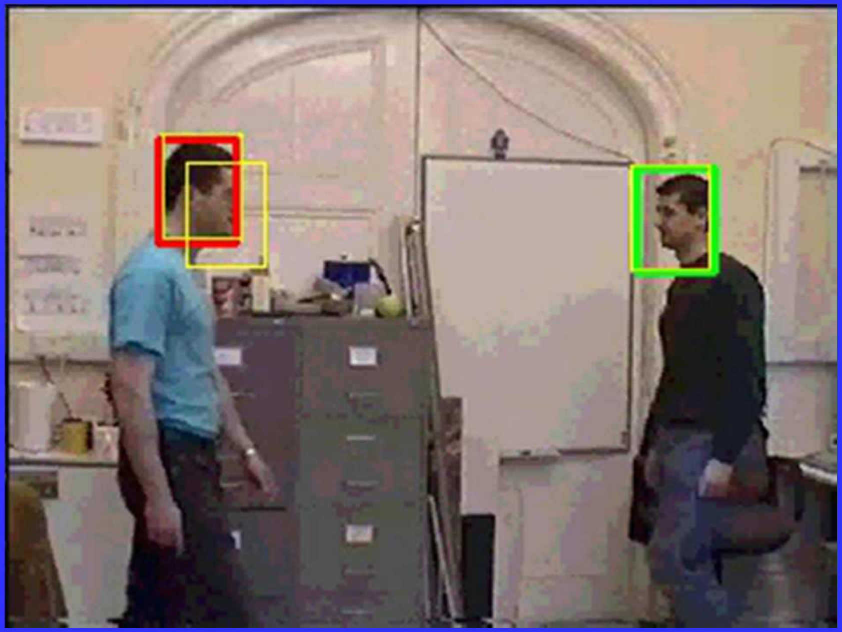


JPDAF

JPDAF: Crossing homogeneous regions

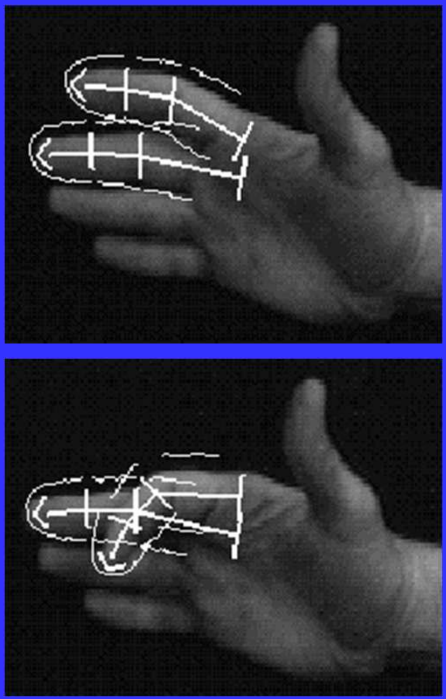


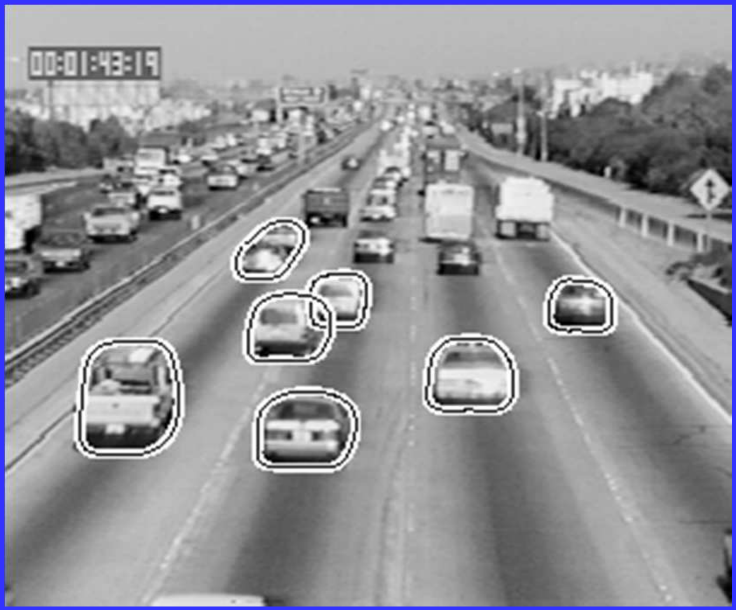
PDAF



JPDAF

A Final Issue: Tracking with Occlusion





Rehg & Kanade, 1994

Koller, Weber, & Malik, 1994

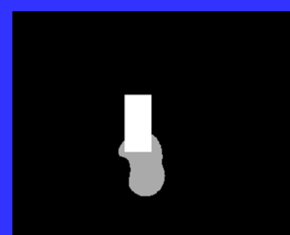
Joint Likelihood Filter (JLF)

- When objects overlap, joint image likelihood

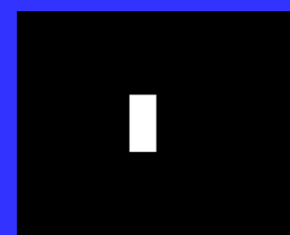
$$p(\mathbf{I} \mid \mathbf{X}_1, \ldots, \mathbf{X}_N) \neq p(\mathbf{I} \mid \mathbf{X}_1) \cdots p(\mathbf{I} \mid \mathbf{X}_N)$$

- Sample joint states $\mathbf{X}^J = (\mathbf{X}_1, \ldots, \mathbf{X}_N)$
  - Hypothesize visibility-affecting depth orderings $\mathbf{d}_i$
  - Evaluate image likelihoods using visibility masks $\mathbf{M}_j$

