Visual Motion Analysis and Tracking
Part I

David J Fleet and Allan D Jepson

CIAR NCAP Summer School

July 12-16, 2005

Outline

Optical Flow and 2D Tracking:

- Optical flow estimation
  (iterative refinement, coarse to fine, robust estimation, mixture models)
- Motion-based tracking
  (EigenTracking, WSL, Feature Correspondence)

Model-Based Tracking:

- Bayesian filtering / smoothing
- Kalman Filter
- Particle filters
- Lessoned learned
- ...

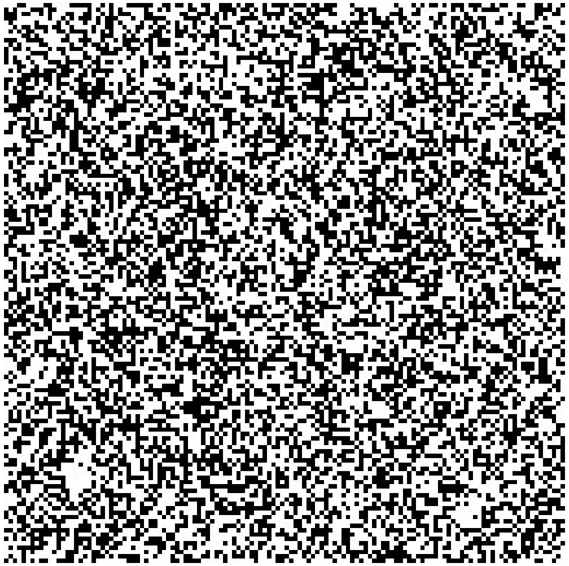
A Brief and Biased History of Image Motion Computation

Introduction

Flower garden sequence

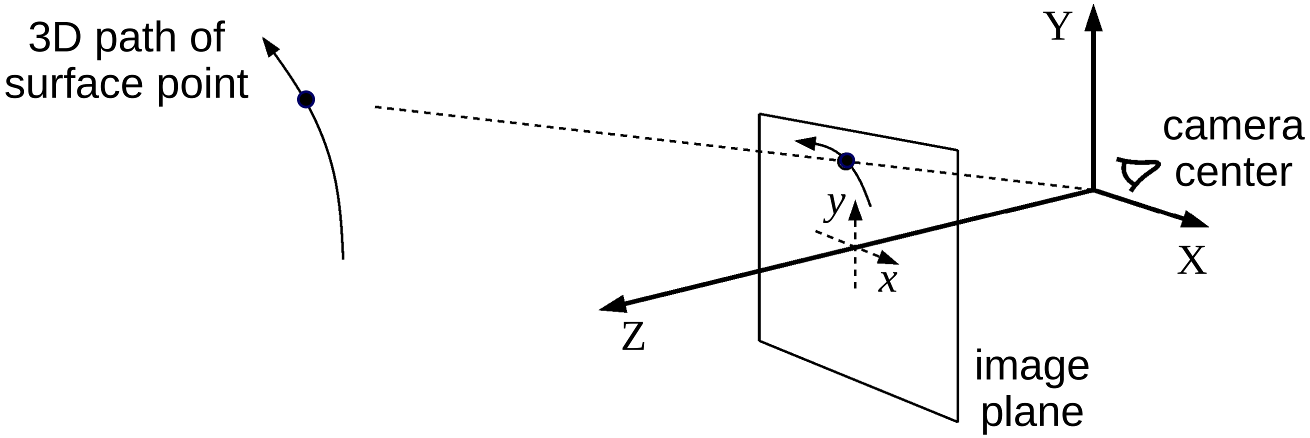
Bill's class

Introduction: Visual Perception



Motion defined form

Introduction

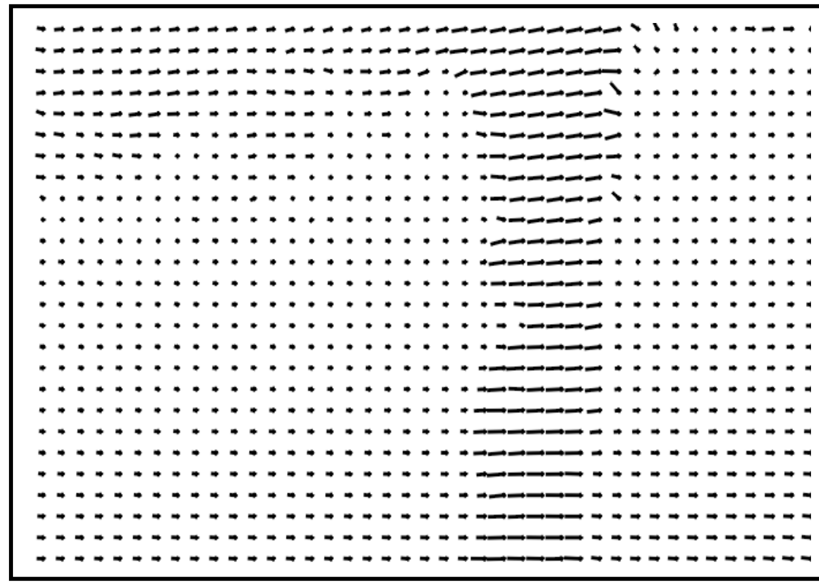


- 3D point $\vec{X}$ traverses a space-time path $\vec{X}(t)$.
- Projection onto the image plane produces a 2D path, the derivative of which is the 2D instantaneous velocity.
- 2D Motion Field: 2D velocities for all visible points.
- Optical Flow Field: Estimate of the 2D motion field.

Introduction

$$I(\mathbf{x}, t)$$

$$I(\mathbf{x}, t+1)$$

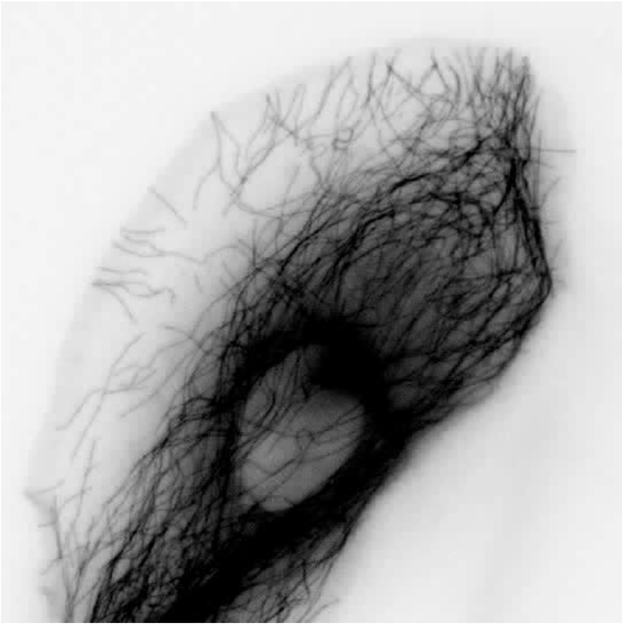


$$\mathbf{u}(\mathbf{x}) = (u(\mathbf{x}), v(\mathbf{x}))$$

Optical Flow

To estimate optical flow:

- Determine what image property to track
- Determine how to track it



Microtubule dynamics

Optical Flow: Brightness Conservation

Assume that we want to find the space-time paths along which image intensity is constant:

$$f(x(t), y(t), t) = c$$

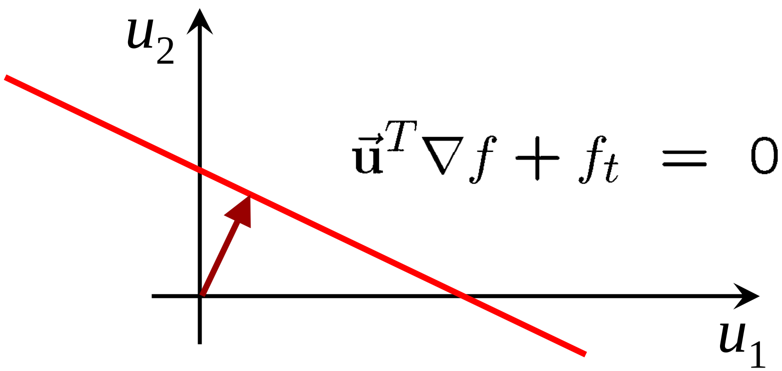
The total derivative must therefore be zero,

$$\begin{aligned} \frac{d}{dt} f(x(t), y(t), t) &= f_x u_1 + f_y u_2 + f_t \\ &= \vec{\mathbf{u}}^T \vec{\nabla} f + f_t = 0 \end{aligned}$$

This is called the Linearized Brightness Constancy Constraint (LBCC).

Optical Flow: Brightness Constancy Constraint Line

The LBCC provides one constraint in 2 unknowns. It defines a line in velocity space:



Only constrains the component of velocity in the gradient direction.

Aside: when the gradient magnitude is zero we get no constraint!

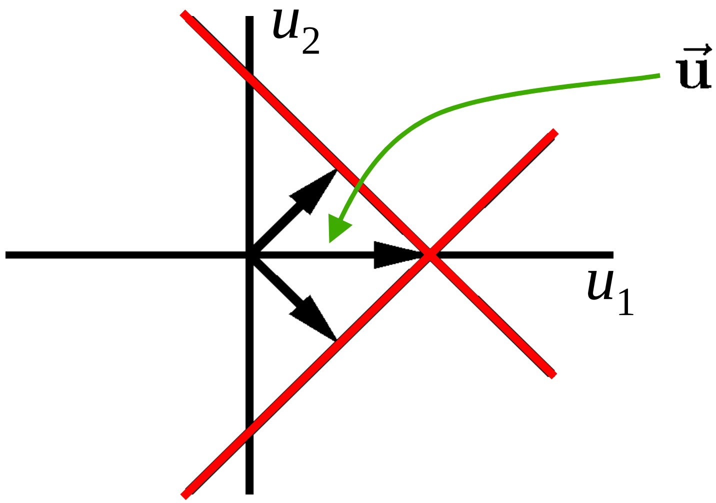
Optical Flow: Area-Based Regression

Assumption of Smoothness

2D motion is smooth in the local image neighborhood of the point at which we are estimating optical flow.

E.g., estimate the flow using two local gradient constraints:

$$\begin{bmatrix} f_x(x_1, y_1, t) & f_y(x_1, y_1, t) \\ f_x(x_2, y_2, t) & f_y(x_2, y_2, t) \end{bmatrix} \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} + \begin{bmatrix} f_t(x_1, y_1, t) \\ f_t(x_2, y_2, t) \end{bmatrix} = \vec{0},$$

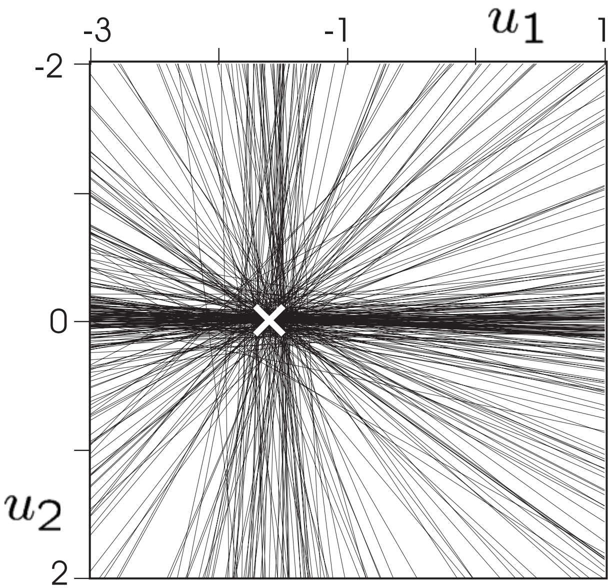


Optical Flow: Area-Based Regression

More generally, minimize the squared residual error in each constraint: (a linear least squares problem for (u_1, u_2)):

$$E(u_1, u_2) = \sum_{x,y} g(x, y)\,[u_1 f_x(x, y, t) + u_2 f_y(x, y, t) + f_t(x, y, t)]^2$$

where $g(x, y)$ is a low-pass window that helps give more weight to constraints at the center of the region. (Linear least squares.)



Optical Flow: Iterative Estimation

Issue 1: The least-squares estimate was based on an approximate objective function (the LBCC).

