Ruch (Motion)

Rozpoznawanie Obrazów

Krzysztof Krawiec

Instytut Informatyki, Politechnika Poznańska

The importance of visual motion

- Adds entirely new (temporal) dimension to visual information.
- Extremely important in human perception: observable even in random dot images.
- Enables us to compute useful properties of the observed 3D world.

Image sequence

Also: video sequence.

An image sequence is a series of N images (frames), acquired at discrete time instants $t_k = t_0 + k\Delta t$

- Same dimensions, depth ...
- Sampling interval Δt typically constant, but not always.

Tasks related to image sequences

Most common tasks:

- Motion detection: presence of motion
- Qualitative motion estimation: how much motion is there?
- Motion analysis: direction, speed,…
- Motion compensation
- Motion segmentation
- Object tracking

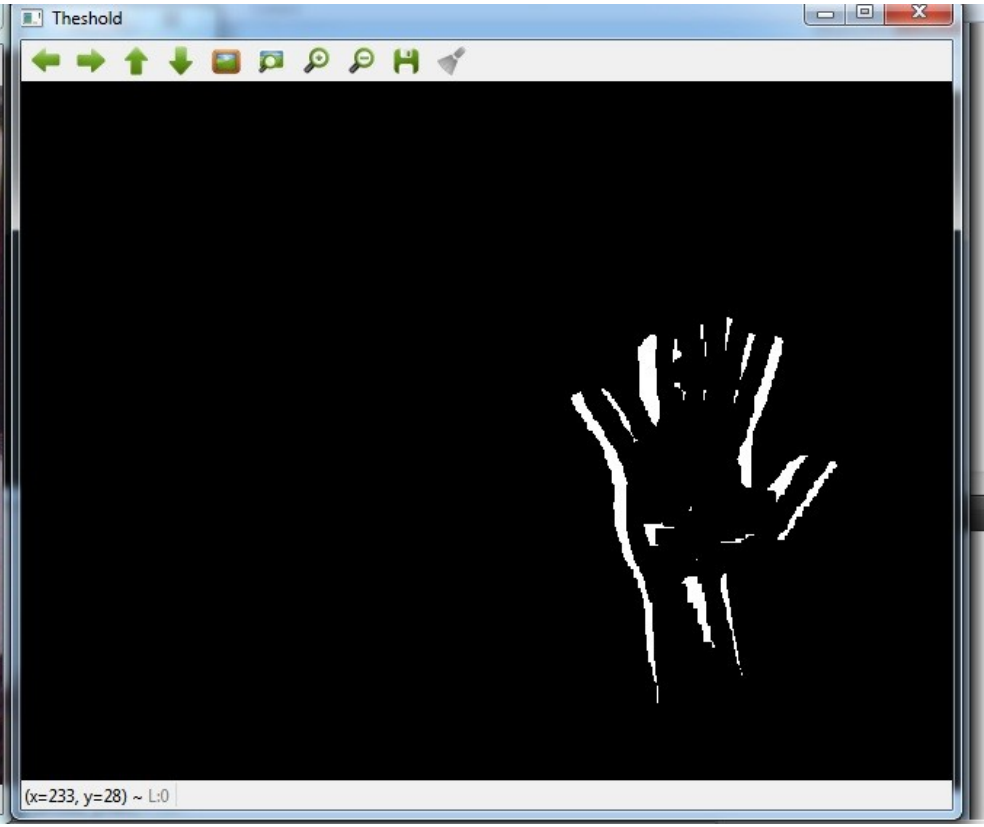
Motion detection

Motion detection:

- Usually implemented as simple *frame differencing*

- A more advanced scheme:
  - Introducing pixel timestamps which store the last luminance change for each pixel separately.
  - ‘Ageing’ the motion information according to timestamps.