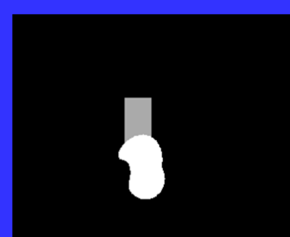
$\mathbf{X}^J$



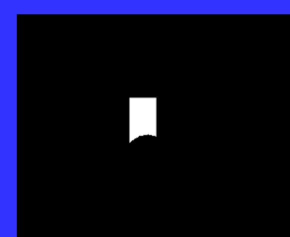
$\mathbf{d}_1$



$\mathbf{M}_{pawn}$



$\mathbf{d}_2$

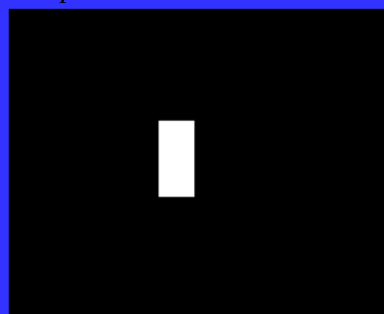


$\mathbf{M}_{pawn}$

JLF: Textured region component image likelihood

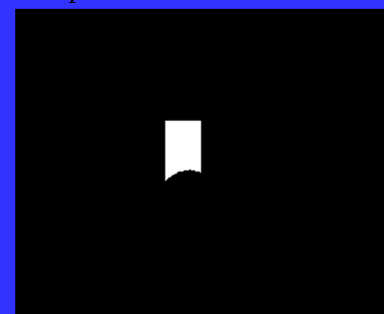
$\mathbf{I}_R$

$\mathbf{M}_{pawn}$



$\mathbf{I}_C$

$\mathbf{M}_{pawn}$

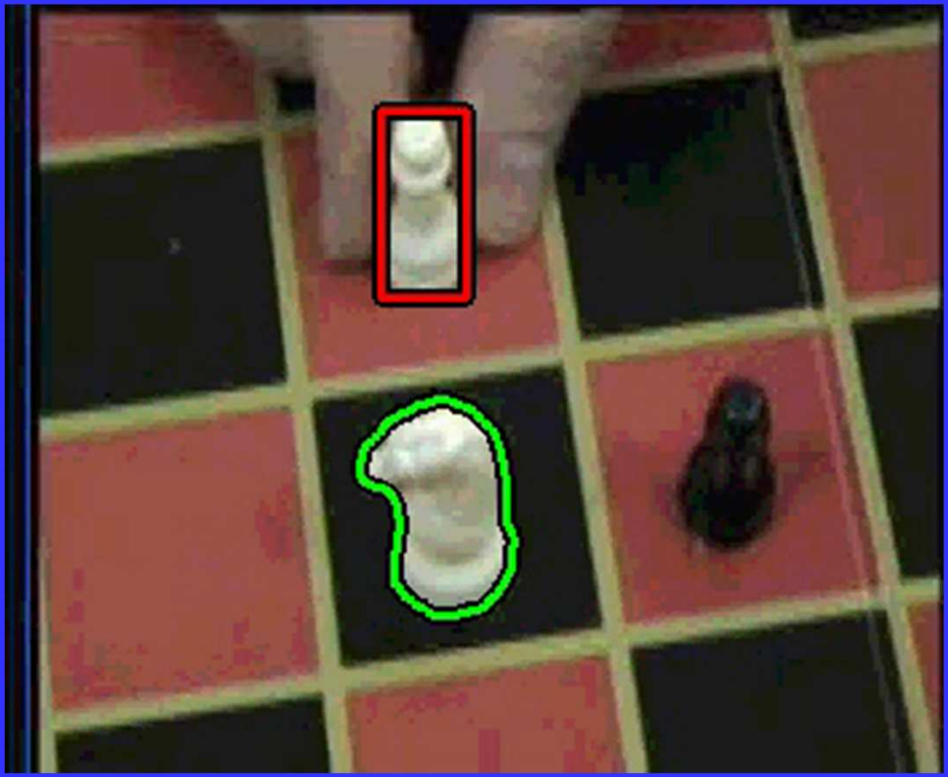


$\mathbf{I}_C$

$$p^J_{tregion}(\mathbf{I} \mid \mathbf{X}_j) = \mathrm{sig}\Big(\sum_{x,y \in \mathbf{I}_R} a(x,y) \cdot \psi^J_{tregion}(x,y) \Big)$$

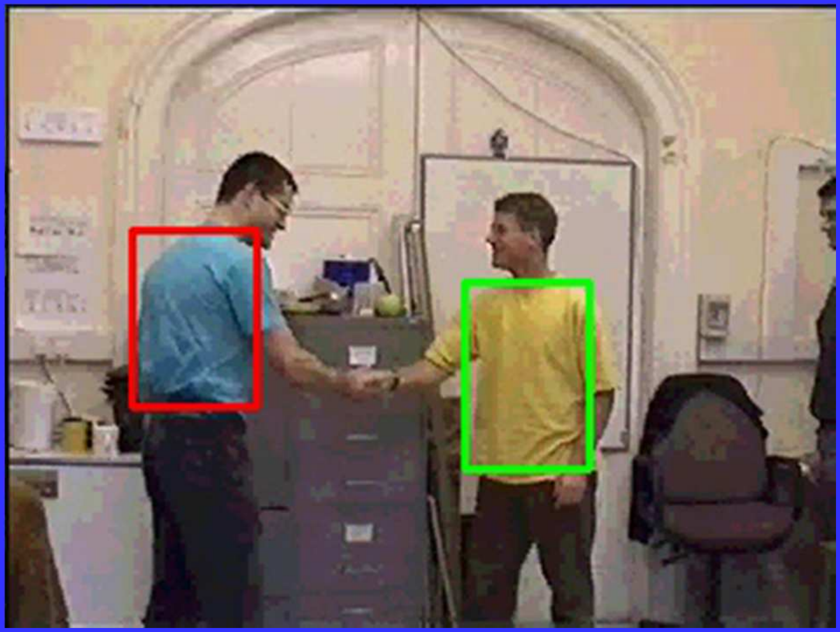
$$\psi^J_{tregion}(x,y) = \begin{cases} 1 & \text{if } \mathbf{M}_j(x,y) = 1 \wedge (\mathbf{I}_R(x,y) - \mathbf{I}_C(x,y))^2 \le \Upsilon_{tregion} \\ -1 & \text{if } \mathbf{M}_j(x,y) = 1 \wedge (\mathbf{I}_R(x,y) - \mathbf{I}_C(x,y))^2 > \Upsilon_{tregion} \\ 0 & \text{otherwise} \end{cases}$$