Sum of squared errors (SSE) objective function:

$$E(u_1, u_2) = \sum_{x,y} g(x,y)\,[f(x,y,t+\delta t) - f(x-u_1, y-u_2, t)]^2$$

Approximate objective function:

$$\widehat{E}(u_1, u_2) = \sum_{x,y} g(x,y)\,[u_1 f_x(x,y,t) + u_2 f_y(x,y,t) + f_t(x,y,t)]^2$$

Aside: These objective functions agree up to higher order terms

$$E(u_1, u_2) \;=\; (\delta t)^2 \widehat{E}(u_1, u_2) + hots$$

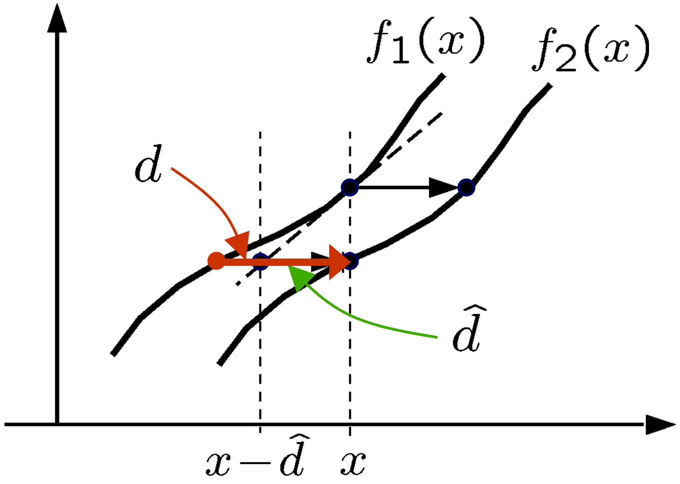
For small motions, $\widehat{E}(u_1, u_2)$ is a good approximation of $E(u_1, u_2)$

[Jepson & Black, "EigenTracking: Robust matching and tracking of articulated objects using a view-based representation." IJCV, 1998]

Optical Flow: Iterative Estimation

The errors resulting from this approximation are second-order in the magnitude of the displacement:

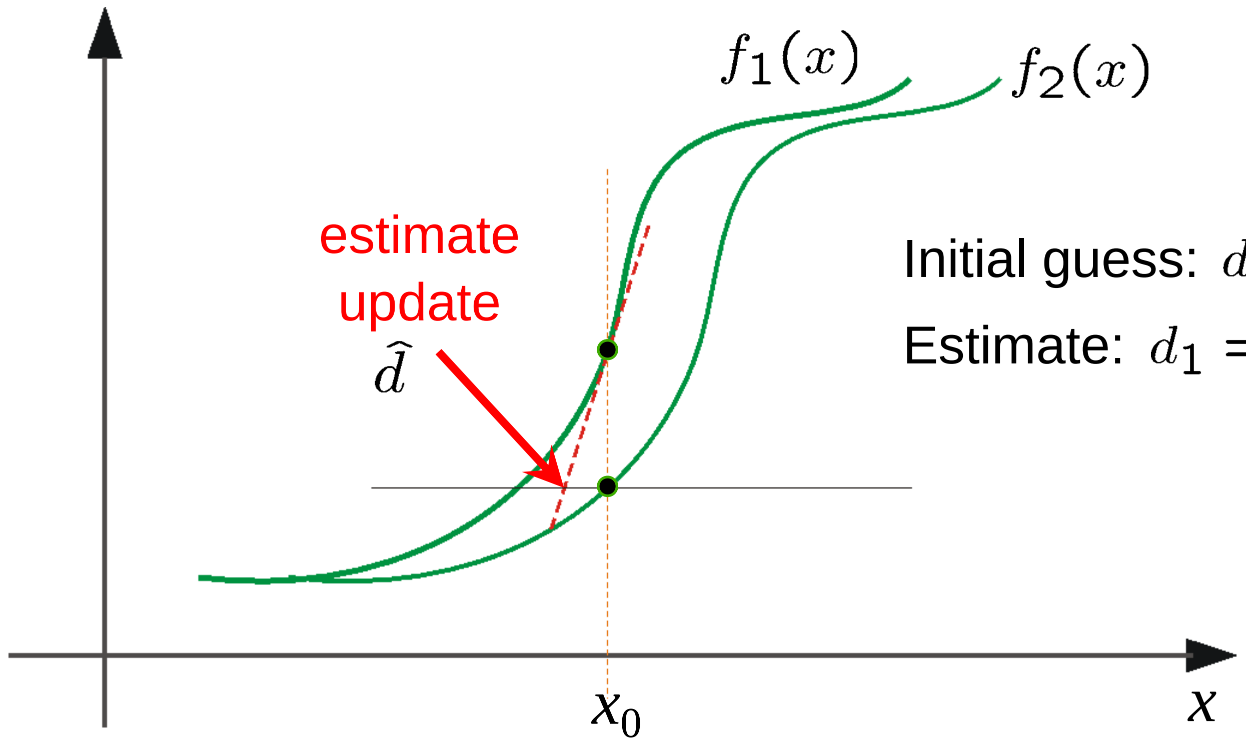
$$|\hat{d} - d| \;\leq\; \frac{d^2\,|f_1''(x)|}{2\,|f_1'(x)|}$$



So we can converge to a solution by iterating the estimation:

1) estimate the shift given the two images,
2) warp one image toward the other
3) repeat

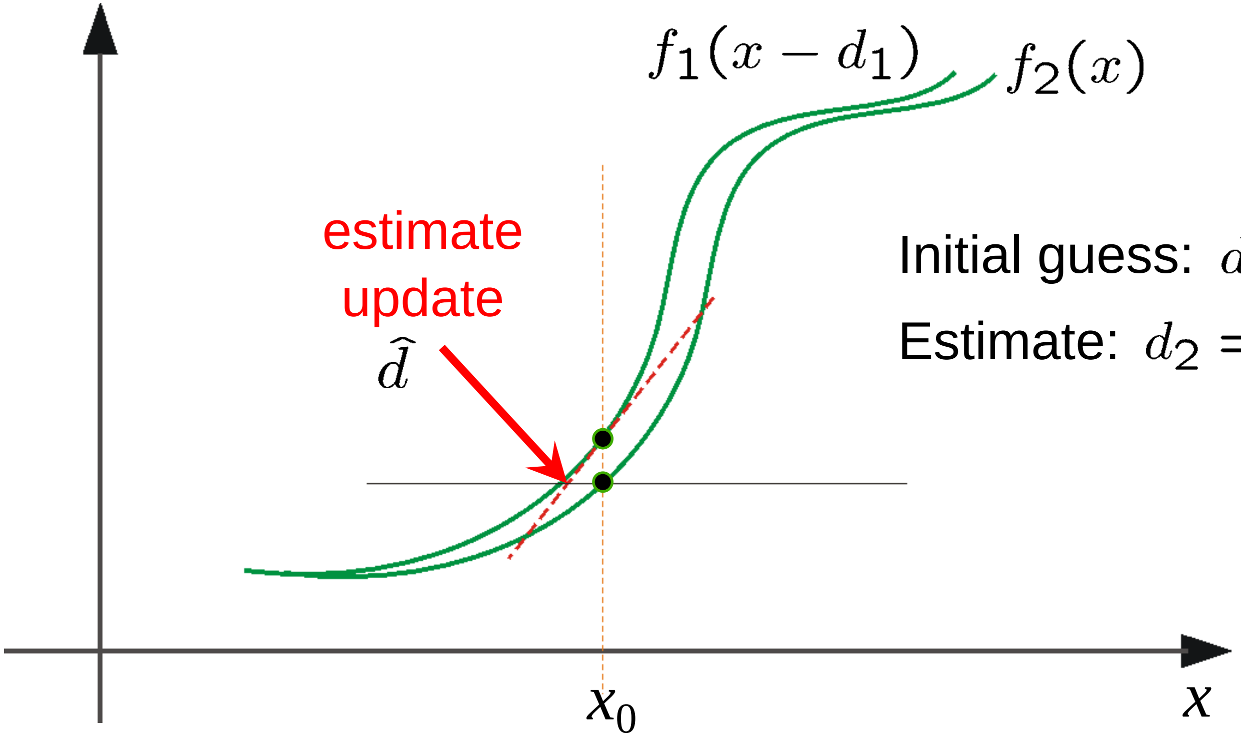
Optical Flow: Iterative Estimation



Initial guess: $d_0 = 0$

Estimate: $d_1 = d_0 + \hat{d}$

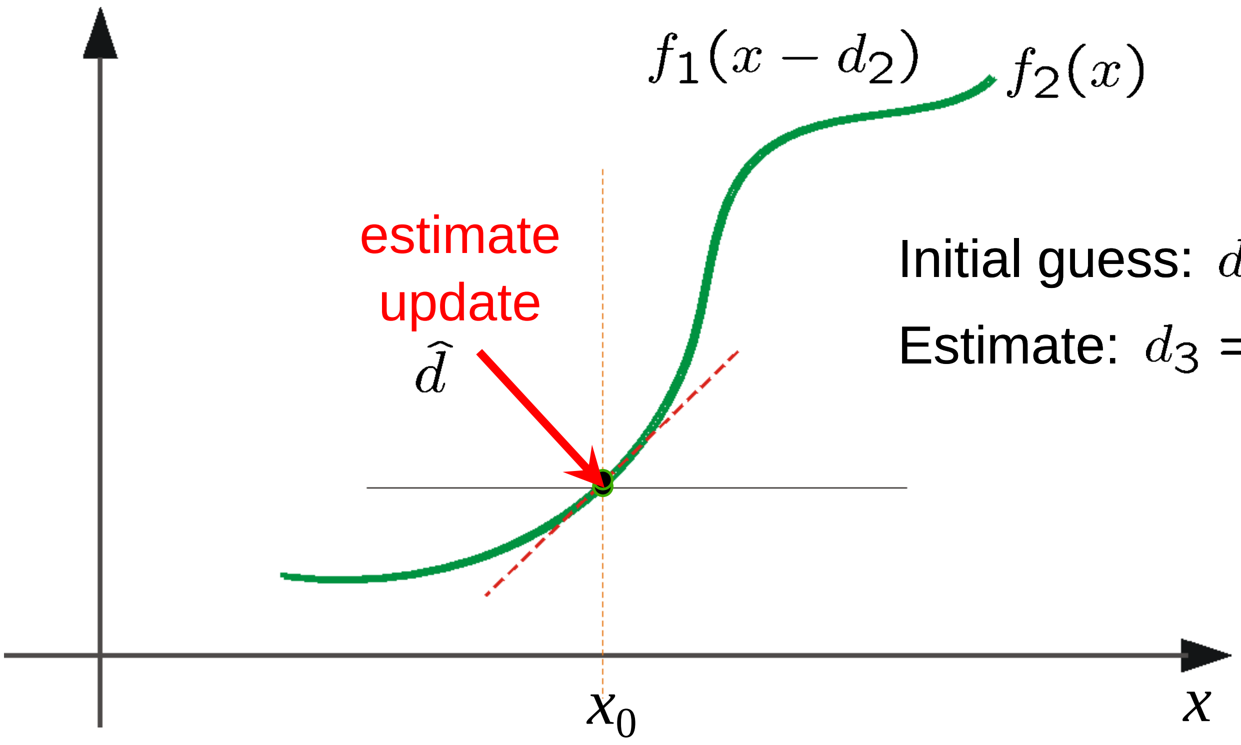
Optical Flow: Iterative Estimation



Initial guess: d_1

Estimate: $d_2 = d_1 + \hat{d}$

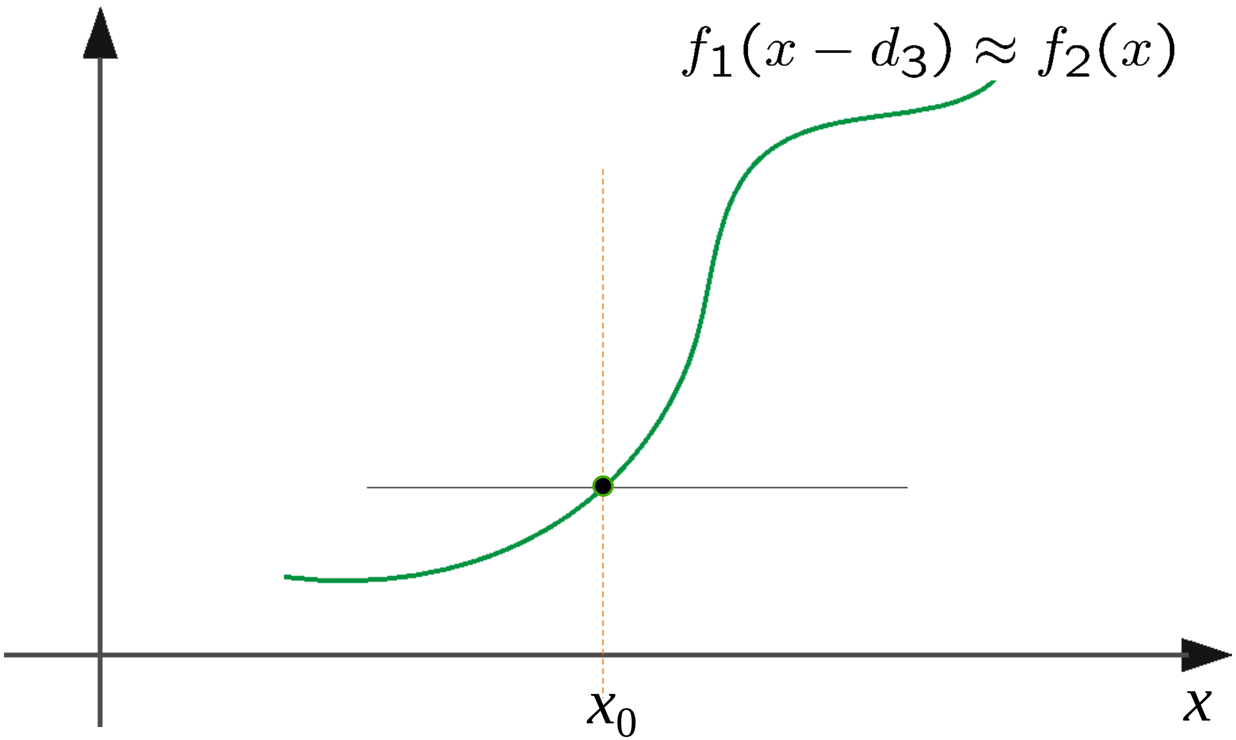
Optical Flow: Iterative Estimation



Initial guess: d_2

Estimate: $d_3 = d_2 + \hat{d}$

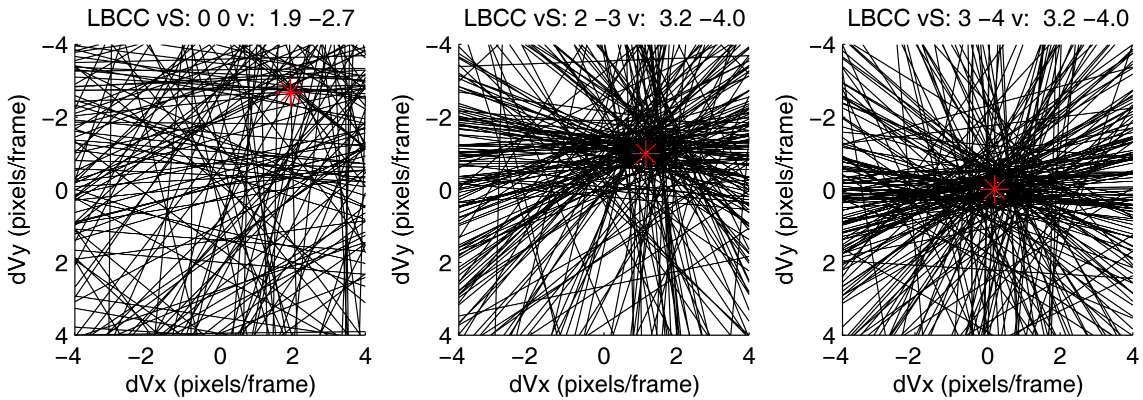
Optical Flow: Iterative Estimation

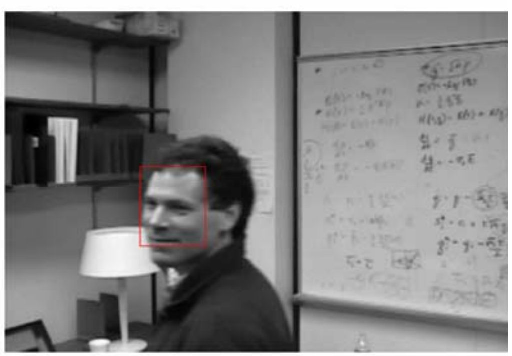


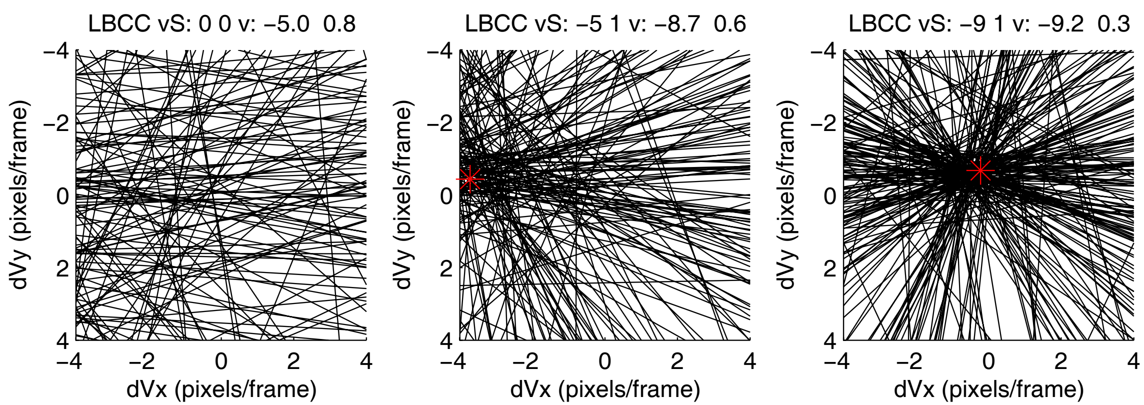
Aside: In 1D, this process is simply Newton's method applied to