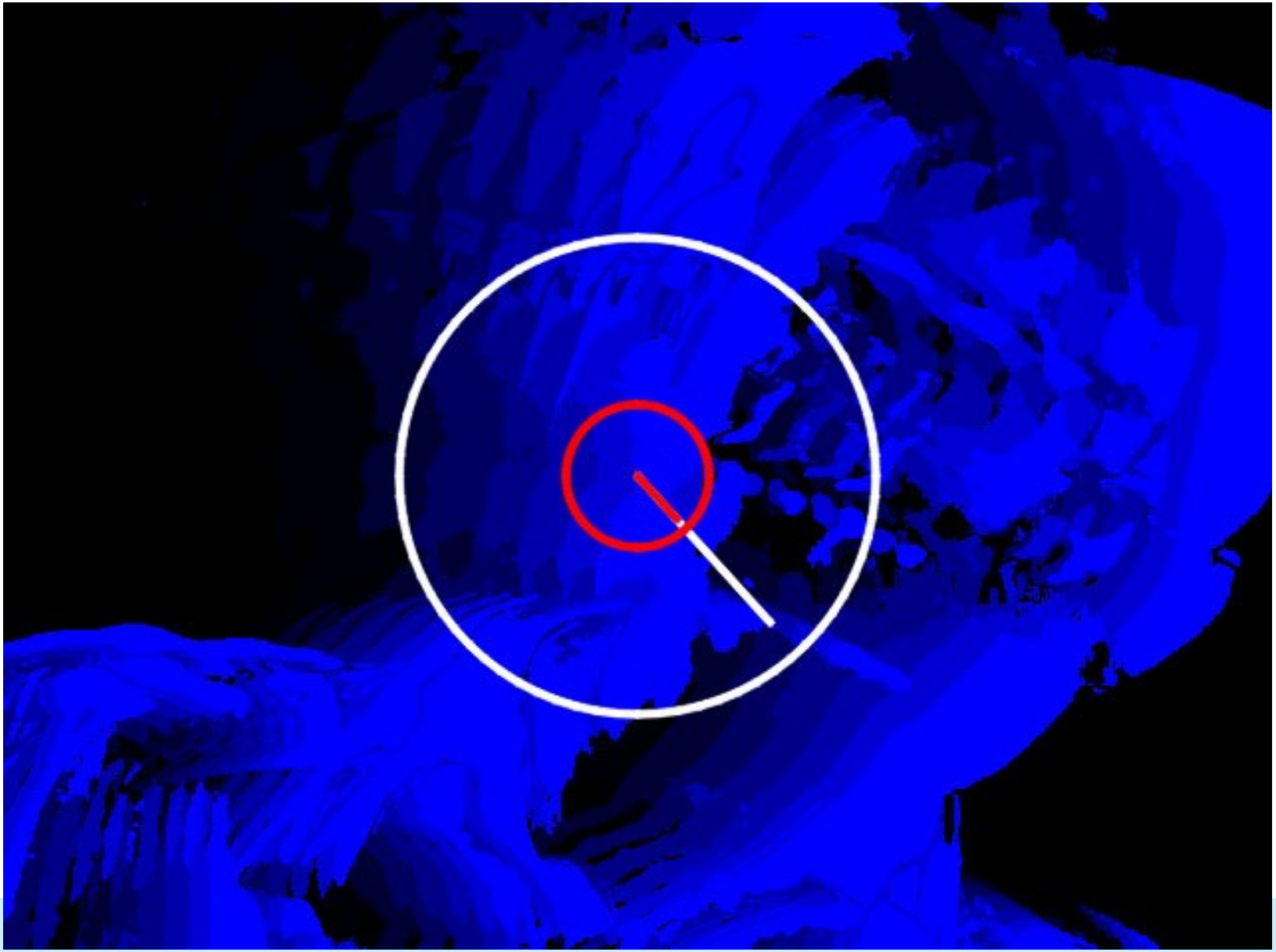
Example



Quantitative motion estimation

In the simplest version, can only give an estimate of amount of motion in the scene.

Example



Motion detection example

Motion detection example

The aspects of motion analysis problem

1. Correspondence: Which elements of the frame correspond to which elements of the next frame of the sequence?
2. Reconstruction: given:
   - a number of corresponding elements, and
   - [possibly] knowledge of the camera's intrinsic parameters

   what can we say about the 3D motion and structure of the observed world?
3. [Segmentation]: If there are multiple independently moving objects in the scene, what are the regions of the image plane which correspond to the different moving objects?

Assumptions

In following, we assume that:

- There is only one, rigid, relative motion between the camera and the observed scene,
- The illumination does not change.

The expected result: Motion field (MF)

Definition:

- *Motion field (MF)* is the 2D vector field (array, matrix) of velocities of the image points, induced by the relative motion between the viewing camera and the observed scene.
- For a single point (*x*,*y*):

$$v = \begin{bmatrix} v_x \\ v_y \end{bmatrix}$$

- May be considered as the projection of the 3D velocity field on the image plane.
- Motion analysis methods aim at estimating MF; MF estimate is called *optical flow*.

Optical flow

The task

Estimate the motion field from image sequence, i.e., from spatial and temporal variations of the image brightness.

In following we formally assume that image brightness is continuous and differentiable as many times as needed in both spatial and temporal domain.

Image brightness constancy equation

Motivation: Under most circumstances, the apparent brightness E of moving objects remains constant:

$$\frac{dE}{dt}=0$$

where E = E(x,y,t), and x and y depend on time t, i.e., x=x(t), y=y(t). Thus:

$$\frac{dE(x(t),y(t),t)}{dt}=\frac{\partial E}{\partial x}\frac{dx}{dt}+\frac{\partial E}{\partial y}\frac{dy}{dt}+\frac{\partial E}{\partial t}=0$$

$$\frac{dE(x(t),y(t),t)}{dt}=\frac{\partial E}{\partial x}\frac{dx}{dt}+\frac{\partial E}{\partial y}\frac{dy}{dt}+E_t=0$$

Image brightness constancy equation

However, dx/dt and dy/dt define the speed of the relative motion, i.e., the motion field v:

$$v = \begin{bmatrix} v_x \\ v_y \end{bmatrix} = \begin{bmatrix} \frac{dx}{dt} \\ \frac{dy}{dt} \end{bmatrix}$$

The partial spatial derivates are components of the spatial image gradient, ∇E

$$\nabla E = \begin{bmatrix} \frac{\partial E}{\partial x} \\ \frac{\partial E}{\partial y} \end{bmatrix} = \begin{bmatrix} E_x \\ E_y \end{bmatrix} = \begin{bmatrix} \frac{\partial E}{\partial x} & \frac{\partial E}{\partial y} \end{bmatrix}^T$$

Image brightness constancy equation

Thus, the equation

$$\frac{dE(x(t),y(t),t)}{dt}=\frac{\partial E}{\partial x}\frac{dx}{dt}+\frac{\partial E}{\partial y}\frac{dy}{dt}+E_t=0$$

becomes:

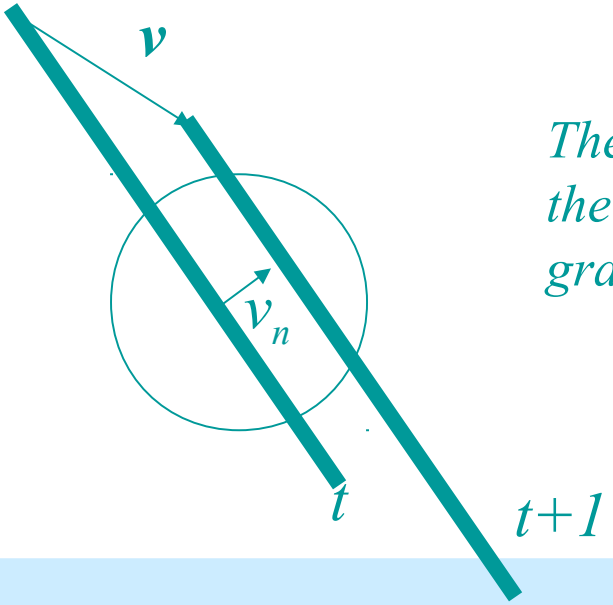
$$(\nabla E)^T v+E_t=0$$

called image Brightness Constancy Equation (BCE)

Note: Can be generalized to Color Constancy

Note: The aperture problem

- BCE allows to compute only a part of the motion field.
- More precisely, the component in direction of the spatial image gradient, v_n (so-called *normal component*).
- Thus: The component of **v** in the direction orthogonal to the spatial image gradient is not constrained by the BCE.
- The velocity in the parallel direction cannot be estimated.



The aperture circle: the support for gradient computation

(drawing for discrete case)

Optical flow

- The optical flow (OF) is a vector field subject to constraint BCE.
  - a.k.a. *apparent motion* of the image brightness pattern.
- OF is an approximation of MF computed under the following assumptions:
  - Lambertian surfaces (reflectance model assuming that each scene point appears equally bright from all viewing directions),
  - Pointwise light source at infinity,
  - No photometric distortion.
- Under these assumptions, the error of approximating MF by OF is:
  - Small at points with high spatial gradient
  - Exactly zero for translational motion or for any rigid motion with the illumination direction parallel to the angular velocity.

Two groups of methods

Two alternative strategies:

- *Differential* methods:
  Perform some computation (examine spatial and temporal variations) at *each* image pixel (dense measures)
- *Matching* methods, a.k.a. feature-based methods
  Perform computation only at a subset of image points – features (sparse measures)

1. Differential (dense) techniques

Differential techniques

Solve BCE by:

1. Solving a system of partial differential equations, or
2. Computing second and higher-order derivatives of the image brightness, or
3. Computing least-squares estimates of parameters characterizing the optical flow.
   Advantages:
   - Non-iterative, genuinely local, less biased than iterative methods.
   - Does not involve higher-order derivatives and are therefore less sensitive to noise.

Methods of estimating MF (computing OF)

The OF algorithm (CONSTANT_FLOW)

A basic differential technique (group 1).

Assumptions:

- The BCE yields a good approximation of the MF.
- The MF is well approximated by a constant vector field within any small patch of the image plane, i.e. the BCE

$$(\nabla E)^T v + E_t = 0$$

is fulfilled for each point p_i within a small NxN patch Q (typically N=5)

CONSTANT_FLOW

We are looking for a constant vector v that minimizes the following expression (within a given image patch Q):

$$\sum_{p_i \in Q} \left[(\nabla E)^T v + E_t \right]^2$$

=> Least squares problem:

$$Av - b = 0$$

where A is N^2x2 matrix. i-th row of A is the spatial image gradient at point p_i, i.e.:

$$A = \begin{bmatrix} \nabla E(p_1) \\ \nabla E(p_2) \\ \vdots \\ \nabla E(p_{N \times N}) \end{bmatrix}$$

$$b = -\left[E_t(p_1), \ldots, E_t(p_{N \times N}) \right]$$

CONSTANT_FLOW

- The least squares solution of that overconstrained system can be computed as:

$$A^T A v = A^T b$$

$$\bar{v} = \left(A^T A \right)^{-1} A^T b$$

- Solved using Gaussian elimination, but more often using singular value decomposition (SVD), or other forms of decomposition.
- Computed value is the optical flow at the center of patch Q
- In this way, we may compute optical flow for all image pixels, considering patches centered at particular image pixels.

Algorithm CONSTANT_FLOW

Input: a time-varying sequence of n images E_1, E_2, …, E_n

1. ***Preprocessing:***
   a) Filter each image with a Gaussian filter of standard deviation σ_s along each spatial dimension (typically σ_s=1.5 pixels)
   b) Filter each image along the temporal dimension with a Gaussian filter of standard deviation σ_t along each spatial dimension (typically σ_t=1.5 frames)
2. ***For each pixel of each image of the sequence:***
   a) Compute matrix **A** and vector **b**
   b) Compute the optical flow **v**

Remarks

Required preprocessing:

- spatial filtering to attenuate noise, and
- temporal filtering to prevent aliasing[1] in temporal domain.

[1] Different continuous signals to become indistinguishable (or aliases of one another) when sampled.