JLF: Deducing depth ordering



Textured region & snake

JLF: Other results



Homogeneous regions

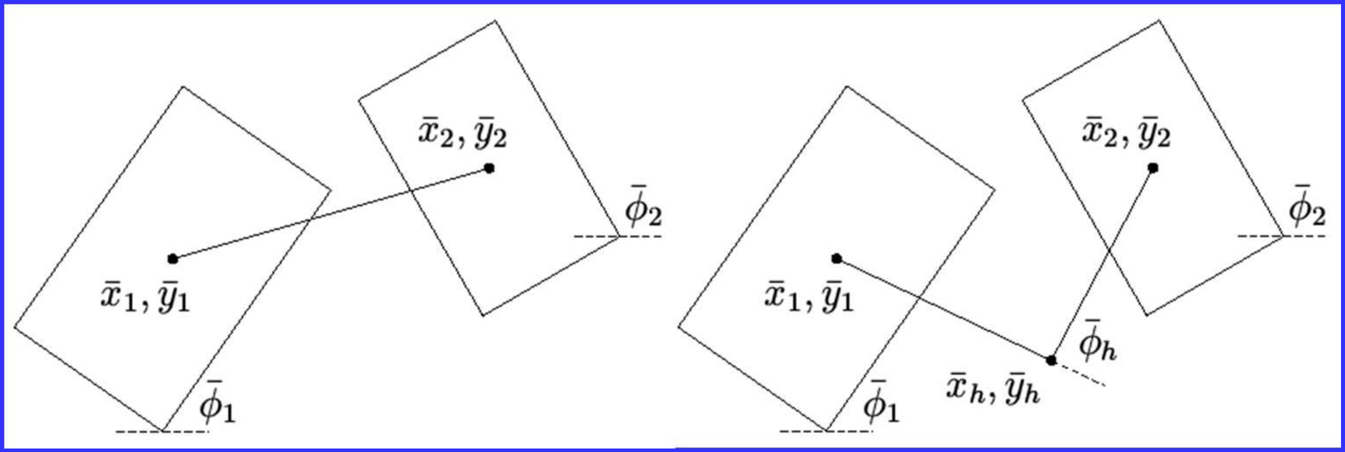
Textured regions

Constrained Joint Likelihood Filter (CJLF)

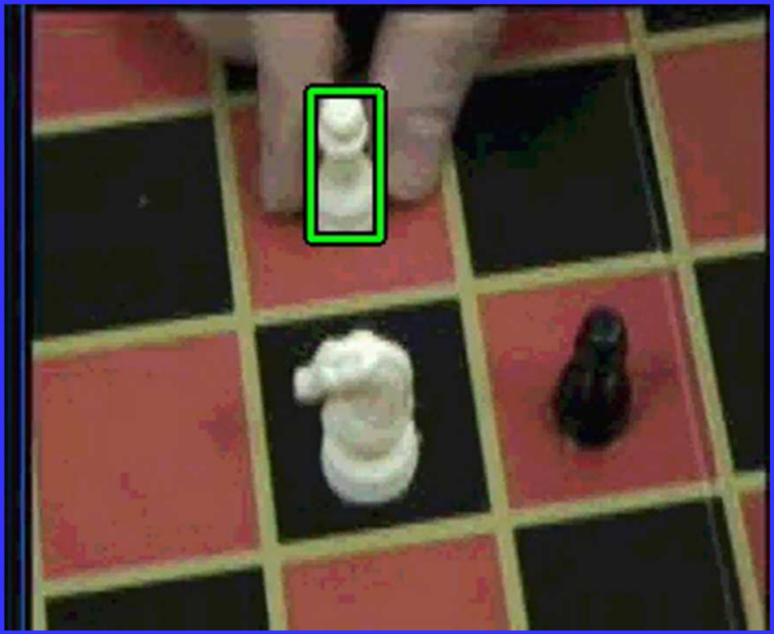
- When parts are linked, joint state prior

$$p(\mathbf{X}_1, \ldots, \mathbf{X}_N) \neq p(\mathbf{X}_1) \cdots p(\mathbf{X}_N)$$

- Write minimal state description: all joint states sampled meet constraints
- Kinds of constraints
  - Rigid link
  - Hinge
  - Depth