$$F(d) \equiv f_1(x - d) - f_2(x) = 0.$$

Optical Flow: Iterative Estimation

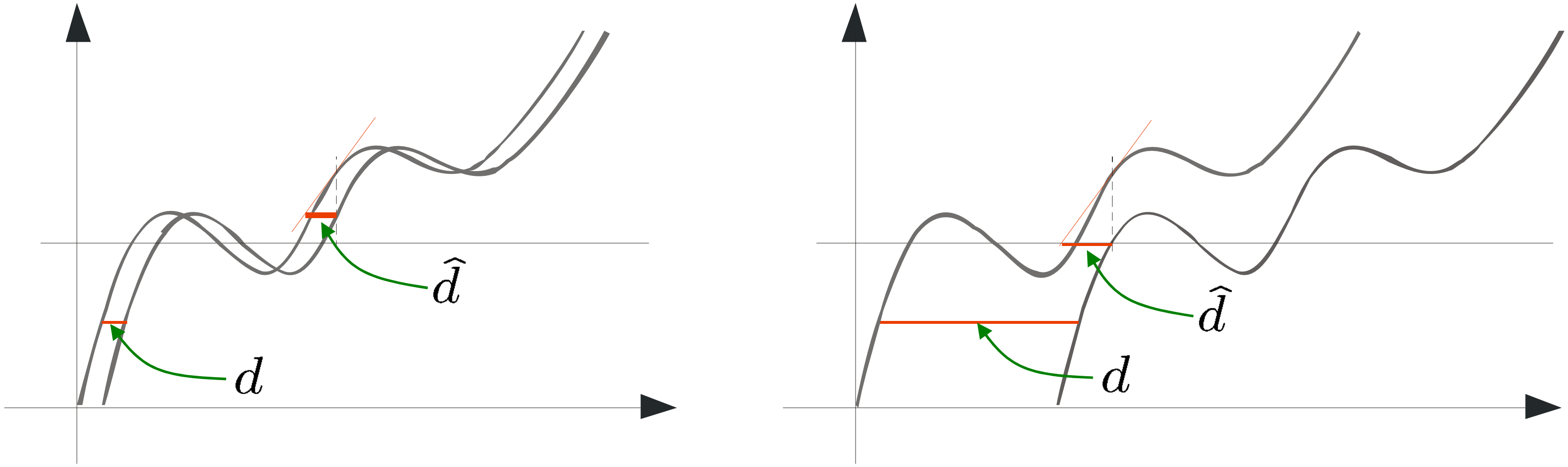






Optical Flow: Temporal Aliasing

Issue 2: Image sequences produced by most cameras can be temporally aliased. The displacement of image structure from one frame to the next can be large relative to that structure's spatial scale.



Small versus large displacement relative to signal's frequency content

Optical Flow: Temporal Aliasing

To avoid suffering from temporal aliasing we need a good prediction for the motion parameters at the current time (relative to the spatial scale of the local image structure).

Strategies for dealing with temporal aliasing:

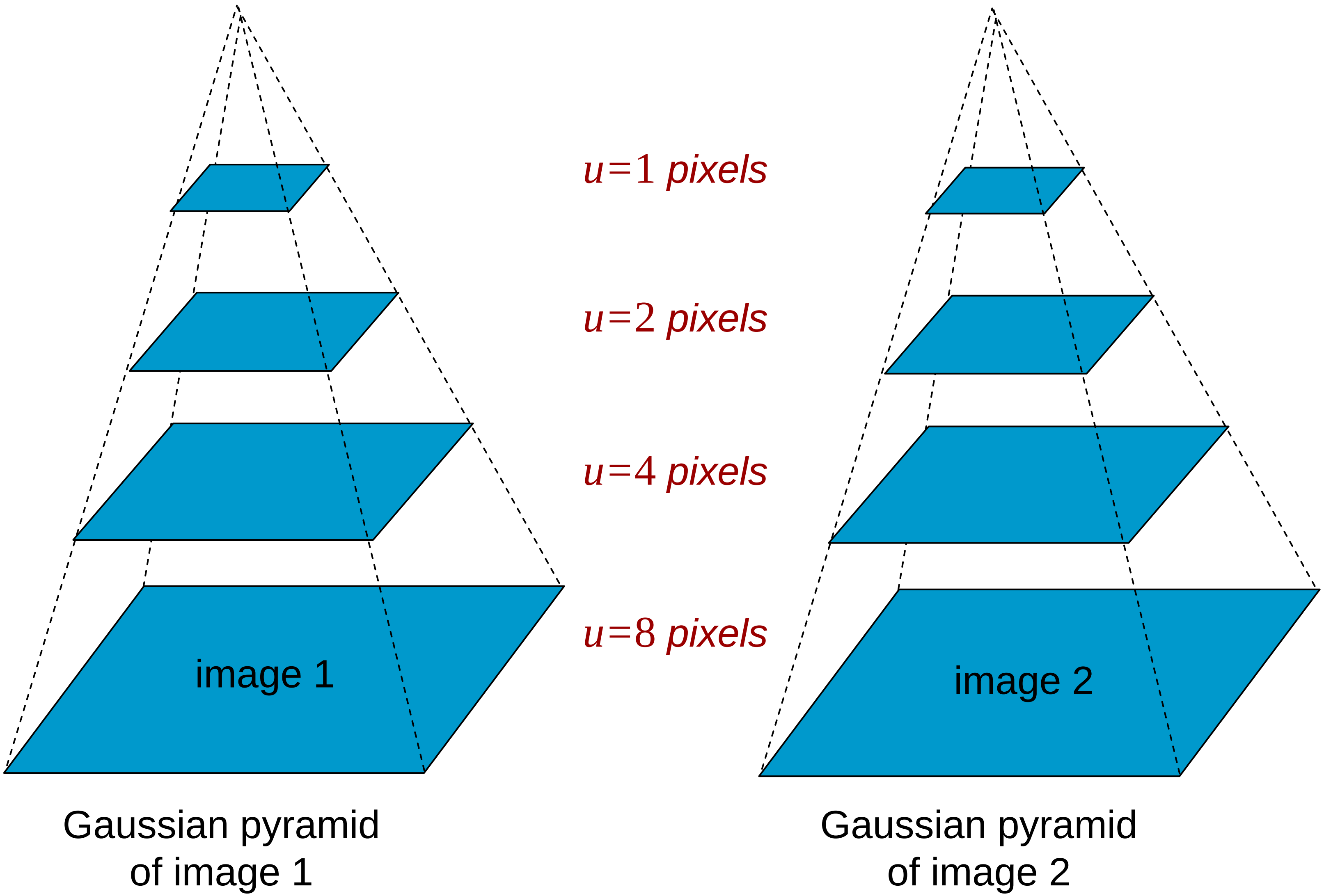
- Predict the motion from previous frames, i.e. use a model of the dynamics (e.g. no acceleration).

  See the Matlab tutorial, motionTutorial.m.

- Use coarse to fine processing in space.

- Use features from the current frame to do a rough initial alignment.

Optical Flow: Coarse-to-Fine Estimation



Optical Flow: Coarse-to-Fine Estimation

So we can converge to a solution by iterating the estimation, within a level, starting with the coarsest level first.

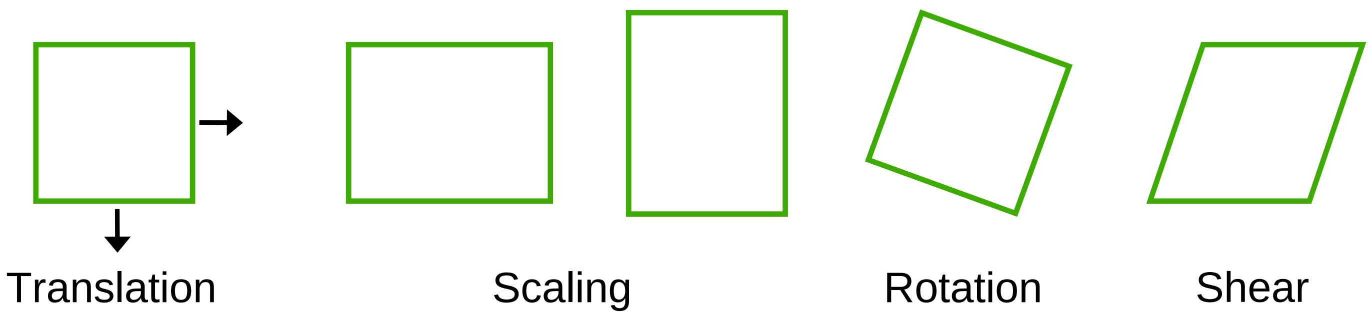
Given low-pass Gaussian pyramids for 2 images and an initial guess for the optical flow, $\vec{\mathbf{u}}_{n+1}$

- Starting at level k=n (coarsest level), warp images using initial guess, then apply warp-estimate iteration, until convergence. Let the cumulative flow estimate be $\vec{\mathbf{u}}_n$
- Warp level k=n-1 using $\vec{\mathbf{u}}_n$, then apply iterative estimation until convergence with flow estimate $\vec{\mathbf{u}}_{n-1}$
- …
- Warp level k=0 using $\vec{\mathbf{u}}_1$, then apply warp-estimate iteration, until convergence with final flow estimate $\vec{\mathbf{u}} = \vec{\mathbf{u}}_0$

Optical Flow: Affine Deformation Model

Issue 3: Assumption of constant optical flow within an image neighborhood is often a relatively poor assumption.

Affine models often provide more suitable model of the image deformation from one time to the next.



Affine flow for an image region centered at location $\vec{\mathbf{x}}_0$ is given by

$$\vec{\mathbf{u}}(\vec{\mathbf{x}}) = \begin{pmatrix} a_1 & a_2 \\ a_3 & a_4 \end{pmatrix} (\vec{\mathbf{x}} - \vec{\mathbf{x}}_0) + \begin{pmatrix} a_5 \\ a_6 \end{pmatrix}$$

Optical Flow: Affine Deformation Model

Consider affine flow about $\vec{\mathbf{x}}_0 = \vec{0}$:

$$\vec{\mathbf{u}}(x,y) \;=\; A(x,y)\,\vec{\mathbf{a}}$$

$$\vec{\mathbf{a}} = (a_1, a_2, \cdots, a_6)^T$$

$$A = \begin{pmatrix} x & y & 0 & 0 & 1 & 0 \\ 0 & 0 & x & y & 0 & 1 \end{pmatrix}$$

Gradient constraint equation then becomes:

$$\begin{aligned} 0 \;&=\; \vec{\mathbf{u}}(x,y)^T \nabla f(x,y,t) + f_t(x,y,t) \\ &=\; \vec{\mathbf{a}}^T \left(A(x,y)^T \nabla f(x,y,t)\right) + f_t(x,y,t) \end{aligned}$$

Least Squares solution for the unknown 6-vector minimizes:

$$\hat{E}(\vec{\mathbf{a}}) \;=\; \sum_{x,y} g(x,y)\,[\vec{\mathbf{a}}^T A^T \nabla f + f_t]^2$$

[Fleet & Jepson, "Component velocity from local phase information." IJCV 1990; Bergen et al., "Hierarchical model-based motion estimation." Proc ECCV, 1992]

Optical Flow: Similarity Deformation

Similarity transform models uniform scaling, rotation and translation:

$$\vec{\mathbf{u}}(\vec{\mathbf{x}}) = \alpha \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} (\vec{\mathbf{x}} - \vec{\mathbf{x}}_0) + \begin{pmatrix} d_1 \\ d_2 \end{pmatrix}$$