The maximum speed that can be ‘measured’ by the algorithm depends on the size of the temporal filter.

More examples

http://eric-yuan.me/coarse-to-fine-optical-flow/

Extension of CONSTANT_FLOW alg.

- The optical flow at the center of each patch Q is computed based on the entire patch.
- The further from the center of the patch, the more the estimate differs from the estimate at the center.
- Improvement: introduce weighting and pay more attention to the pixels close to the patch center.
- Weighted least-square method (W – weight matrix):

$$\bar{v} = \left(A^T W^2 A \right)^{-1} A^T W^2 b$$

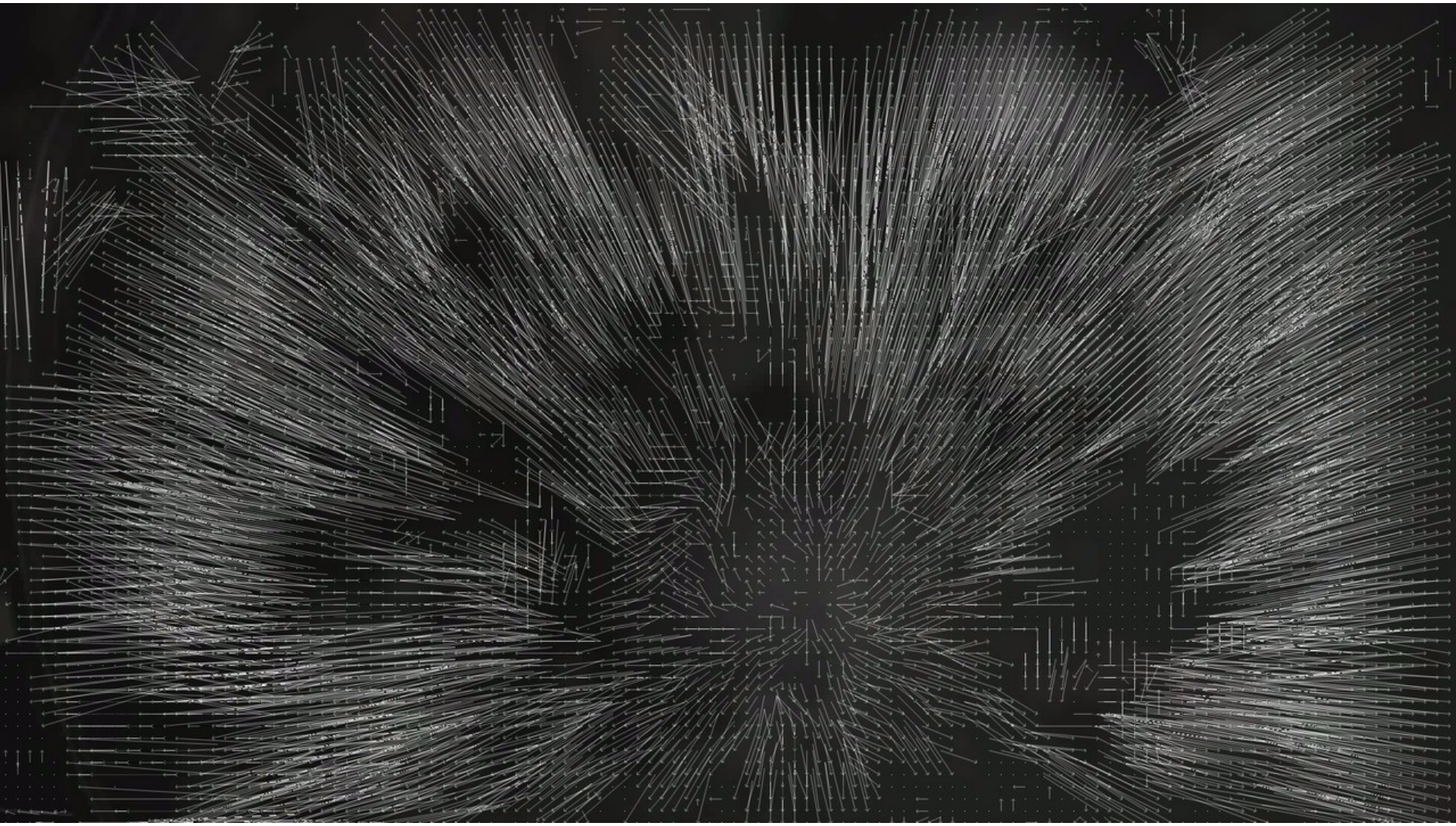
When CONSTANT_FLOW fails?