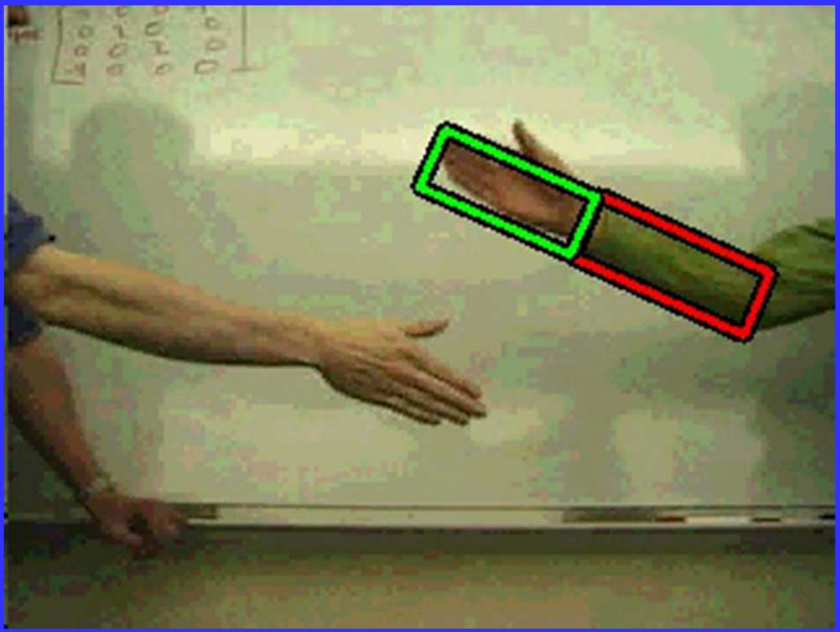
CJLF: Layered homogeneous region & snake



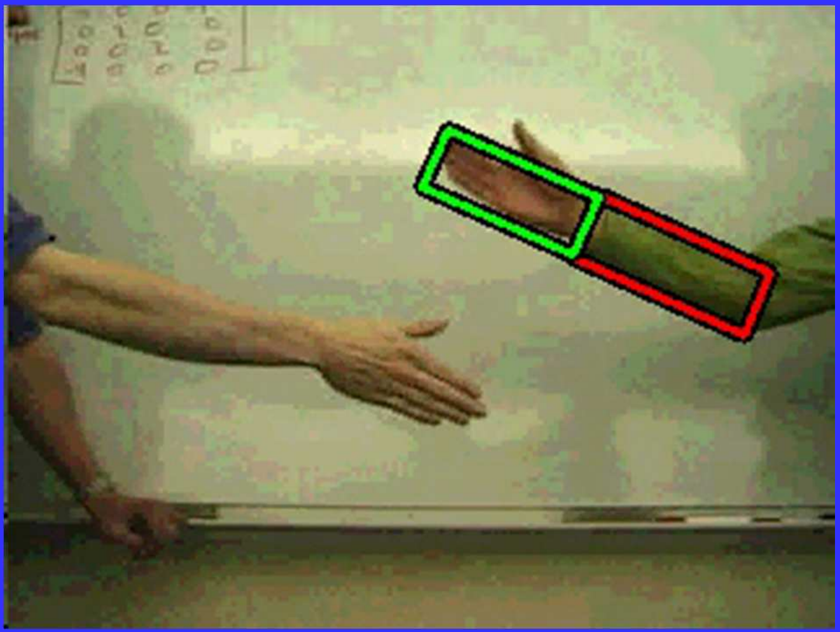
Homogeneous region

Homogeneous region and snake

CJLF: Hinge between homogeneous regions

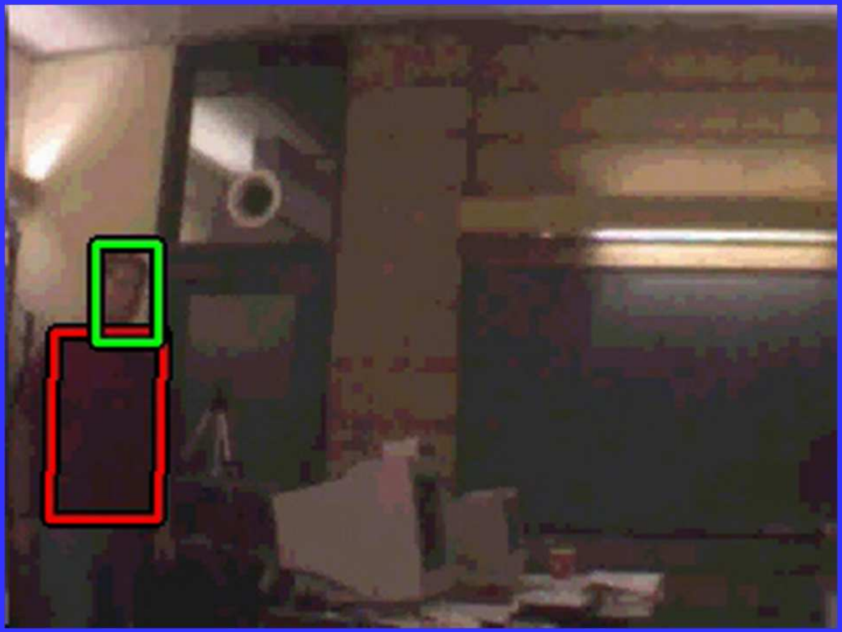


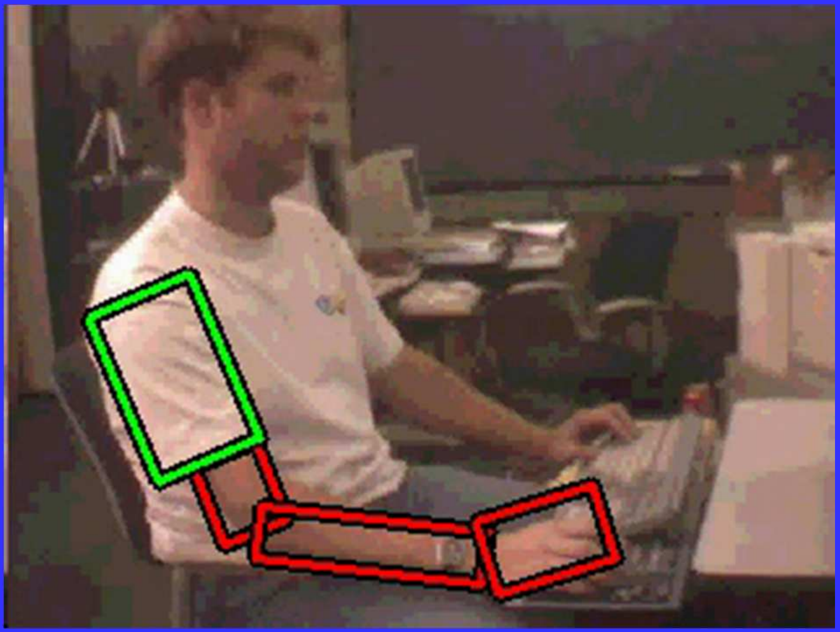
JLF



CJLF

Other CJLF results





Rigidly linked homogeneous regions

Kinematic chain of textured, homogeneous regions

Condensation (Blake/Isard)