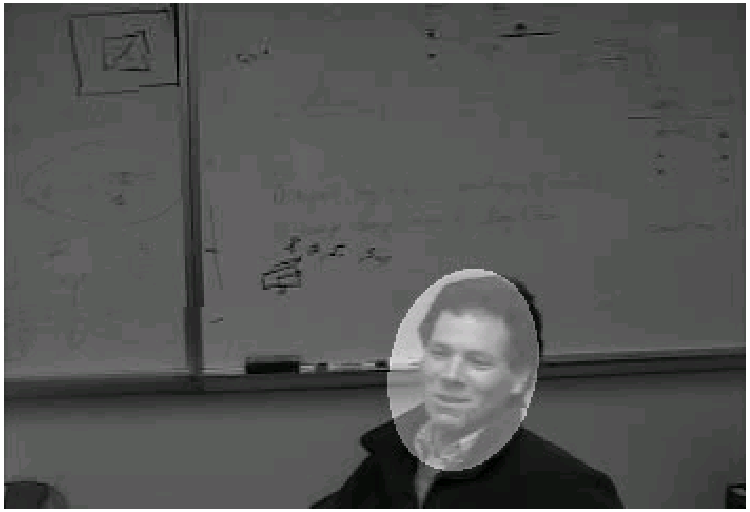
Similarity motion model about $\vec{\mathbf{x}}_0$:

$$\vec{\mathbf{u}}(x, y) = A(x, y)\,\vec{\mathbf{a}}$$

where

$$\vec{\mathbf{a}} = (\alpha\cos\theta,\, \alpha\sin\theta,\, d_1,\, d_2)^T$$

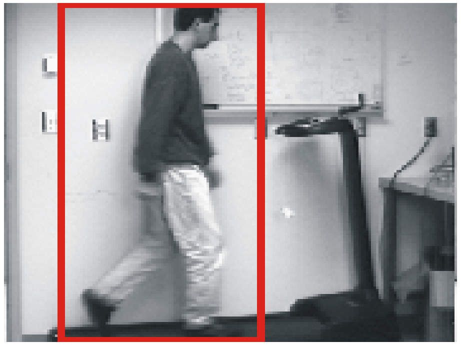
$$A = \begin{pmatrix} x - x_0 & -y + y_0 & 1 & 0 \\ y - y_0 & x - x_0 & 0 & 1 \end{pmatrix}$$



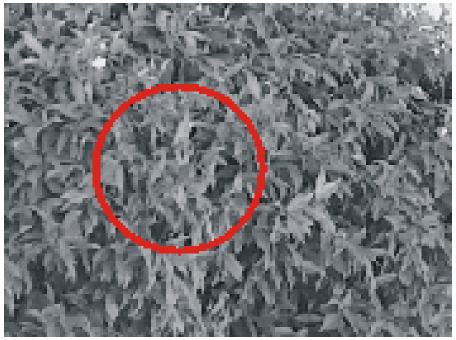
Optical Flow: Sub-Space Motion Models

Issue 4: What about optical flow that is not effectively described by a simple generic model such as affine motion or a homography?

motion boundaries



non-rigid motion

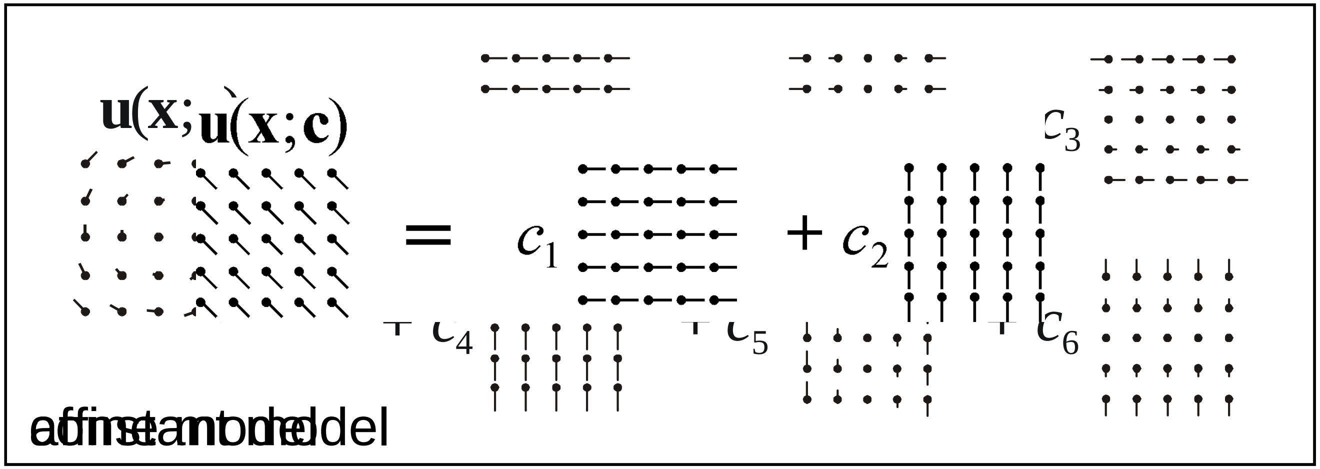


motion texture

Optical Flow: Sub-Space Motion Models

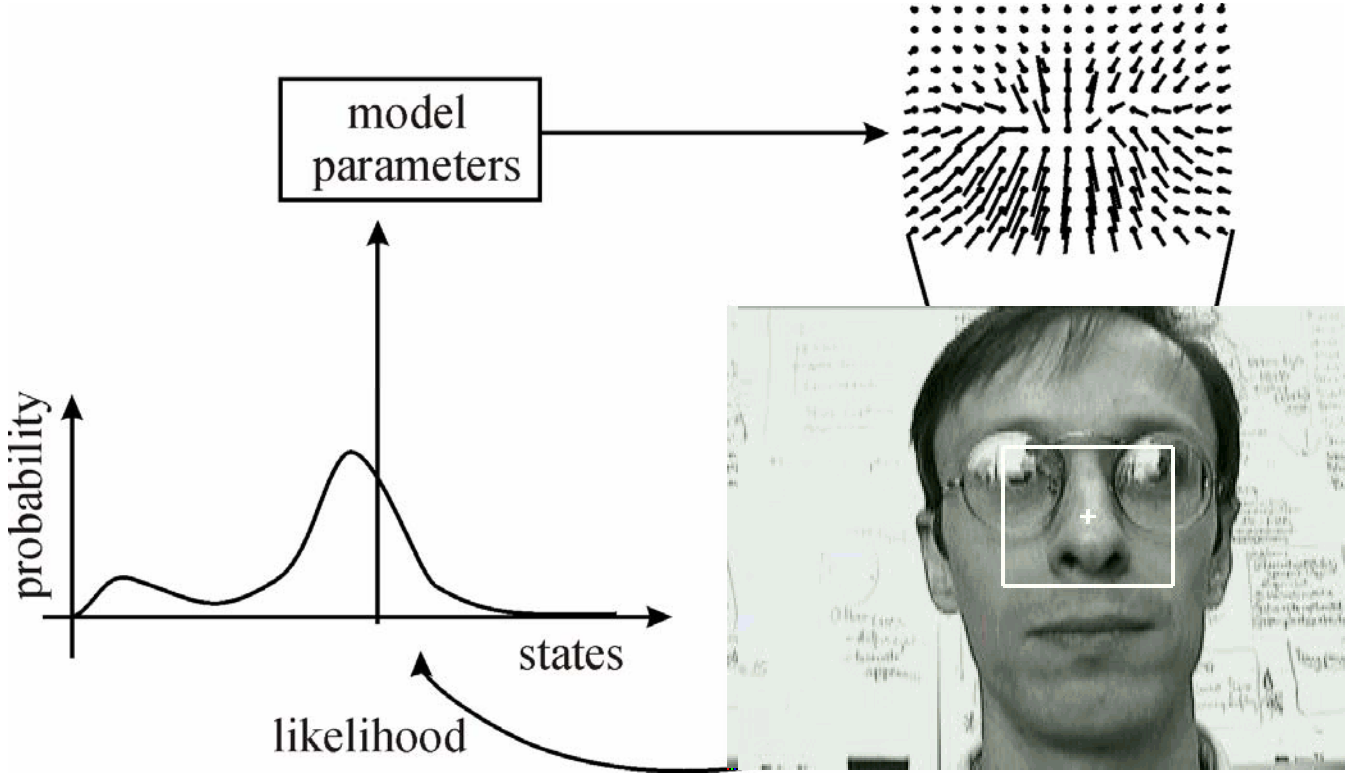
Express the deformation as a linear basis of flow fields, $\vec{\mathbf{b}}_1(\vec{\mathbf{x}}) \cdots \vec{\mathbf{b}}_n(\vec{\mathbf{x}})$ with coefficients $\vec{\mathbf{c}} = (c_1, \cdots, c_n)^T$

$$\vec{\mathbf{u}}(\vec{\mathbf{x}}; \vec{\mathbf{c}}) = \sum_j c_j \vec{\mathbf{b}}_j(\vec{\mathbf{x}})$$



Gradient constraint is linear in $\vec{\mathbf{u}}(\vec{\mathbf{x}}; \vec{\mathbf{c}})$, and $\vec{\mathbf{u}}(\vec{\mathbf{x}}; \vec{\mathbf{c}})$ is linear in the parameters of the motion model, $\vec{\mathbf{c}}$.

Optical Flow: Learned Motion Models

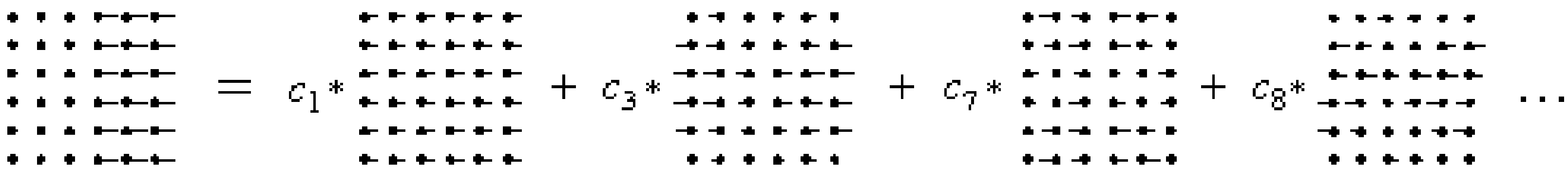


Detection of mouths and
simple word recognition

[Fleet et al, "Design and Use of linear models for image motion analysis", IJCV, 2000]

Optical Flow: Steerable Flow Fields

Linear parametric models can approximate motion discontinuities or arbitrary orientation in local neighborhoods (Fleet et al, 2000)

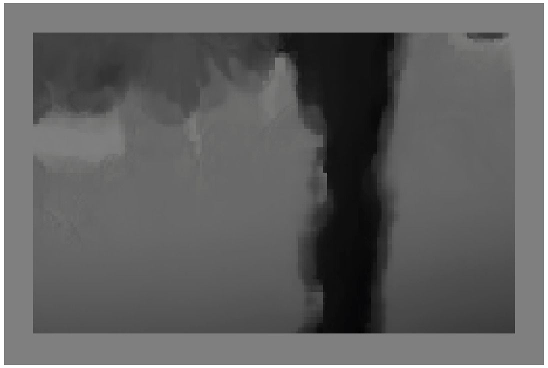


This flow field basis spans both smooth motion as well as an approximation to motion boundaries (e.g., at surface boundaries)

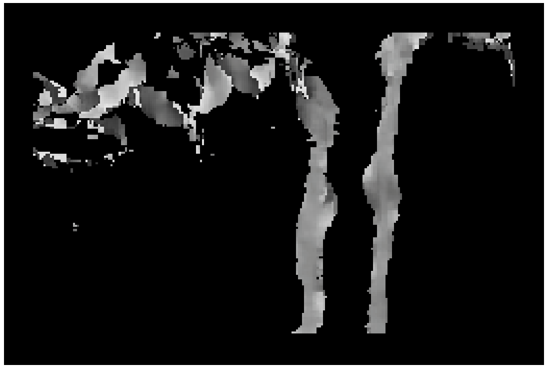
- So we can estimate both smooth flow and discontinuous flow with a single basis set.
- We can use the linear coefficients to detect locations at which flow is likely discontinuous, and the boundary orientation.

Low-Level Detector: Flower Garden Results

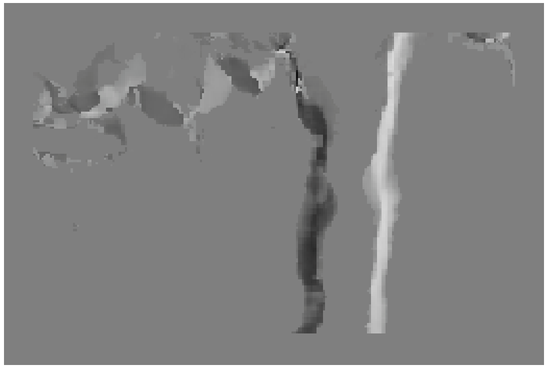
Flower garden sequence



Horizontal Velocity



Edge Orientation



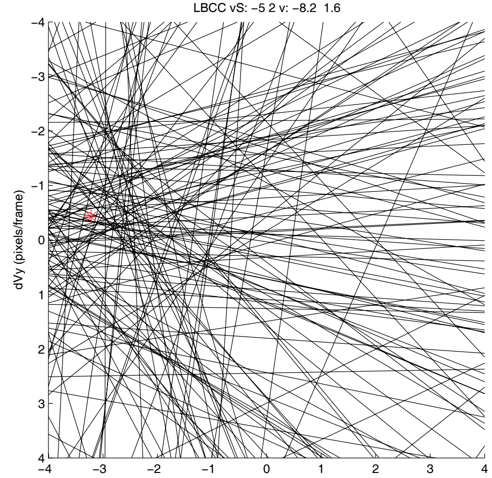
Velocity Difference

Optical Flow: Measurement Outliers

Issue 5: The assumptions of brightness constancy and smoothness are often violated. Noise distributions are often non-Gaussian, having heavier tails. Noise samples from the tails are called outliers.

Sources of outliers:

- temporal aliasing,
- specularities / highlights
- jpeg artifacts / interlacing / motion blur
- multiple motions (occlusion boundaries, transparency)

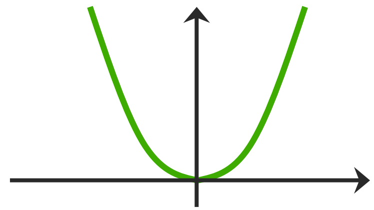


Optical Flow: Robust Estimation

Problem: Least-squares estimators penalize deviations between data & model with quadratic error (extremely sensitive to outliers)

error penalty function

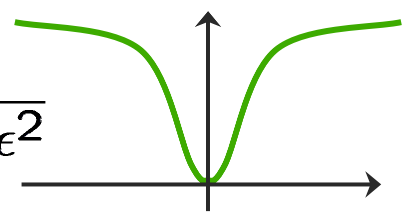
$$\rho(\epsilon) = \epsilon^2$$



Saturating error functions (e.g., Geman-McClure) help to reduce the influence of outlying measurements.

error penalty function

$$\rho(\epsilon; s) = \frac{\epsilon^2}{s + \epsilon^2}$$



[Black & Anandan, "The robust estimation of multiple models, parametric and piecewise smooth flow fields." CVIU 1996]

Optical Flow: Robust Estimation

Previous objective function:

$$E(\vec{\mathbf{u}}) = \sum_{x,y} g(\vec{\mathbf{x}})\,[f(\vec{\mathbf{x}}, t+1) - f(\vec{\mathbf{x}}-\vec{\mathbf{u}},\, t)]^2$$

Robust objective function:

$$E(\vec{\mathbf{u}}) = \sum_{x,y} g(\vec{\mathbf{x}})\;\rho(f(\vec{\mathbf{x}}, t+1) - f(\vec{\mathbf{x}}-\vec{\mathbf{u}},\, t); s)$$

Solution:

- Iteratively re-weighted least-squares is a common method. The inner loop only involves solving weighted LBCCs.
- Initial Guess? It's common to anneal the solution by starting with s large, and slowly decreasing it to the desired level.