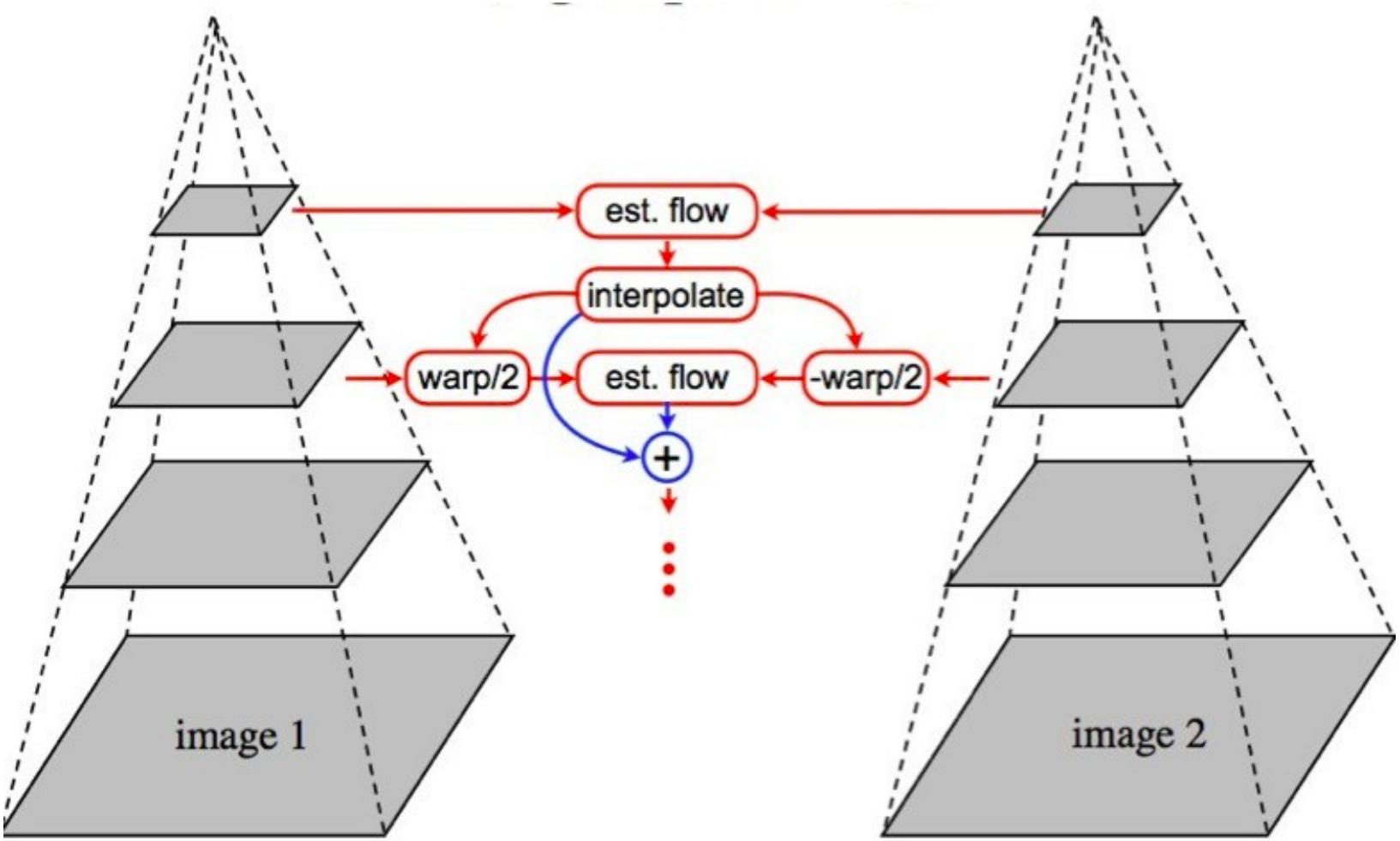
- The matrix:

$$A^T A = \begin{bmatrix} \sum E_x^2 & \sum E_x E_y \\ \sum E_x E_y & \sum E_y^2 \end{bmatrix}$$

computed over an image region Q is singular if and only if all the spatial gradients in Q are null or parallel.

- In such a case, only the normal flow may be estimated (see: aperture problem).
- (It may be shown that the eigenvectors of the above matrix encode edge directions, while eigenvalues encode edge strength)
  - For *uniform regions*, both eigenvalues will be zero.
  - For *edges* only one eigenvalue will be non-zero.
  - For *corners* both eigenvalues will be significantly greater than zero.