- One problem in general is the nonlinearity of the basic problem
  - Outliers
  - Dynamics
  - Measurement functions

- Condensation is one of a class of "factored sampling" algorithms that seek to do "nonparametric" computation
  - Perform Bayes solutions using points

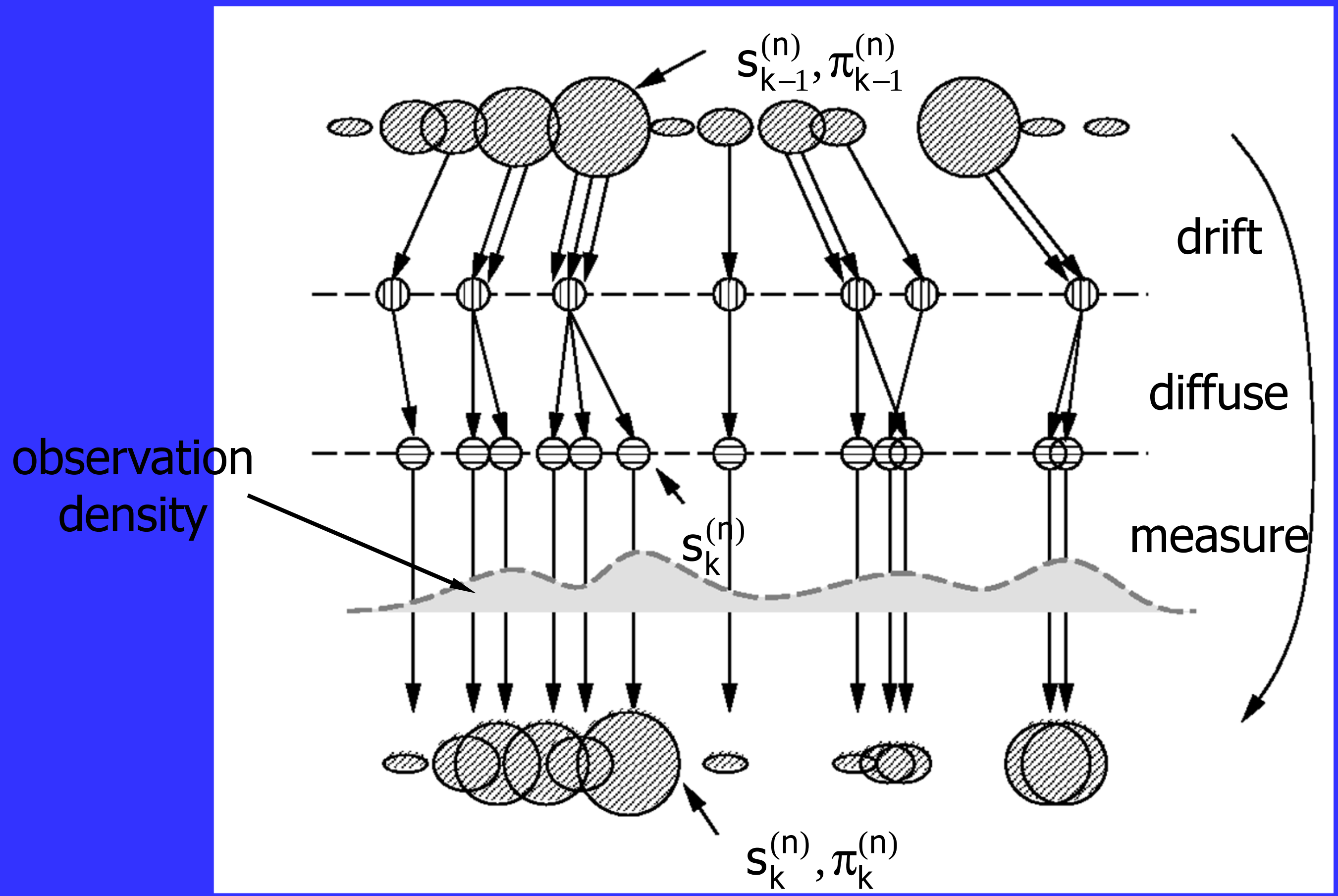
$$p(x \mid z) \propto p(z \mid x) p(x)$$

Condensation: Algorithm Details

(from Blake and Isard 98)