Optical Flow: Probabilistic Flow Estimation

Issue 6:

- Confidence for least-squares optical flow estimates?
- How can we incorporate prior information about the flow?
- When the normal matrix is almost singular, what do we know?
- How do we take noise into account in the formulation?
- **...**

We can reformulate optical flow estimation in terms of probability theory so we can address these issues.

[Simoncelli et al., "Probability distributions of optical flow." Proc CVPR, 1991]

Optical Flow: Probabilistic Flow Estimation

Let's assume, for example, that spatial derivatives are accurate, but temporal derivative measurements have additive Gaussian noise:

$$\tilde{f}_t(x,y,t) \;=\; f_t(x,y,t) + \eta(x,y,t)$$

where $\eta(x,y,t)$ has a Gaussian density with mean zero and variance σ^2, i.e.

$$\vec{\mathbf{u}}^T \nabla f + \tilde{f}_t \;\sim\; N(0,\sigma^2)$$

Therefore, if we know, $\vec{\mathbf{u}}$, the likelihood of observing $\nabla f, \tilde{f}_t$

$$p(\nabla f, \tilde{f}_t \,|\, \vec{\mathbf{u}}) \;=\; \frac{1}{\sqrt{2\pi}\sigma} \exp\left(\frac{-(\vec{\mathbf{u}}^T \nabla f + \tilde{f}_t)^2}{2\sigma^2} \right)$$

Likelihood

In this case, maximum likelihood fitting of $\vec{\mathbf{u}}$ is identical to our previous least squares objective function (LBCC).

Optical Flow: Probabilistic Flow Estimation

Bayesian Estimation: Use *prior* belief about flow, $p(\vec{\mathbf{u}})$, we can use Bayes' rule to express the *posterior* flow distribution:

$$p(\vec{\mathbf{u}} \,|\, \{\nabla f, f_t\}_{\vec{\mathbf{x}}}) \;=\; \frac{p(\{\nabla f, f_t\}_{\vec{\mathbf{x}}} \,|\, \vec{\mathbf{u}})\, p(\vec{\mathbf{u}})}{p(\{\nabla f, f_t\}_{\vec{\mathbf{x}}})}$$

E.g., let's express our prior belief that flow is usually small with a Gaussian density:

$$p(\vec{\mathbf{u}}) \;=\; \frac{1}{\sqrt{2\pi}\sigma} \exp\left(\frac{-\vec{\mathbf{u}}^T\vec{\mathbf{u}}}{2\sigma_u^2}\right)$$

Maximum a posteriori (MAP) estimate for a constant flow model then becomes:

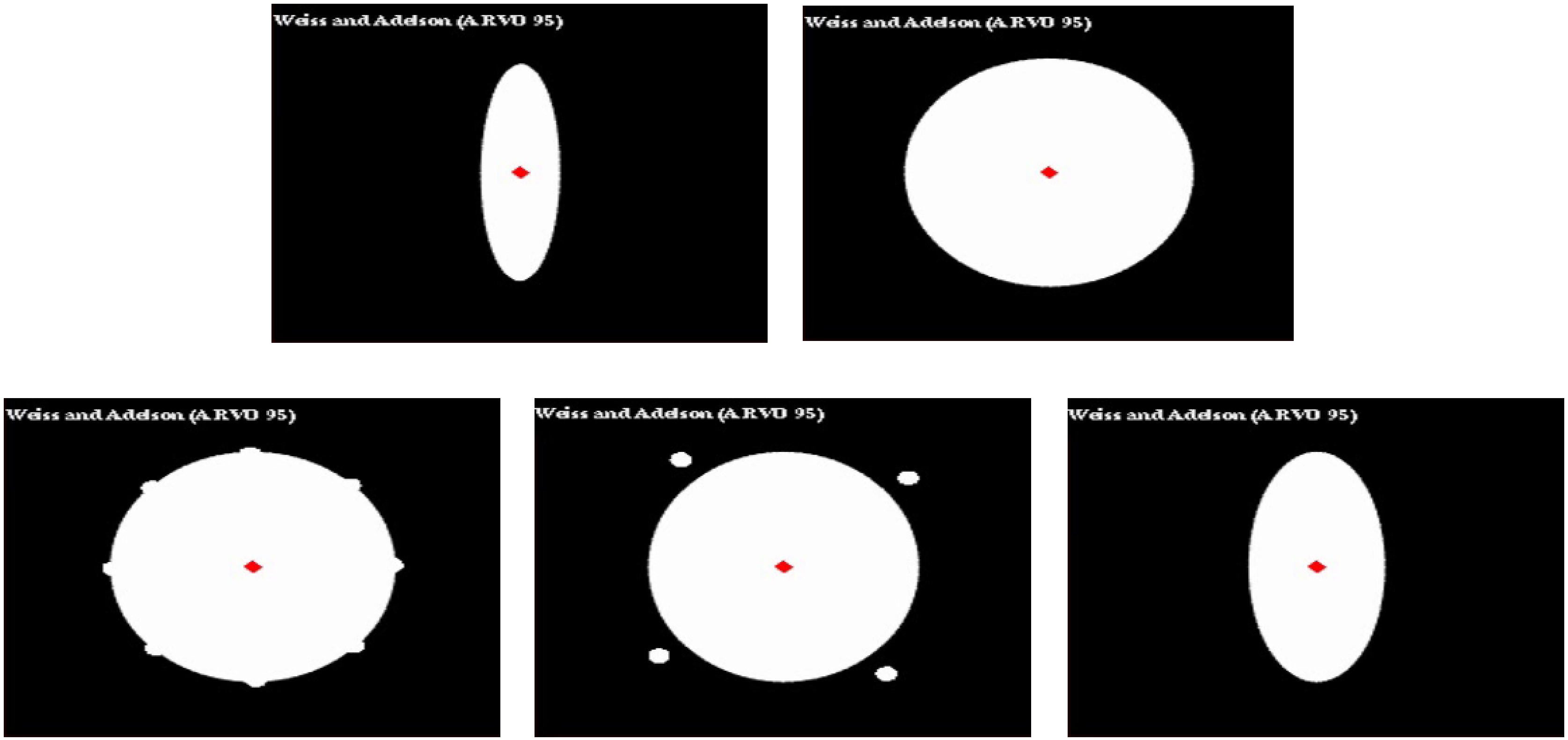
$$\hat{\mathbf{a}} = \left(M + \frac{1}{\sigma_u^2}\mathbf{I}\right)^{-1} \vec{\mathbf{b}}$$

2×2 identity

$$M = \frac{1}{\sigma^2}\sum_{x,y} \nabla f \, \nabla f^T$$

$$\vec{\mathbf{b}} = -\frac{1}{\sigma^2}\sum_{x,y} \nabla f \, f_t$$

Slow and Smooth: Rigid vs Non-rigid Motion



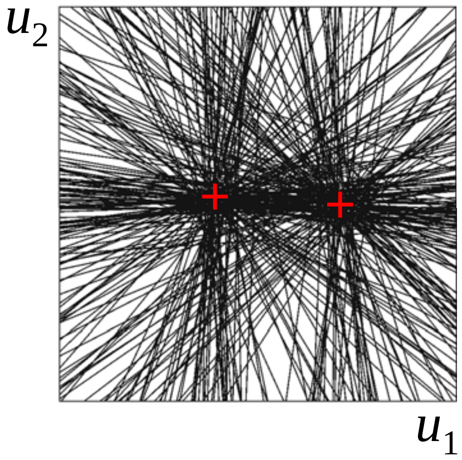
(Weiss & Adelson, 95)

Optical Flow: Mixture Models

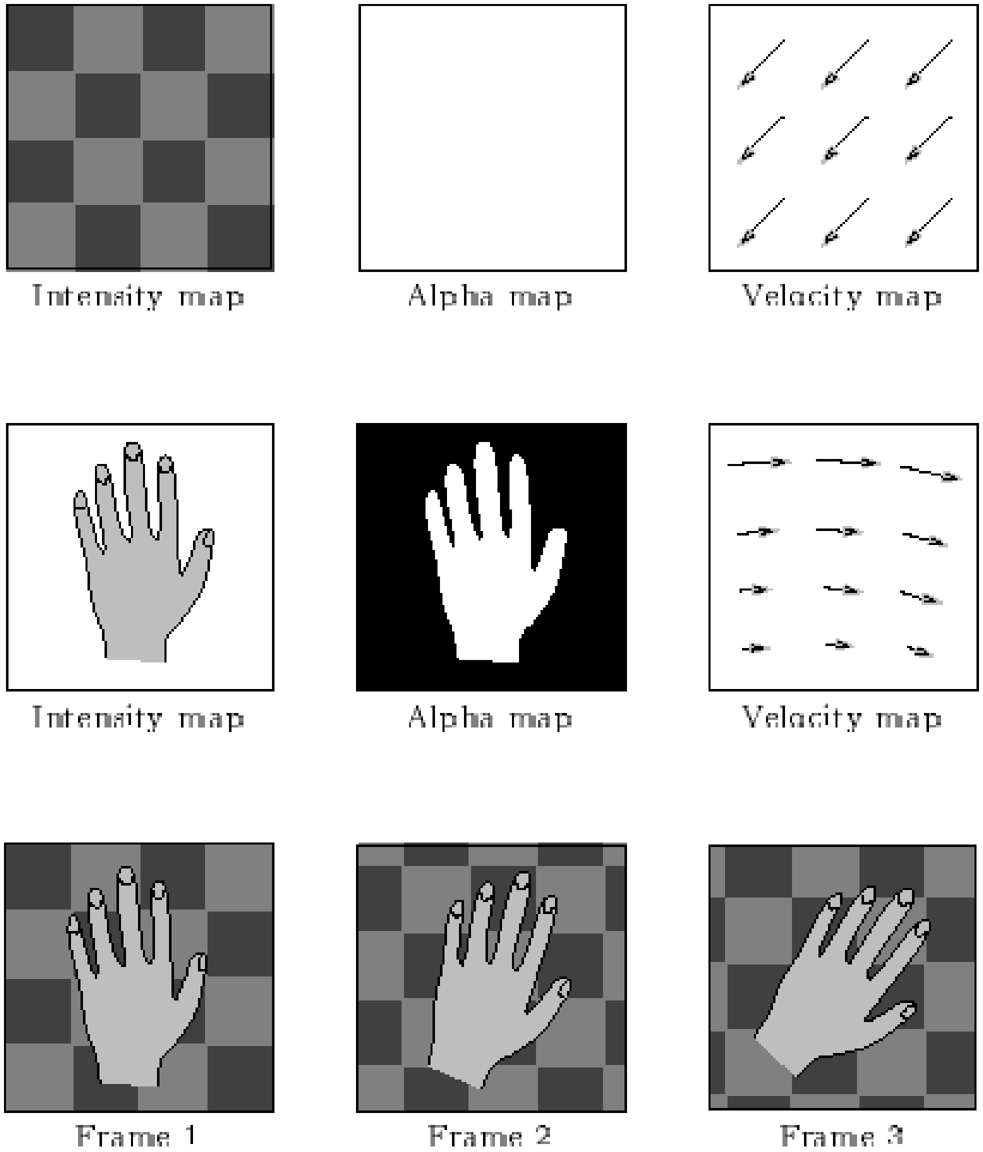
Issue 7: Structured outliers arise from multiple motions (occlusion boundaries, transparency, reflections).

Structured outliers can cause significant estimator bias away from the true solution, even when robust techniques are used (primarily in the case of nearby clusters).

velocity space



Optical Flow: Layered Motion Models



Wang and Adelson, '93

Optical Flow: Mixture Models for Layered Motion

General Idea: The motion constraint at each pixel arises from a probabilistic mixture of several simple layers.

Details: Each individual layer consists of a parametric motion model (e.g. affine) along with a Gaussian likelihood function:

$$p_n(\vec{c}_k \,|\, \vec{x}_k, \vec{a}_n) \;=\; G(\vec{\nabla} f(\vec{x}_k) \cdot \vec{u} + f_t(\vec{x}_k);\, \sigma_v)$$

Here $\vec{a}_k$ is the vector of parameters for the k^{th} motion and

$$\vec{c}_k \;=\; (\vec{\nabla} f(\vec{x}_k, t), f_t(\vec{x}_k, t)).$$

The probabilistic mixture model *M* is then

$$p(\vec{c}_k \,|\, \vec{x}_k, M) \;=\; m_0\, p_0(\vec{c}_k) + \sum_{n=1}^{N} m_n\, p_n(\vec{c}_k \,|\, \vec{x}_k, \vec{a}_n)\ .$$

Here the mixing coefficients $m_n \geq 0$ satisfy $\sum_{n=0}^{N} m_n = 1$ and $p_0(\vec{c}_k)$ is a broad density modeling outliers.

[Jepson & Black, "Mixture models for optical flow computation." Proc CVPR 1993]

Optical Flow: Mixture Models for Layered Motion

Maximum Likelihood: Assuming the observations are IID samples from the mixture model M, we fit the parameters in M by maximizing the log likelihood:

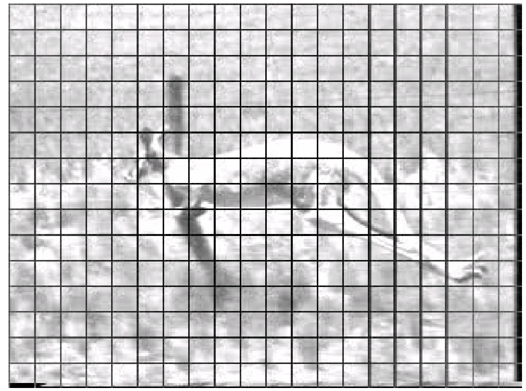
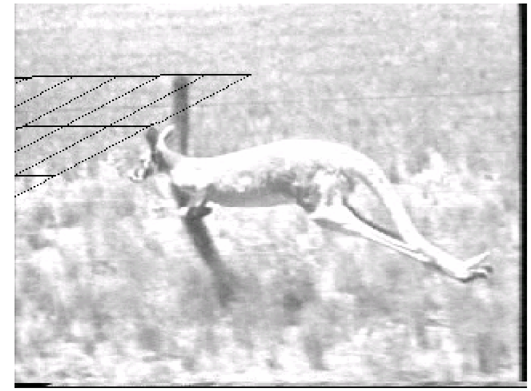
$$L(M) = \sum_k \log p(\vec{c}(\vec{x}_k)|M)$$

Expectation-Maximization Algorithm:

- **E-step**: Given the current model, compute the ownership probabilities which (softly) associate each constraint to each of the layers.
- **M-step**: Given these ownership probabilities, compute maximum likelihood estimates for the mixing proportions m_n, and for the motion parameters $\vec{a}_k$ of each layer.

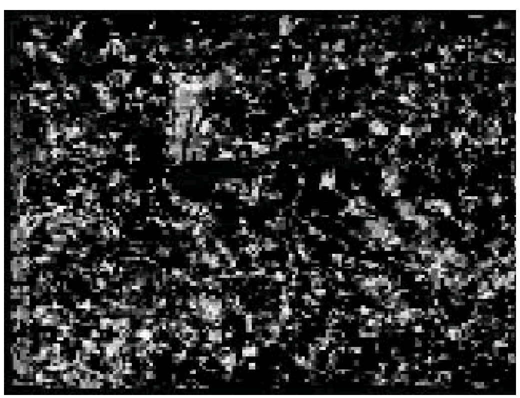
Iterate these two steps until convergence.

Optical Flow: Mixture Models for Layered Motion

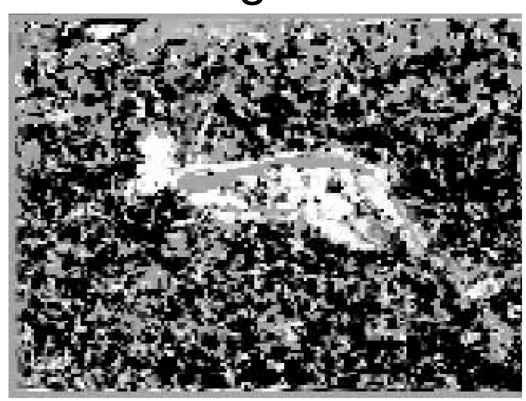




Layer Ownerships

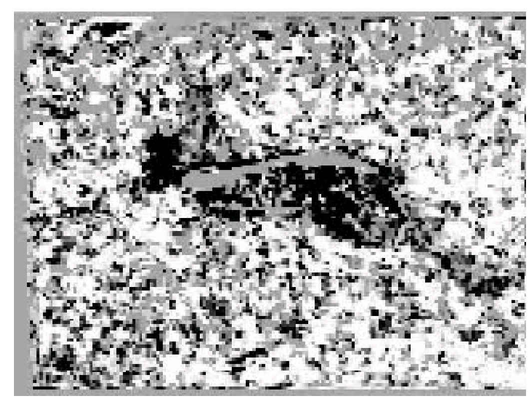
Outliers



Foreground



Background



2D Tracking

Essentially the same image matching algorithms apply to tracking.

For example, suppose we have an appearance model for an object to be tracked. For example,

$$B(\vec{x}, \vec{c}) = \sum_k B_k(\vec{x}) c_k$$

And we seek image warp parameters $\vec{a}_t$ and appearance parameters $\vec{c}_t$ such that

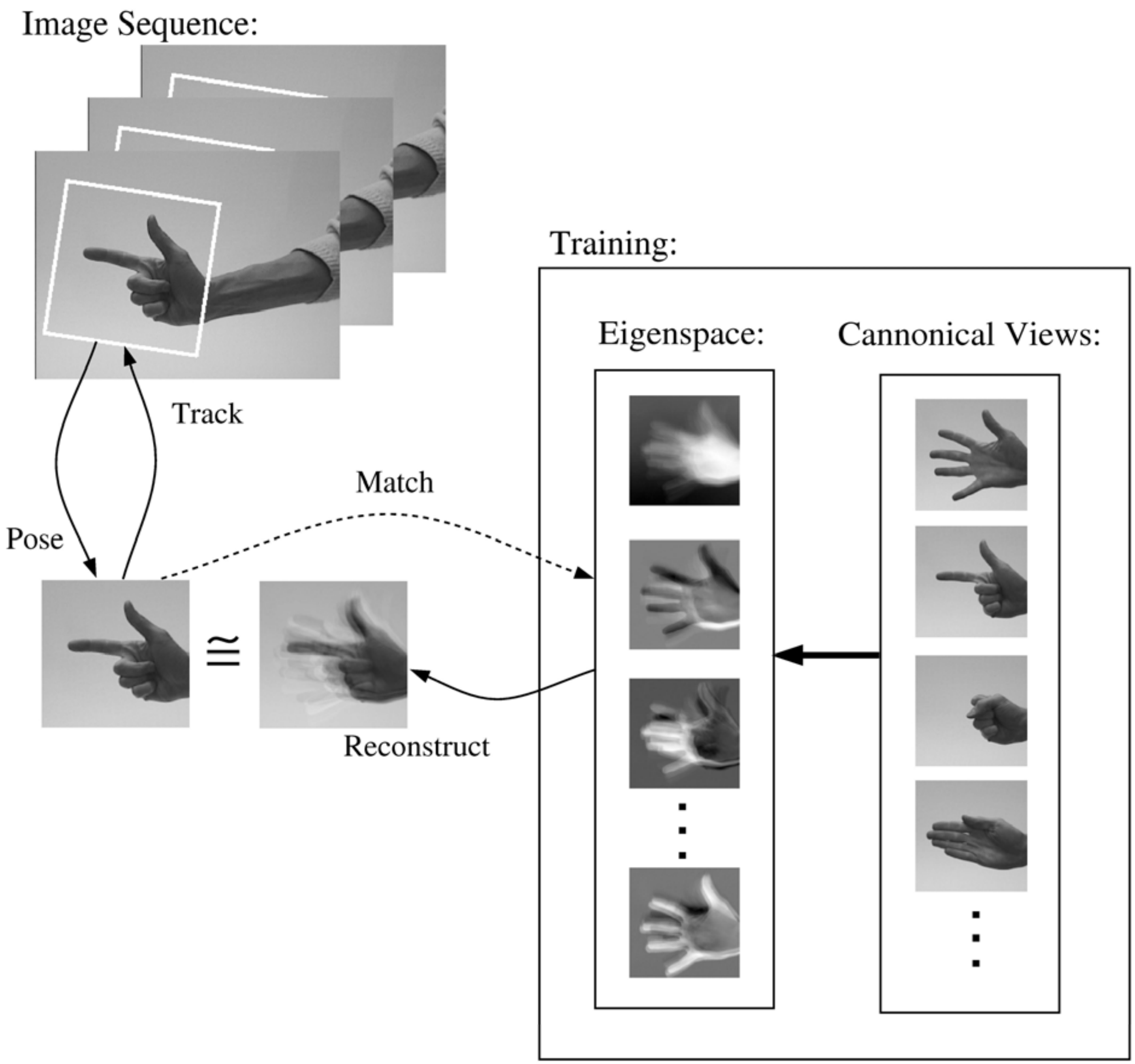
$$f(\vec{W}(\vec{x}, \vec{a}_t), t) \approx B(\vec{x}, \vec{c}_t)$$

The natural objective function is then

$$E(\vec{a}_t, \vec{c}_t) = \sum_{\vec{x}} \rho(f(\vec{W}(\vec{x}, \vec{a}_t), t) - B(\vec{x}, \vec{c}_t))$$

[Jepson & Black, "EigenTracking: Robust matching and tracking of articulated objects using a view-based representation." IJCV, 1998]

Eigen-Tracking



Online Appearance Models for Visual Tracking

Adaptively model image appearance to identify and track stable image properties.

Appearance Dynamics:

- slow/smooth deformation (3D motion, non-rigidity)
- abrupt changes (hair, glasses, smiling)
- outliers (occlusions, noise)

[Jepson, Fleet & El-Maraghi, "Robust online appearance models for visual tracking." IEEE Trans PAMI, 2003]

Greg’s Face

WSL Predictive Model

Predictive components of appearance for a single stream of observed data d_t

- Wandering process (first-order random walk):

$$p_w(d_t \mid d_{t-1}) = \mathcal{N}(d_t - d_{t-1}, \sigma_w^2)$$

- Stable process (given stable observation history):

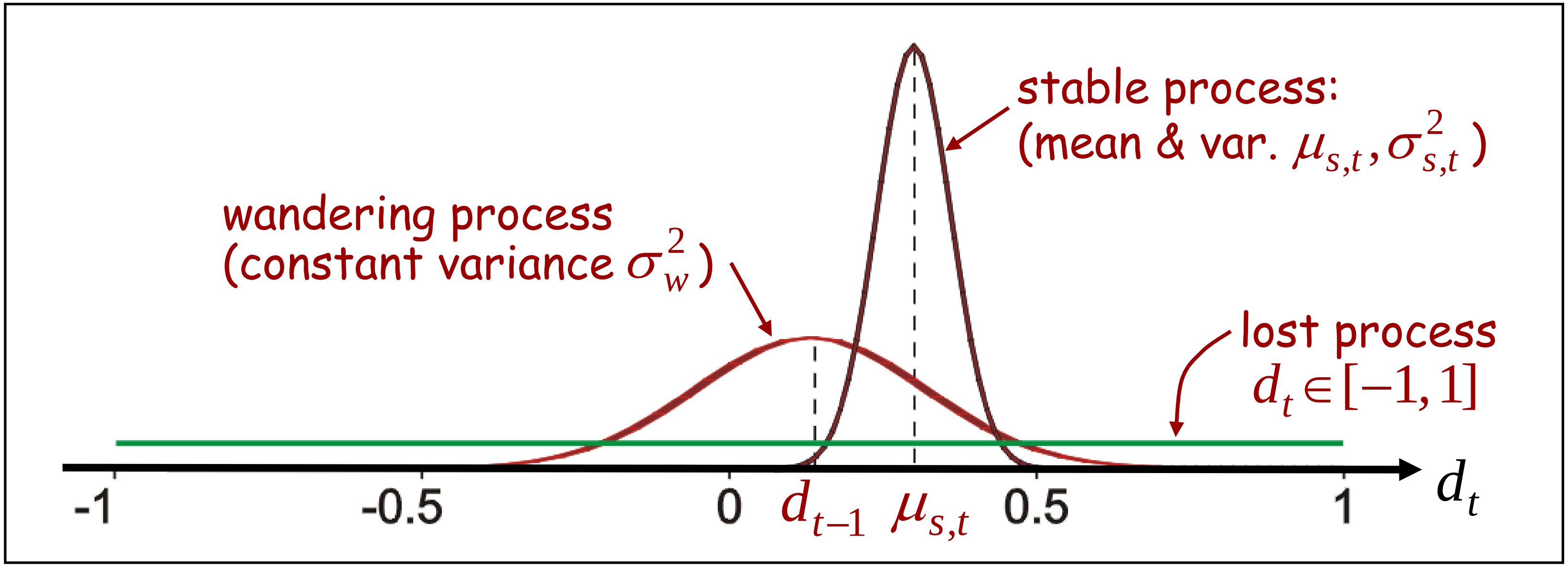
$$p_s(d_t \mid \mu_s, \sigma_s^2) = \mathcal{N}(d_t - \mu_s, \sigma_s^2)$$

- Lost process (outlier data):

$$p_l(d_t) = \mathcal{U}(d_t)$$

Parameters estimated recursively on-line.

Predictive Mixture Model



Mixture model for prediction density:

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

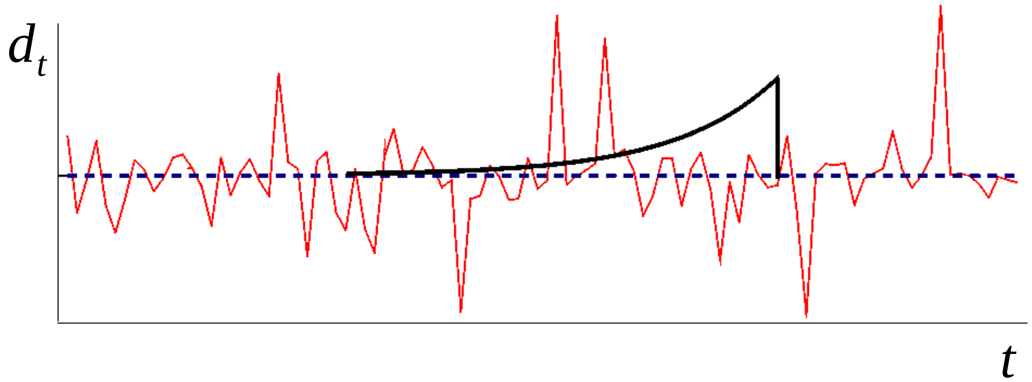
stable parameters
$\mathbf{q}_t = (\mu_{s,t}, \sigma^2_{s,t})$

mixing probabilities
$\mathbf{m}_t = (m_s, m_w, m_l)$

Local Parameter Estimation

Observed Data:

$$\mathbf{d}_t = \{d_k\}_{k=1}^{t}$$



Temporal support:

$$S_t(k) = \alpha\, e^{-(t-k)/\tau} \quad \text{where} \quad \alpha = 1 - e^{-1/\tau}$$

Temporally windowed, data log likelihood:

$$\log p(\mathbf{d}_t \mid \mathbf{q}_t, \mathbf{m}_t, \mathbf{d}_{t-1}) = \sum_{k=t}^{-\infty} S_t(k) \log p(d_k \mid \mathbf{q}_t, \mathbf{m}_t, d_{k-1})$$

Goal: Find model parameters that maximize log likelihood

Parameter Estimation via EM

E-Step: computer data ownerships

$$o_{j,t}(d_k) = \frac{m_{j,t}\, p_j(d_k \mid \mathbf{q}_t, d_{k-1})}{p(d_k \mid \mathbf{q}_t, \mathbf{m}_t, d_{k-1})} \qquad k \le t,\;\; j \in \{s, w, l\}$$

M-Step: update weighted data moments

$$M_{j,t}^{(i)} = \sum_{k=t}^{-\infty} S_t(k)\, o_{j,t}(d_k)\, d_k^{\,i} \qquad j \in \{s, w, l\}$$

update mixing probabilities (0^{th} data moments)

$$m_{j,t} = M_{j,t}^{(0)} \qquad j \in \{s, w, l\}$$

updated mean and variance of stable process

$$\mu_{s,t} = \frac{M_{s,t}^{(1)}}{M_{s,t}^{(0)}} \qquad \sigma_{s,t}^2 = \frac{M_{s,t}^{(2)}}{M_{s,t}^{(0)}} - \mu_{s,t}^2$$

On-Line (Approximate) EM

One E-Step: approximate ownerships

assume $o_{j,t}(d_k) \approx o_{j,k}(d_k) \qquad k \le t,\ \ j \in \{s, w, l\}$

One M-Step: recursive update of weighted data moments

$$\hat{M}^{(i)}_{j,t} = \sum_{k=t}^{-\infty} S_t(k)\, o_{j,k}(d_k)\, d_k^{\ i}$$

$$= \alpha\, o_{j,t}(d_t)\, d_t^{\ i} + (1-\alpha)\, \hat{M}^{(i)}_{j,t-1}$$

With 'stable' data, the model parameters vary slowly:

⇨ ownerships also change slowly, so one-step updates at each frame produce similar updates to EM iterations.

Initialization and Restarts

Problem: Online, nonlinear optimization requires initial guesses to avoid getting trapped at local extrema.

At frame 0, and when the stable component makes poor predictions (e.g., $m_{s,t} < 0.1$), we re-start the model with the mean at the current data.

Estimation Demo

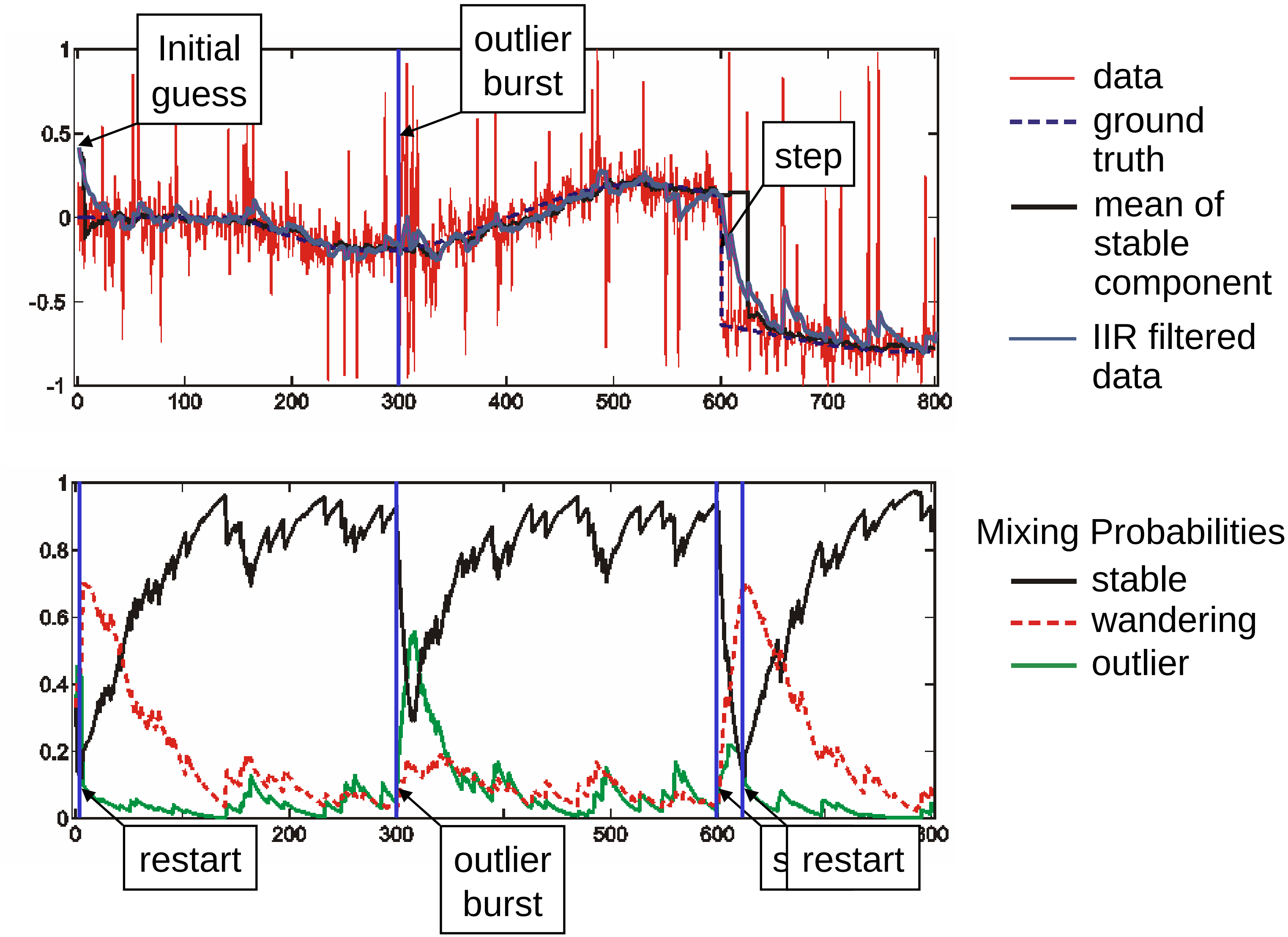
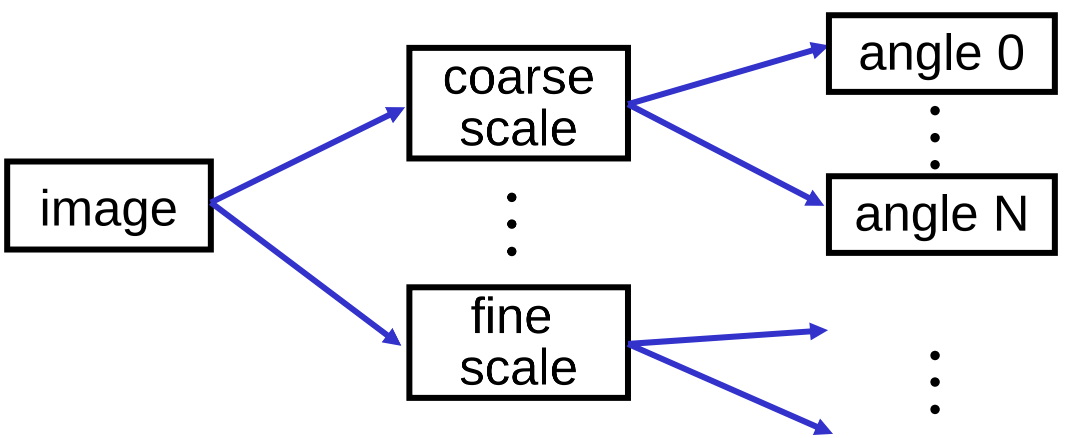


Image Appearance Model

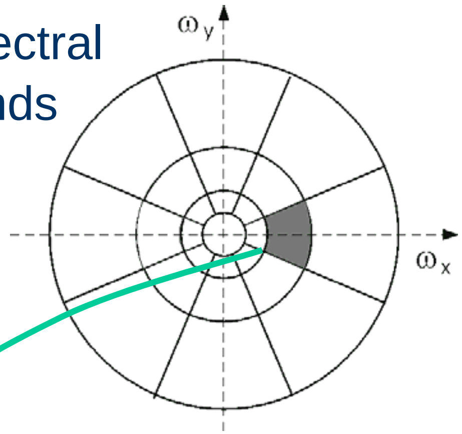
Goal: Identify stable image properties at different locations, scales and orientations.

Approach: Family of linear filters that are tuned to different scales and orientations (wavelets, QMFs, …)

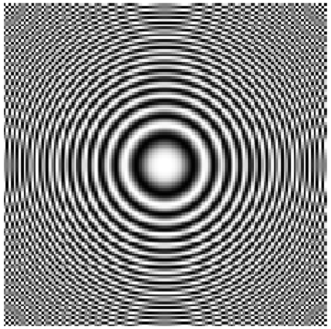


Steerable Pyramid Transform

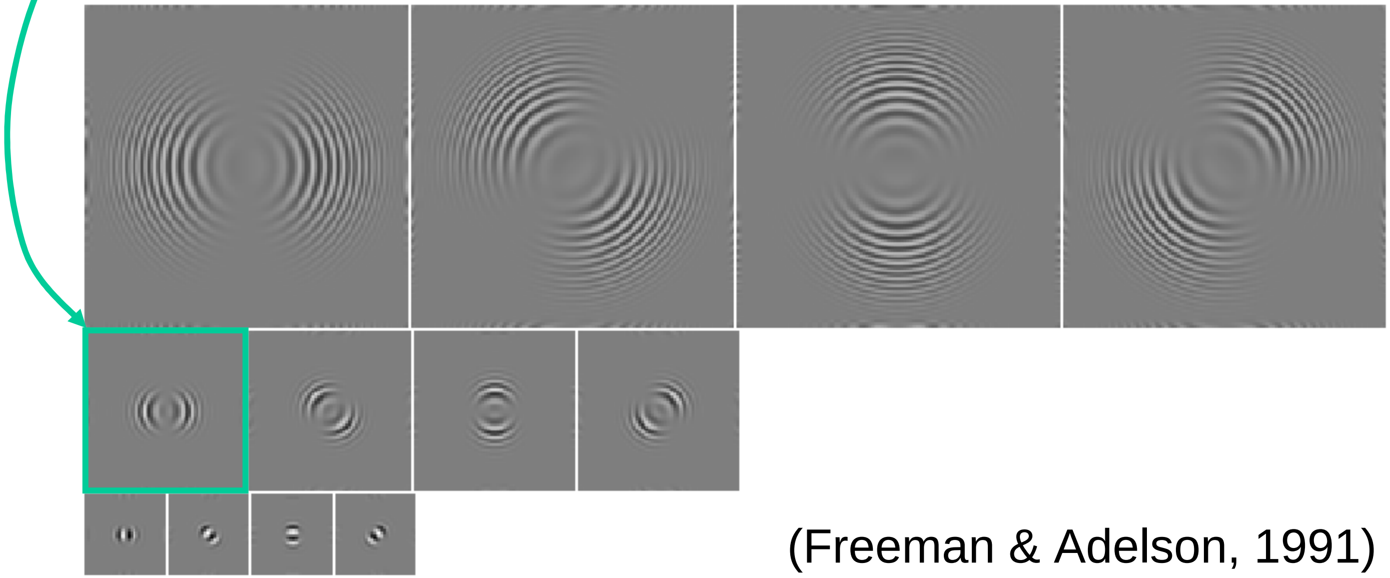
Spectral bands



Zone Plate



Subsampled filter responses at 4 orientations & 3 scales



(Freeman & Adelson, 1991)

Phase-Based Appearance

Band-pass filters at each scale/orientation produce complex-valued responses at each image position:

$$R(\mathbf{x}, t) \;=\; A(\mathbf{x}, t)\, e^{-i\phi(\mathbf{x}, t)}$$

- amplitude $A(\mathbf{x}, t)$ captures signal strength
- phase $\phi(\mathbf{x}, t)$ captures signal structure

Identify filters with stable phase responses:

- one data stream, $d_t = \phi(\mathbf{x}, t) / \pi$, for each filter output
- usually high-contrast oriented image edges & texture

Demo: Background Appearance

Goal: Learn background model over the entire image.

- time constant = 20 frames

stable mixing probability

stable mean phase

MAP Estimation of Motion Parameters

MAP estimate the motion (warp) parameters maximizes the sum of log likelihood and log prior :

$$O(\mathbf{u}_t) = \log L(D_t \mid \mathbf{u}_t, A_{t-1}, D_{t-1}) + \log p(\mathbf{u}_t \mid \mathbf{u}_{t-1})$$

likelihood prior

where:

warp parameters: $\mathbf{u}_t$

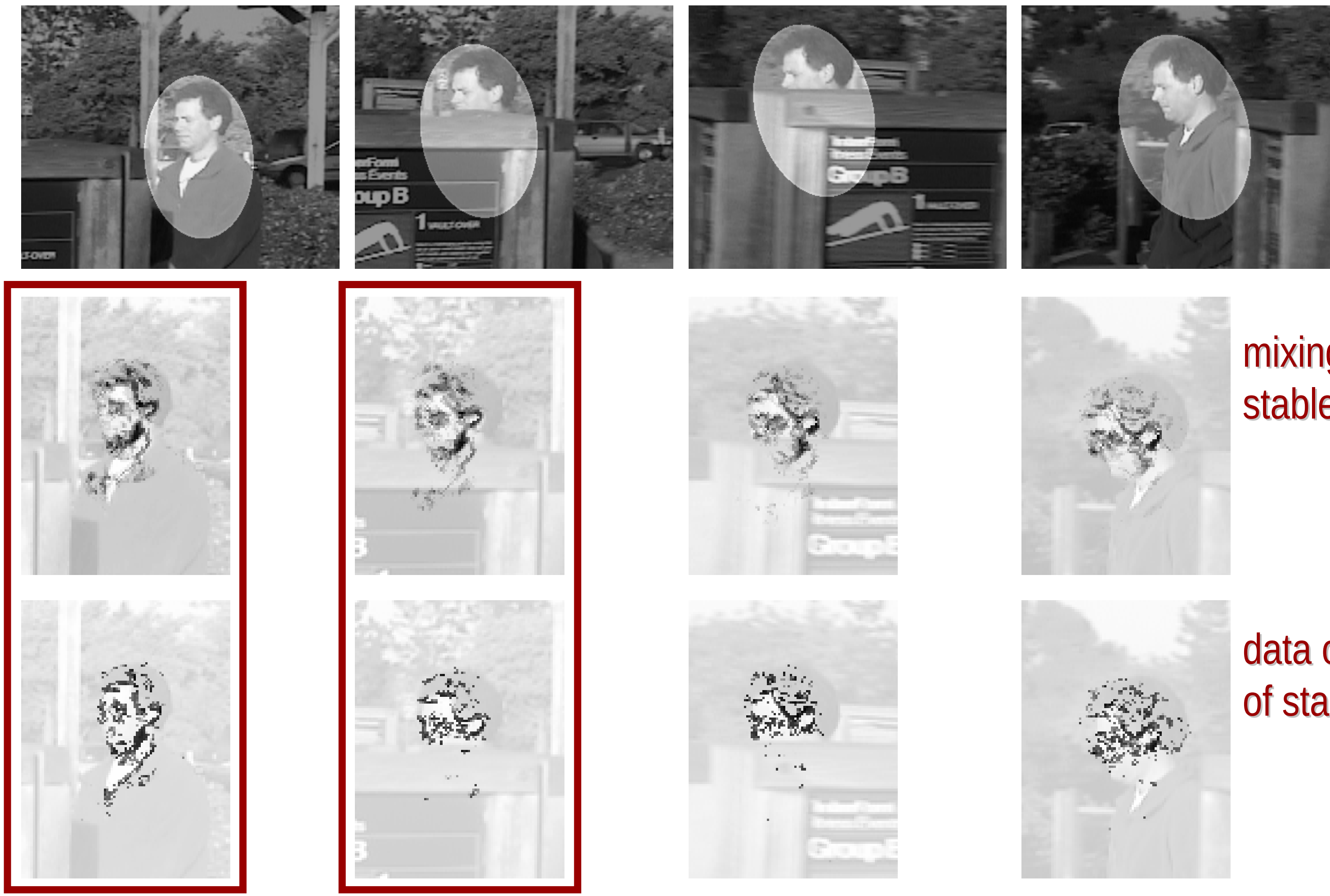
data at time *t*-1: $D_{t-1} = \{d(\mathbf{x}, t-1)\}_{\mathbf{x} \in \mathcal{R}_{t-1}}$

appearance model: $A_t = (\mathbf{q}_t, \mathbf{m}_t)$

parametric motion: $\mathbf{x}_t = \mathbf{w}(\mathbf{x}_{t-1}; \mathbf{u}_t)$

Demo: Occlusion

Demo: Occlusion Details



original images

mixing prob of stable process

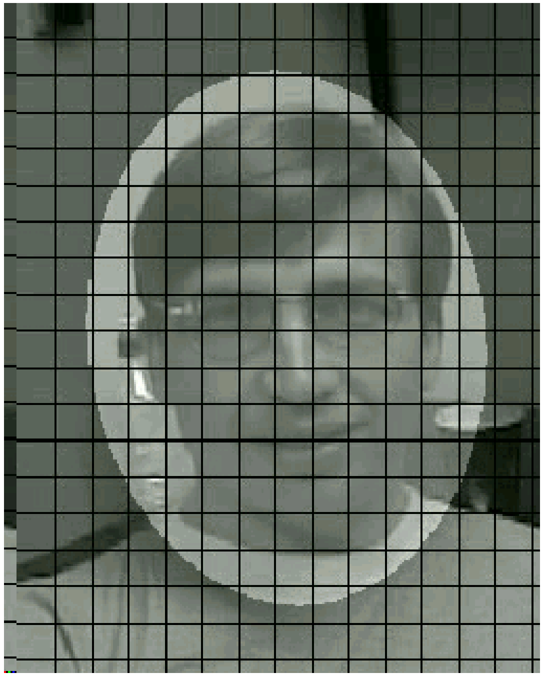
data ownership of stable process

frame 244 frame 257 frame 274 frame 289

Demo: Parking Lot

Failure Mode: complete occlusion.

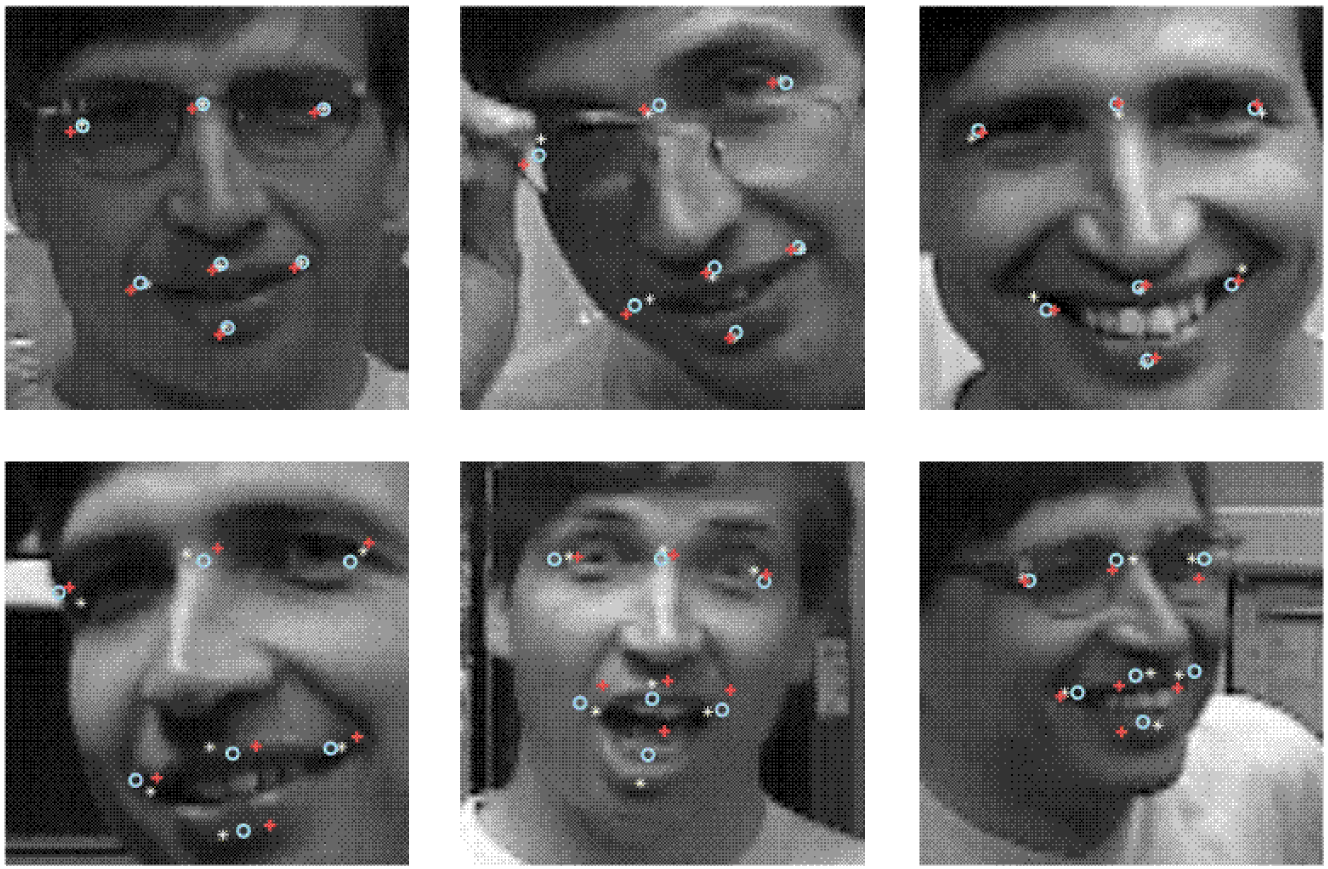
Demo: Alignment Precision



Precise alignment despite:

- smooth deformation (3D motion, non-rigidity)
- abrupt changes (hair, glasses, smiling)
- outliers (occlusions, noise)

RMS Alignment Errors



Manual ground truth (white stars)

Optimal similarity transform (blue circles, 3.1 RMS error)

WSL tracker (red crosses, 5.2 RMS error)

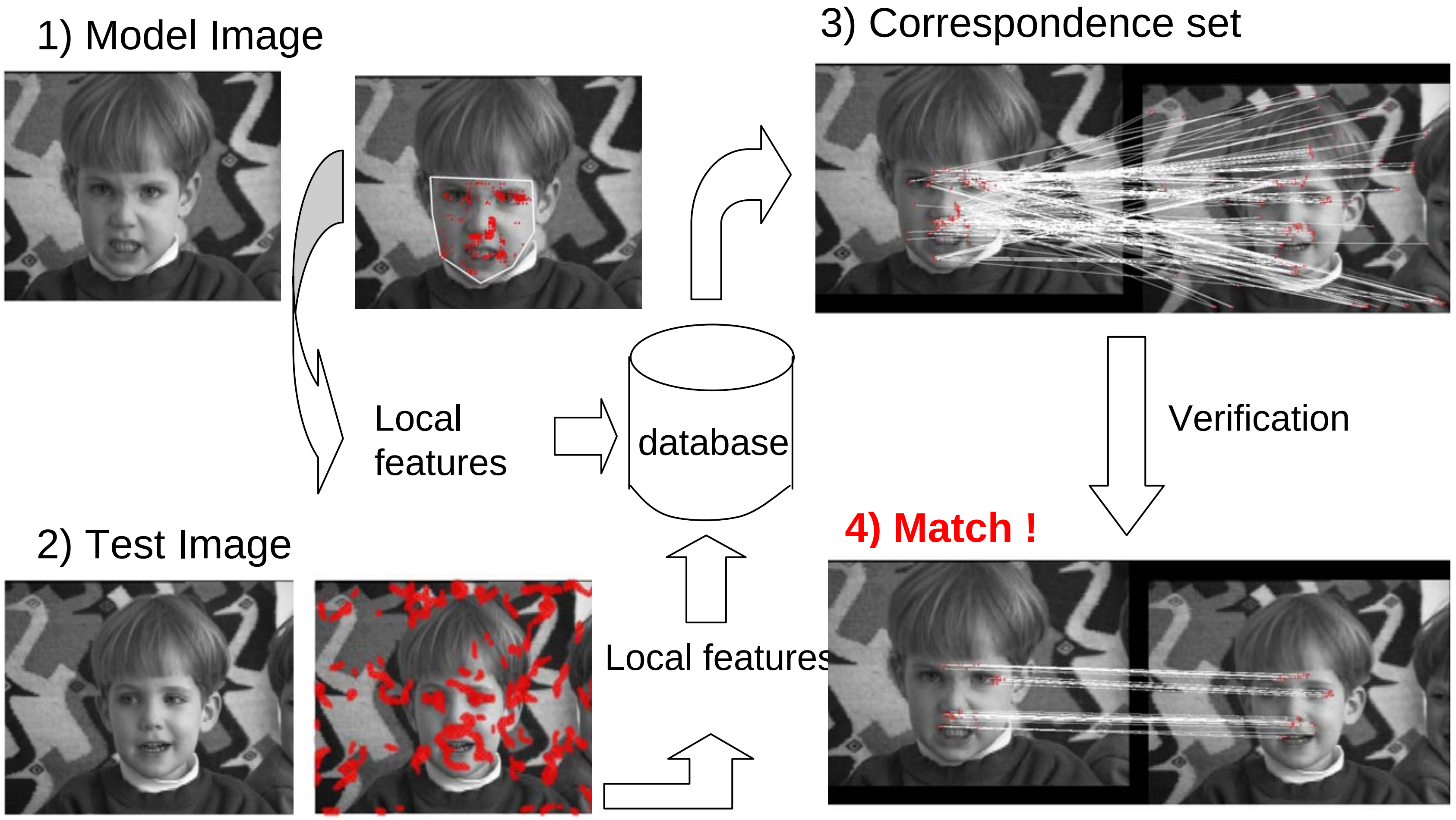
Demo: Bill's Class

Demo: Bill's Class

Demo: Bill's Class

Demo: Bill's Class

Long-Range Motion and Feature Matching



Strategy used for SIFT features introduced by Lowe.

Flexible Matching

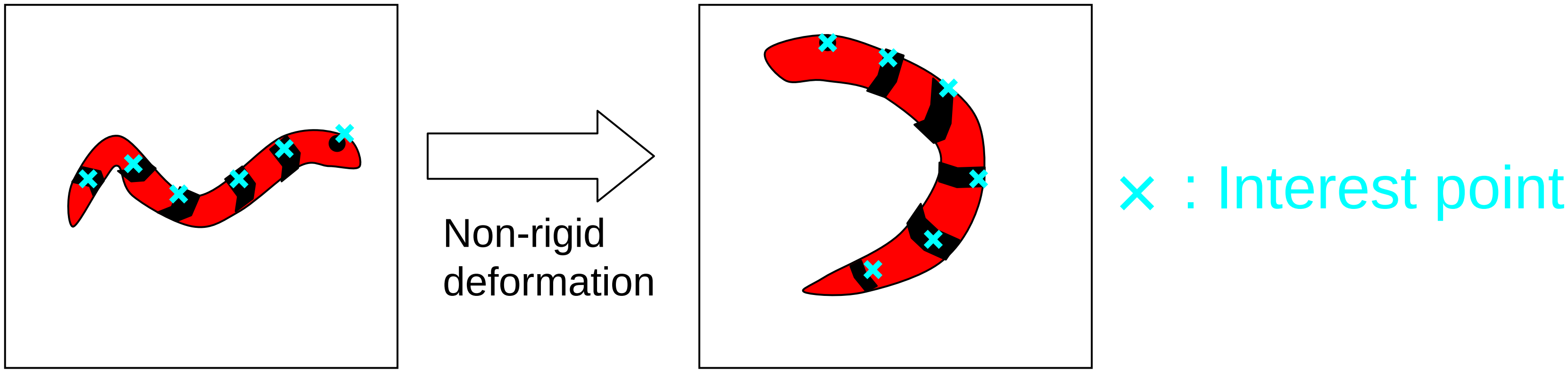


Image patches in interest point neighbourhoods are represented relative to a principal orientation and scale (i.e. local image features such as SIFT).

Flexible spatial arrangements of these local features can be encoded, and efficiently retrieved.

[Carneiro and Jepson, Proc. CVPR 2004]

Long-Range Matching Demos

Image Model.

Matches.