est. flow
interpolate
warp/2
est. flow
-warp/2
+
image 1
image 2

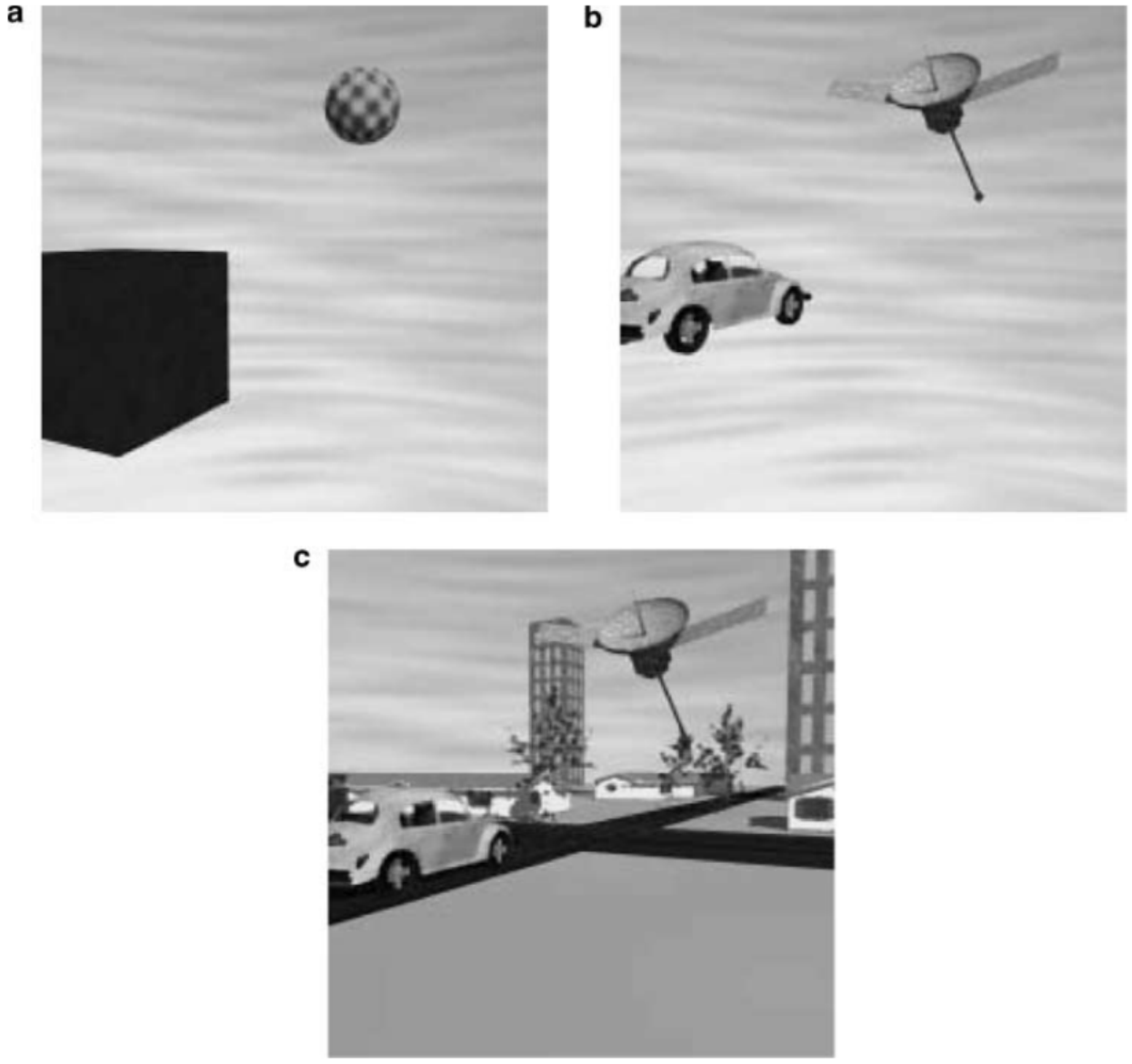
Benchmarking optical flow

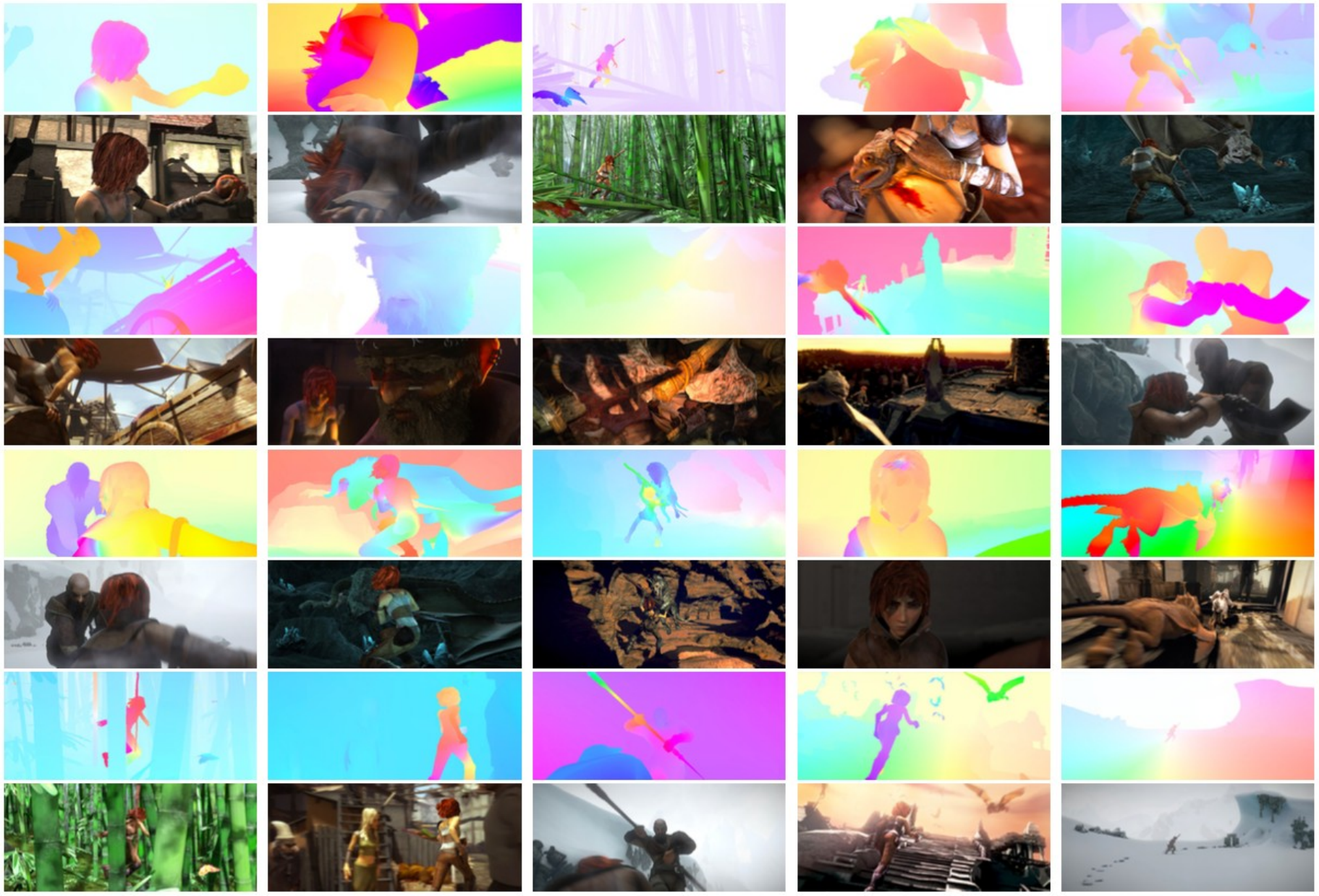
Examples from:

B. McCane, K. Novins, D. Crannitch and B. Galvin, *On Benchmarking Optical Flow*

Daniel J. Butler, Jonas Wulff, Garrett B. Stanley, and Michael J. Black, *A Naturalistic Open Source Movie for Optical Flow Evaluation*

Benchmarking optical flow





Problems with motion analysis

- More sensitive to noise than stereo analysis (as the ‘base’ shift between pairs is usually much smaller than in stereo)
- Often requires on-line processing at high frame rate => computational requirements

2. Feature-based (matching) techiques

Feature-based (matching) techiques

- Perform computation only at a subset of image points – features (sparse measures)
- Two subtypes:
  - Two-frame methods = feature matching
  - Multiple-frame methods = feature tracking

2a. Two-frame method

- The idea: iterate the CONSTANT_FLOW algorithm over a set of feature points.
- Feature points = centers of small, square image regions, preselected using some procedure.
  - E.g., the centers of those regions for which the smallest eigenvalue of $A^T A$ is larger than a threshold

Alg. FEATURE_POINT_MATCHING

Input:

- Two frames of video sequence I_1 and I_2
- The sets of feature points in I_1 and I_2

Let:

- Q_1, Q_2, and Q' be three NxN image regions,
- τ a fixed positive number
- **d** – unknown displacement between I_1 and I_2 of a feature point **p** on which Q_1 is centered.

Alg. FEATURE_POINT_MATCHING

For each feature point p:

1. ***Set d=0 and center Q_1 on p***
2. ***By running CONSTANT_FLOW algorithm, estimate the displacement d_0 of p (center of Q_1), and let $d:=d+d_0$***
3. ***Warp (displace) Q_1 according to d_0, obtaining Q'***
   Sample Q' from I_2 and compare it to Q_2 (corresponding patch), computing $SSD(Q',Q_2)$
4. ***If $SSD(Q',Q_2) > \tau$, set $Q_1:=Q'$ and go to step 1;***
   otherwise exit and proceed with next feature point

Remarks

- SSD – sum of square differences
- For smoothing in CONSTANT_FLOW algorithm, it is recommended to use regions larger (at lest twice) than patches Q,

Demonstration

- Using pyramidal version of Lucas-Kanade
- Extension:
  - finding good features to track using by locating pixels with significantly positive minimal eigenvalue

https://www.youtube.com/watch?v=zNqCNMefyV8

https://www.youtube.com/watch?v=5rR_9YIcg_s

Motion vs. stereo vision

- Spatial differences between consecutive frames are usually much smaller than those found in the typical stereo pairs.
  - => The correspondence problem may be cast as the problem of estimating the apparent motion of the image brightness pattern, i.e., optical flow.
- The relative displacement between the viewing camera and the scene is not necessarily caused by a single 3D rigid transformation (as it is the case in stereo).
- More data available in video stream than in stereo pair
  - Past history of features' motion enable predicting disparities in subsequent frames.

2b. Multiple-frame methods

A.k.a feature tracking: matching features from frame to frame in long sequences of images.

Observations:

- If the motion of the observed scene is continuous, we should be able to make predictions on the motion of the image points.
  - Possible improvement in comparison with two-frame matching
  - In other words: we should be able to use the disparities from previous frames to make predictions on the disparities between following frames.

Intro

Consider one feature point $p_k=[x_k,y_k]^T$ (k-frame number) acquired at instant t_k, moving with velocity $v_k=[v_{x,k},v_{y,k}]^T$