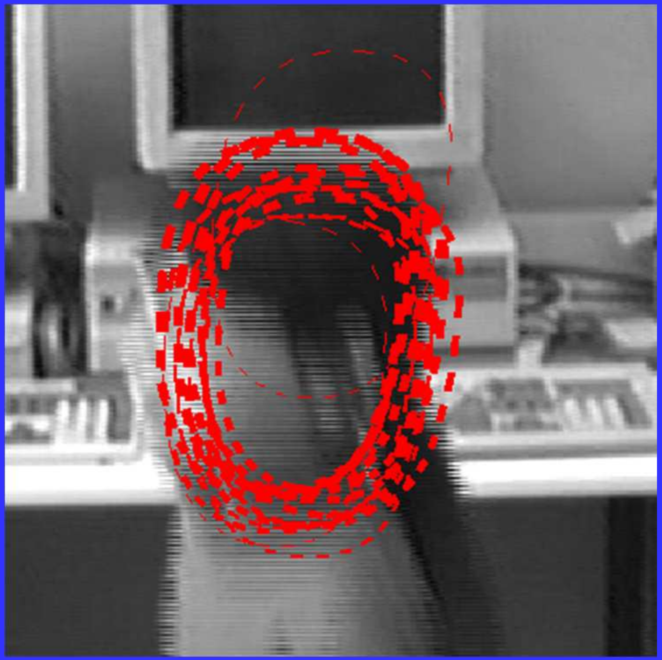
- Given N state samples $s_t(i)$ with weights (probabilities) $w_t(i)$ and cumulative probabilities $c_t(i)$
  1. Generate i=1..N new samples by
     1. uniformly choose k from [0,1]
     2. choose $s'_{t+1}(i) = s_t(j)$ where j is smallest index with $c_t(j) > k$
  2. Predict
     1. $s_{t+1}(i) = F(s'_{t+1}(i),w)$ where F is a dynamical model and w is process noise
  3. Measure **z** and compute
     1. $w_{t+1}(i) = p(z \mid x = s_{t+1}(i))$
     2. normalize so that $w_{t+1}(i)$ i=1..N sums to 1
     3. compute $c_{t+1}(i)$

CONDENSATION: Conditional density propagation



From *Isard & Blake, 1998*

CONDENSATION: Estimating Target State

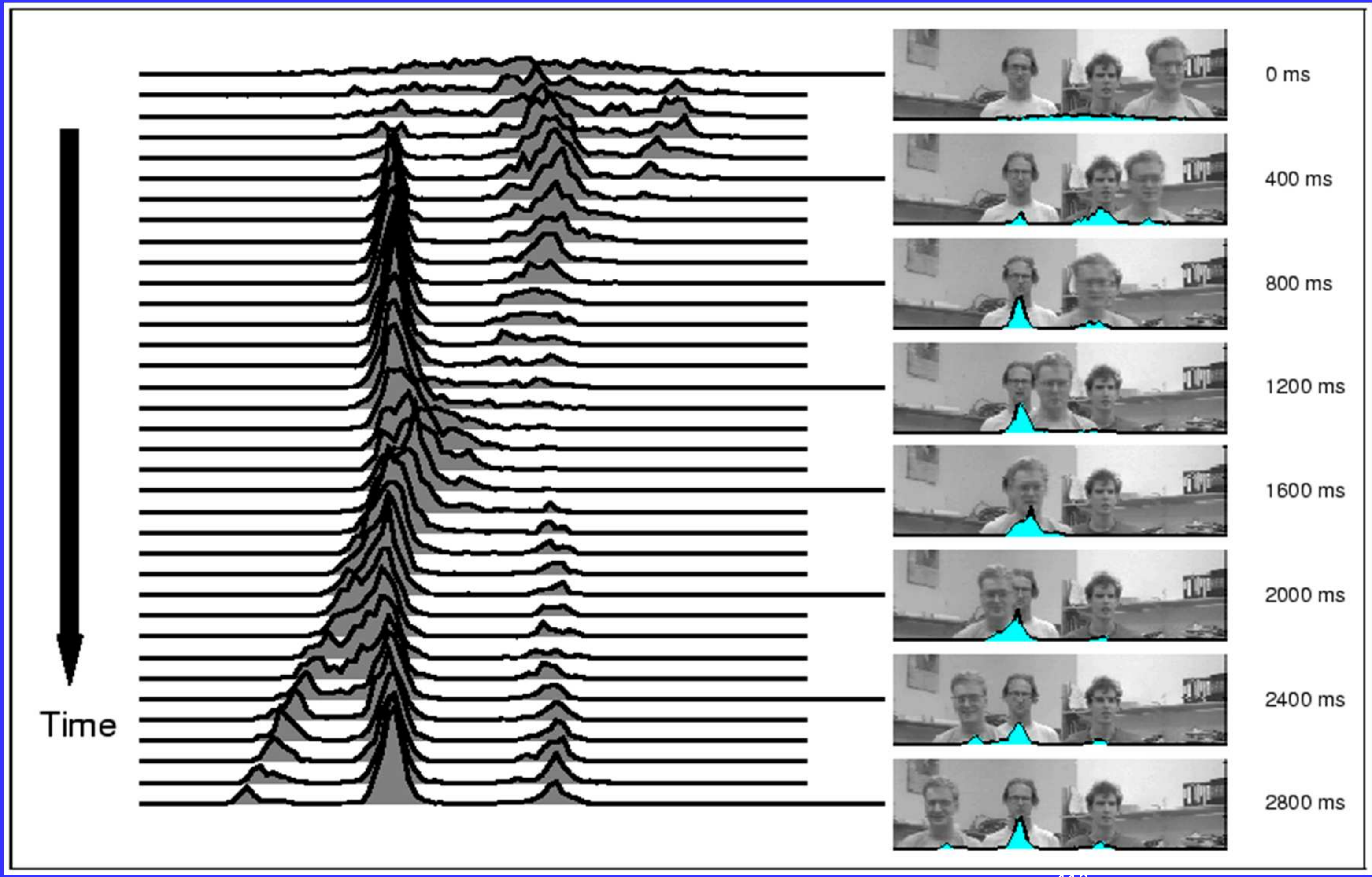


State samples

Mean of weighted state samples

From *Isard & Blake, 1998*

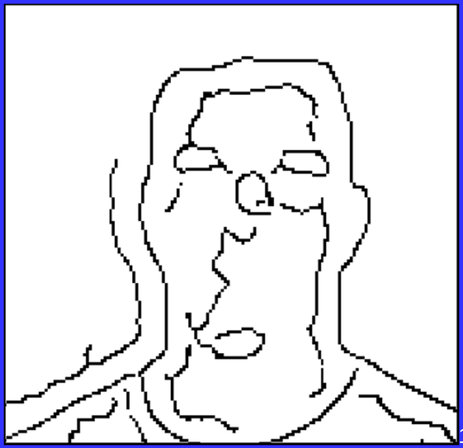
CONDENSATION: State posterior



Snakes: Edge Detection

Raw image

Sobel



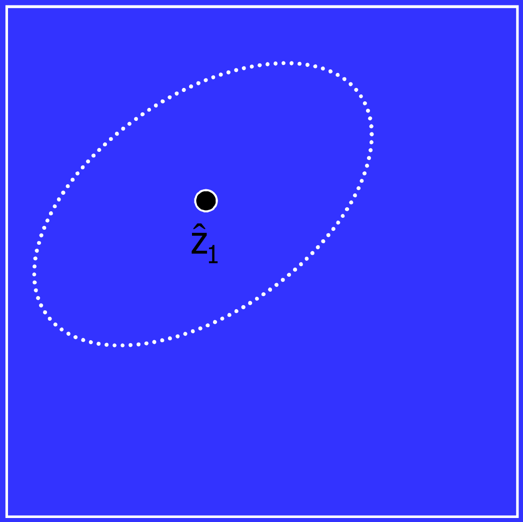
Canny

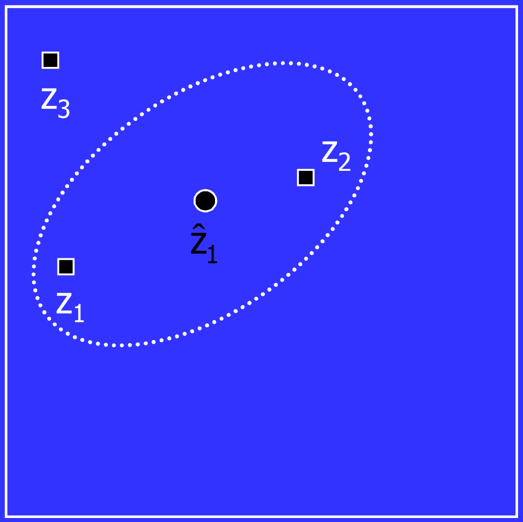
Measurement Generation

- Sample from prior
- Evaluate image likelihood and sort
- Keep top fraction
- Hill-climb and sort
- Enforce minimum separation
- Remaining samples become measurements

Dealing with Measurements

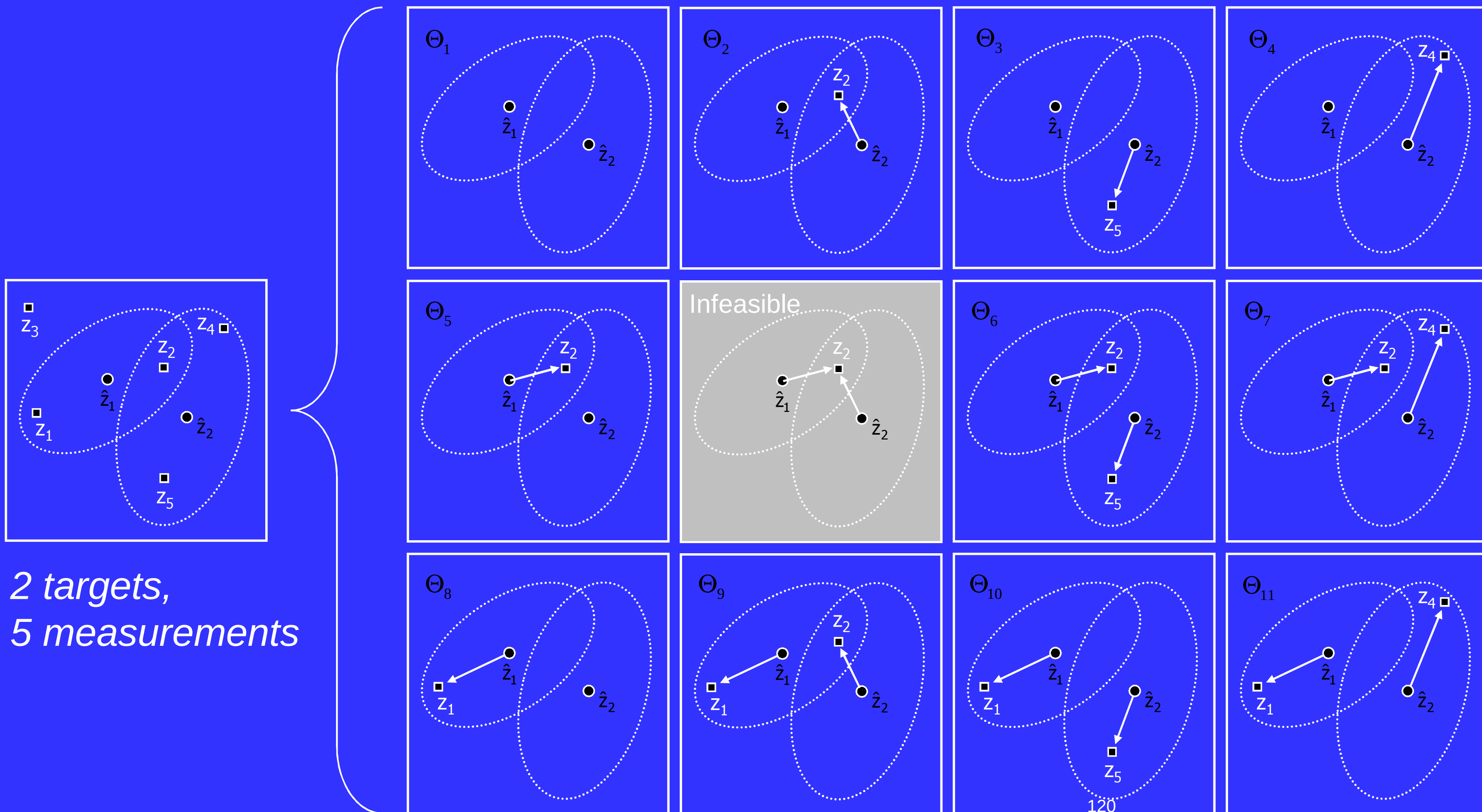
How do we update the Kalman filter when there are...





…no measurements? …multiple measurements?

JPDAF: Feasible Joint Events



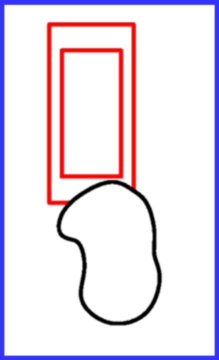
2 targets,
5 measurements

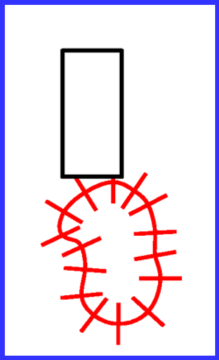
JPDAF's limitations

- Combinatorial expense
- Objects must be identical
- Overlaps not modeled

Joint Likelihood Filter (JLF)

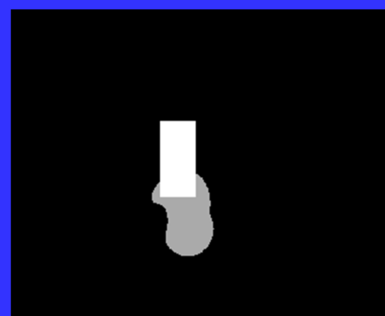
- Sample joint states $p(\mathbf{I} \mid \mathbf{X}_1, \ldots, \mathbf{X}_N) \neq p(\mathbf{I} \mid \mathbf{X}_1) \cdots p(\mathbf{I} \mid \mathbf{X}_N$
  - Hypothesize depth orderings, model occlusion interactions
  - Account for depth-independent interactions $\mathbf{X}^J = (\mathbf{X}_1, \ldots, \mathbf{X}_N)$
- Joint image likelihood
  - Redefine $p_a(\mathbf{I} \mid \mathbf{X})$ as *component* image likelihoods $p_a^J(\mathbf{I} \mid \mathbf{X}_j)$



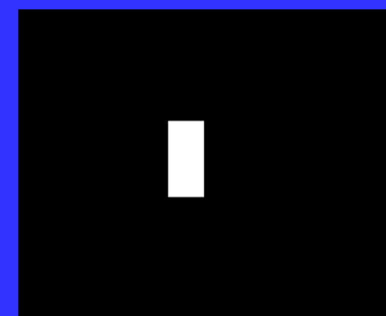


JLF: Occlusion Reasoning

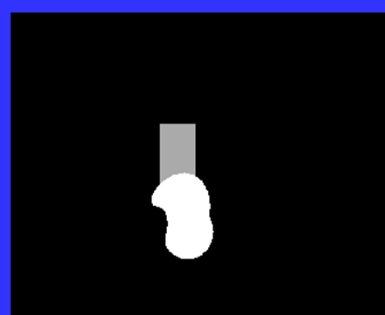
$\mathbf{d}_1$



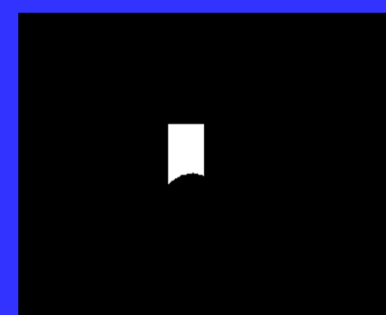
$\mathbf{M}_{pawn}$



$\mathbf{X}^J$

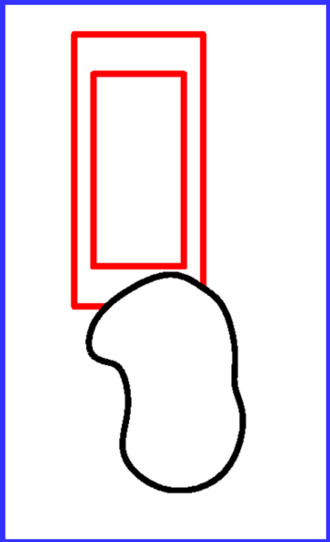


$\mathbf{d}_2$

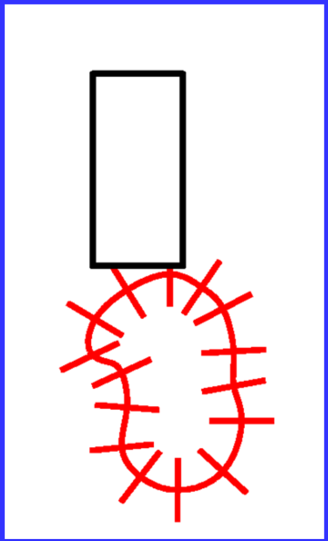


$\mathbf{M}_{pawn}$

JLF: Depth-independent interactions



Inhibitory frame



Edge search

JLF: Joint Image Likelihood

$$p^J(\mathbf{I} \mid \mathbf{X}^J) = \prod_{t_j \in H} p^J_{hregion}(\mathbf{I} \mid \mathbf{X}_j) \prod_{t_j \in T} p^J_{tregion}(\mathbf{I} \mid \mathbf{X}_j) \prod_{t_j \in S} p^J_{snake}(\mathbf{I} \mid \mathbf{X}_j)$$

Textured regions: Crossing planes

$\mathbf{I}_R$

PDAF

0 60 120 180

JLF

0 60 120 180