The motion on the image plane may be described by the state vector

$$x=\left[x_k, y_k, v_{x,k}, v_{y,k}\right]^T$$

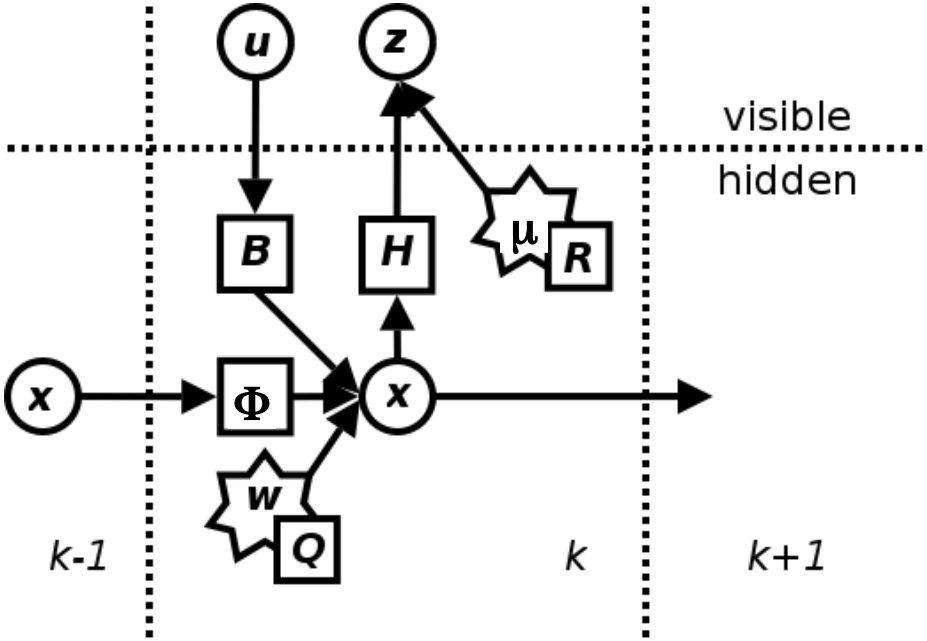
Assuming a sufficiently small sampling interval, we write the system model of the linear Kalman filter:

$$p_k = p_{k-1} + v_{k-1} + \xi_{k-1}$$
$$v_k = v_{k-1} + \eta_{k-1}$$

Where ξ and η are zero-mean, white Gaussian random processes modeling the system noise.

Linear Kalman filter

- a.k.a. Linear Quadratic Estimation (in control theory)
- An analog to Hidden Markov Model; the differences:
  - The hidden state variables are continuous
  - Assumes Gaussian noise
- Components:
  - State vector x
  - State transition model Φ,
  - Control input model B,
  - Observation model H
  - Control vector u,
  - Noises: w, μ given by covariance matrices Q and R



In terms of state vector

$$x_k = \Phi_{k-1} x_{k-1} + w_{k-1}$$

$$\text{where } \Phi_{k-1} = \begin{bmatrix} 1 & & 1 & \\ & 1 & & 1 \\ & & 1 & \\ & & & 1 \end{bmatrix}, \; w_{k-1} = \begin{bmatrix} \xi_{k-1} \\ \eta_{k-1} \end{bmatrix}$$

Measurements

- The precise position of the point is not known explicitly to the observer.
- The observer knows only the measured position $\mathbf{z}_k$, which is subject to some measurements errors.
- Measurement model of the Kalman filter for this case:

$$z_k = H \begin{bmatrix} p_k \\ v_k \end{bmatrix} + \mu_k, \; H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

- Where $\boldsymbol{\mu}_k$ is a zero-mean, white Gaussian random process (measurement noise).

[Linear] Kalman Filter Algorithm

Input:

- Covariance matrices of system and measurement noise
- 'History': position measurement at time t_k

$$
\begin{aligned}
P_k^{'} &= \Phi_{k-1} P_{k-1} \Phi_{k-1} \Phi_{k-1}^T + Q_{k-1} \\
K_k &= P_k^{'} H_k^T \left(H_k P_k^{'} H_k^T + R_k \right)^{-1} \\
\hat{x}_k &= \Phi_{k-1} \hat{x}_{k-1} + K_k \left(z_k - H_k \Phi_{k-1} \hat{x}_{k-1} \right) \\
P_k &= \left(I - K_k \right) P_k^{'} \left(I - K_k \right)^T + K_k R_k K_k^T
\end{aligned}
$$

- Output: the optimal estimation of the position and velocity at time t_k

Remarks

- The filter integrates the noisy measurements with model predictions,
- The filter quantifies the uncertainty on the state estimate, in the form of the diagonal elements of the state covariance matrix
  - The size of the region to be searched is adjusted dynamically.

Demonstration

- Tracking of a rotating point.
  - Rotation speed is constant (?).
  - Both state and measurements vectors are 1-element (a point angle),
  - Measurement is the real point angle + Gaussian noise.
- The *real* and the *estimated* points are connected with yellow line segment,
- The *real* and the *measured* points are connected with red line segment.
- The yellow segment should be shorter than the red one.

https://www.youtube.com/watch?v=SxtY1jQJ2fc

Object tracking

Object tracking

Methods:

- Based on color histograms
- Based on snakes

https://www.youtube.com/watch?v=36j238XtcIE

Applications of motion estimation

Motion compensation

- The goal: To compensate the camera motion (relative to the scene).
- Motion types:
  - dolly (forward, backwards),
  - track (left, right),
  - boom (up, down),
  - pan (left, right),
  - tilt (up, down)
  - roll (along the view axis).
- Subtypes:
  - Global motion compensation
  - Block motion compensation
    - Advanced case: with variable block size

Motion compensation

Hardware solutions: physics-based image stabilization methods (a.k.a. vibration reduction), which usually involve:

- Moving a lense in the objective(e.g., *Image Stabilizer* by Canon, *Vibration Reduction* by Nikon),
- Moving the image sensor (e.g., Sony's *Steady Shot*),

Software approach:

- Compute the OF and move the frame in the opposite direction

..\..\..\root\secpl\vis\bin\stabilizer.exe

Related topics

- Figure-ground separation/segmentation
  Cognitive ability to separate elements based upon contrast, that is, dark and light, black and white
- May give not unique results: