Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

MIN-Fakultät
Fachbereich Informatik
Arbeitsbereich SAV/BV (KOGS)

Image Processing 1 (IP1)
Bildverarbeitung 1

Lecture 17 – Motion Analysis 1

Winter Semester 2015/16

Slides: Prof. Bernd Neumann
Slightly revised by: Dr. Benjamin Seppke & Prof. Siegfried Stiehl

Motion Analysis

Motion analysis of digital images is based on a temporal sequence of image frames of a coherent scene.

"sparse sequence"	→	few frames, temporally spaced apart, considerable differences between frames
"dense sequence"	→	many frames, incremental time steps, incremental differences between frames
video	→	50 half frames per sec, interleaving, line-by-line sampling

Motion detection

Register locations in an image sequence which have changed due to motion

Moving object detection and tracking

Detect individual moving objects, determine and predict object trajectories, track objects with a moving camera

Derivation of 3D object properties

Determine 3D object shape from multiple views ("shape from motion")

Case Distinctions for Motion Analysis

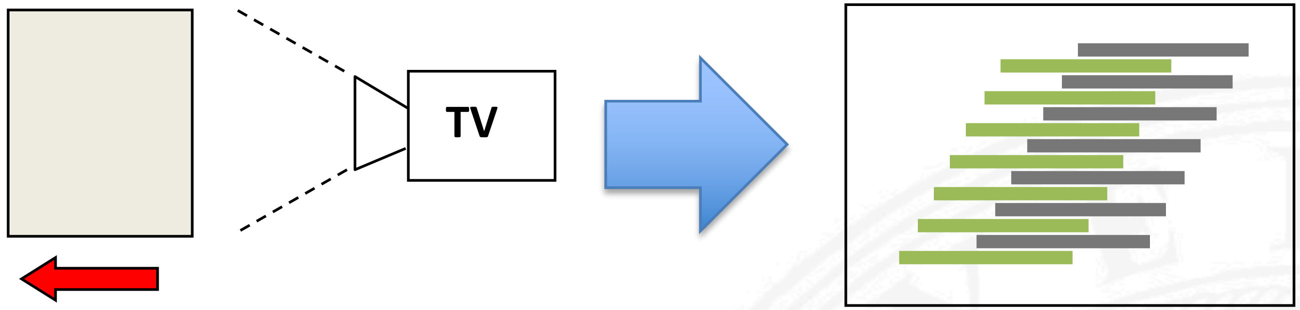
stationary observer
moving observer
single moving object
multiple moving objects
rigid objects
jointed objects
deformable objects
perspective projection
weakly perspective projection
orthographic projection
rotation only
translation only
unrestricted motion
2 image analysis
multiple image analysis
incremental motion
large-scale motion
B/W images

colour images
xray images
IR images
natural images
noisy data
ideal data
monocular images
stereo images
dense flow
sparse flow
no flow
paralaxis
quantitative motion
qualitative motion
small objects
extended objects
polyeder
smooth objects

arbitrary objects
matte surfaces
specular surfaces
textured surfaces
arbitrary surfaces
without occlusion
with occlusion
uncalibrated camera
calibrated camera
data-driven
expectation-driven
real-time
no real-time
parallel computation
sequential computation

Many motion analysis methods are only applicable in restricted cases!

Motion in Video Images



moving object

TV halfframes

TV-rate sampling affects images of moving objects:

- contours show saw-tooth pattern
- deformed angles
- limited resolution

Example:

- 512 pixels per row
- length of dark car is ca. 3.5 m → 130 pixel
- speed is ca. 50 km/h → 14 m/s
- displacement between halfframes is ca. 10 pixel

Difference Images

An obvious technique for <u>motion detection</u> is based on difference images:

- take the pixelwise difference of images of a sequence
- threshold the magnitude of the differences
- regions above threshold may be due to motion

Examples:


frame1


frame12


frame2 - frame1 threshold 30


frame12 - frame1 threshold 30


frame34 - frame1 threshold 30


frame34 - frame1 without threshold

Note effects which prohibit reliable motion detection:

- phase jitter between frames (pixels do not correspond exactly)
- spurious motion of branches, pedestrians, dogs, etc.
- motion of uniform brightness regions does not show
- temporal changes of illumination cause non-motion differences

Counting Differences

If the goal is to isolate the images of moving objects, it may be useful to

- count how often a pixel differs from its initial value (first-order difference picture **FODP**)
- count how often a pixel of a FODP region differs from its previous value (second-order difference picture **SODP**)

(R. Jain 76)

frame1

frame4 - frame1
FODP (yellow)
SODP (red)

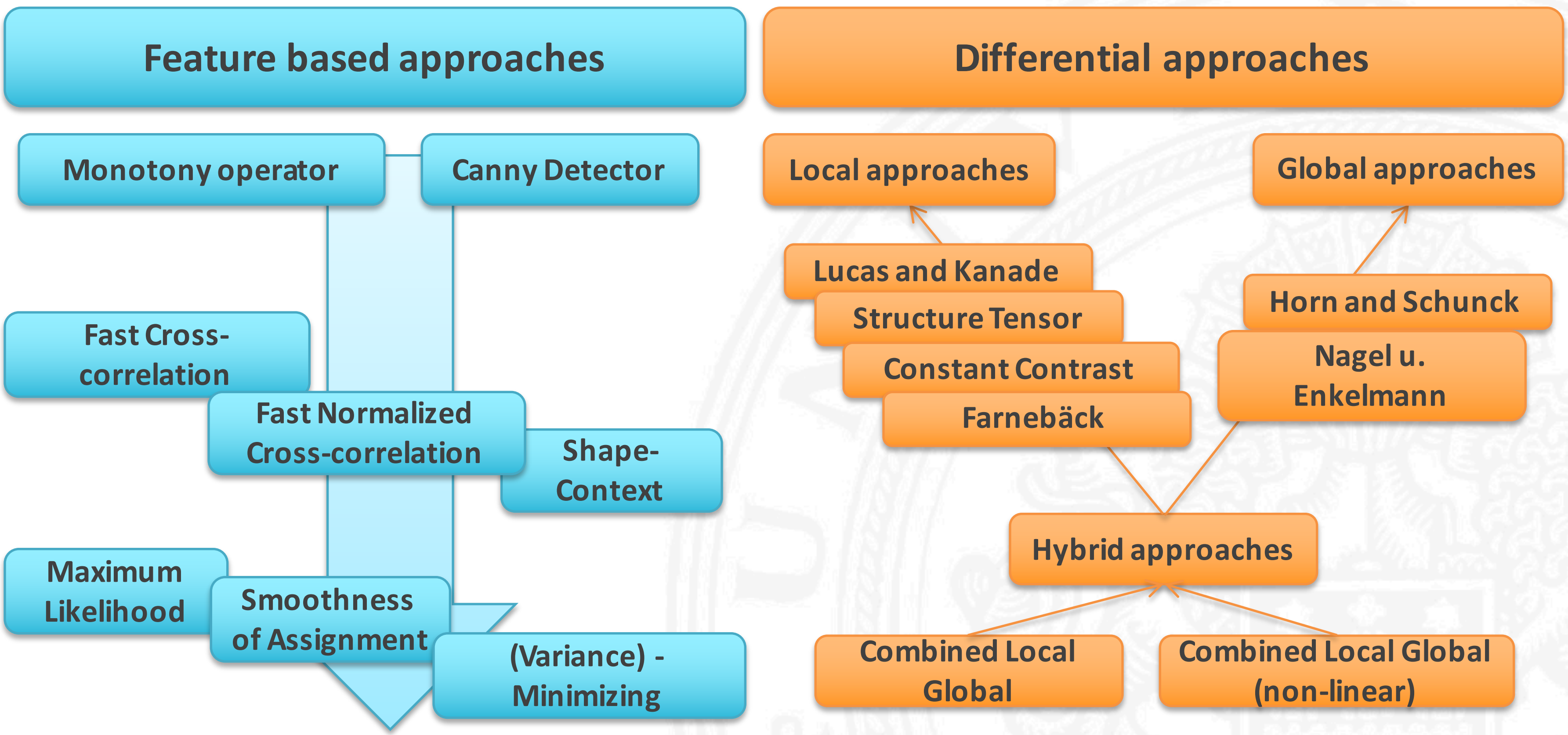
frame10 - frame1
FODP (yellow)
SODP (red)

frame30 - frame1
FODP (yellow)
SODP (red)

Note: The problem of uniform brightness regions is not really overcome.

Motion Derivation Algorithms

Some of the most commonly used motion detection approaches:



Feature Based Approaches

The intuitive way to derive motion from an image sequence:

1. Detect features on each (subsequent) image of the sequence.
2. Determine (statistical) correspondence between the features.
3. Based on (2) and other constraints, perform an assignment of the detected features.
4. Collect all displacements and combine them to describe the motion inside the sequence.

For these approaches, the derivation of motion equals the Solution of the Correspondence Problem

The Correspondence Problem

The correspondence problem is to determine which interest points in different frames of a sequence mark the same physical part of a scene.

Difficulties:

- The scene may not offer enough structure to uniquely locate points.
- The scene may offer too much structure to uniquely locate points-
- Geometric features may differ strongly between frames.
- Photometric features differ strongly between frames.
- There may be no corresponding point due to occlusion.

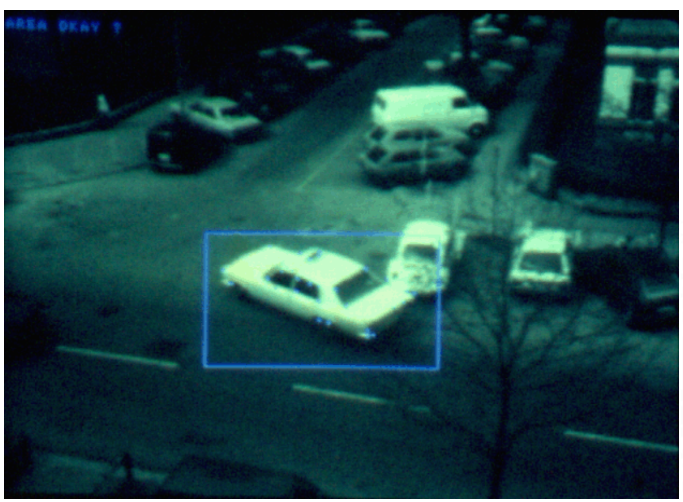
Note that these difficulties apply to single-camera motion analysis as well as multiple-camera 3D analysis (e.g. binocular stereo).

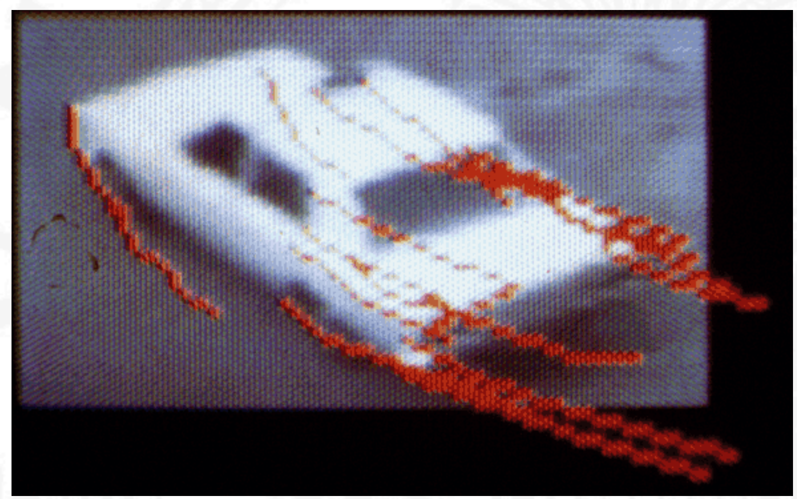
Corresponding Interest Points

Detection of moving objects by

1. finding "interest points" in all frames of a sequence
2. determining the correspondence of interest points in different frames
3. chaining correspondences over time
4. grouping interest points into object candidates

Example: Tracking interest points of a taxi turning off Schlüterstraße
(Dreschler and Nagel 82)



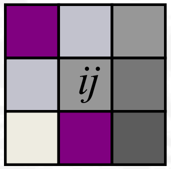


Moravec Interest Operator

Interest points (feature points) are image locations where an interest operator computes a high value. Interest operators measure properties of a local pixel neighbourhood.

Moravec interest operator:

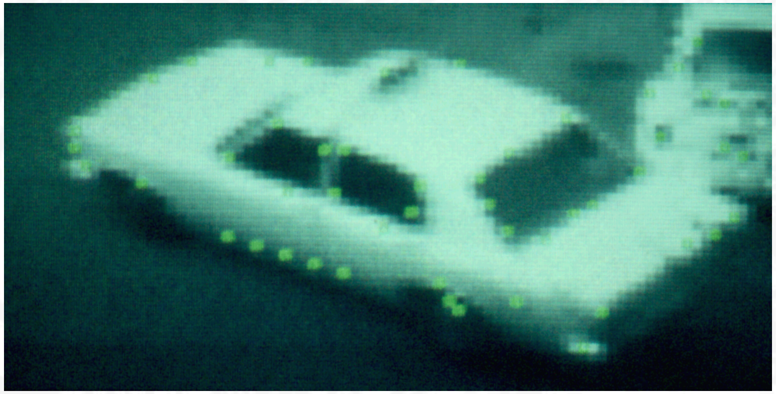
$$M(i,j) = \frac{1}{8} \sum_{m=i-1}^{i+1} \sum_{n=j-1}^{j+1} | g(m,n) - g(i,j) |$$



This simple operator measures the dissimilarity of a point w.r.t. its surrounding.

Refinement of Moravec operator:

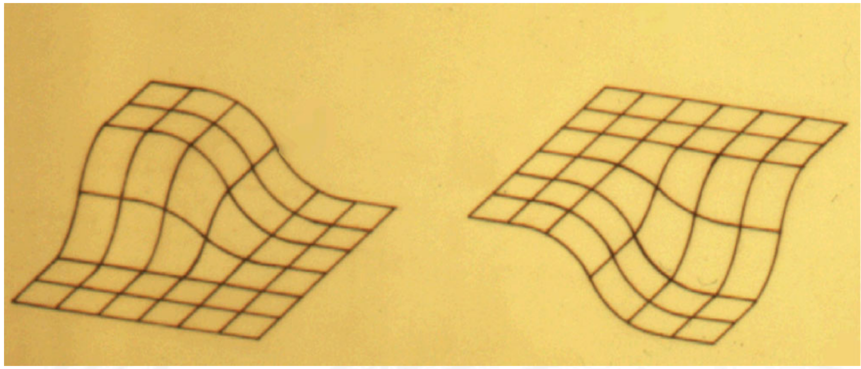
Determine locations with strong brightness variations along two orthogonal directions (e.g. based on variances in horizontal, vertical and diagonal direction).



Interest points in different frames may not correspond to identical physical object parts due to their small neighbourhood and noise.

Corner Models

Interest points may be based on models of interesting facets of the image function, e.g. corners.



"corner" = location with extremal Gaussian curvatures

(Dreschler and Nagel 81)

Zuniga-Haralick operator:

- fit a cubic polynomial

$$f(i,j) = c_1 + c_2x + c_3y + c_4x^2 + c_5xy + c_6y^2 + c_7x^3 + c_8x^2y + c_9xy^2 + c_{10}y^3$$

For a 5x5 neighbourhood the coefficients of the best-fitting polynomial can be directly determined from the 25 greyvalues

- compute interest value from polynomial coefficients

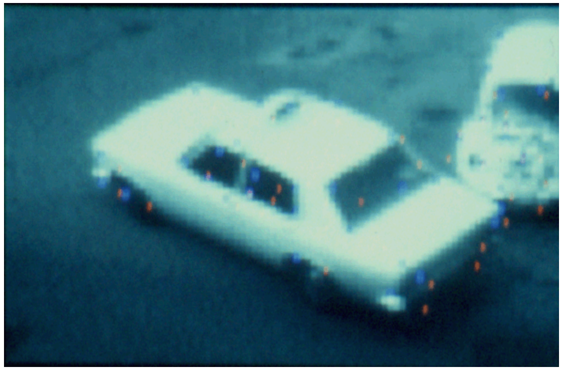
$$ZH(i,j) = \frac{-2(c_2^2c_6 - c_2c_3c_5 - c_3^2c_4)}{\left(c_2^2 + c_3^2\right)^{\frac{3}{2}}}$$

measure of "cornerness" of the polynomial

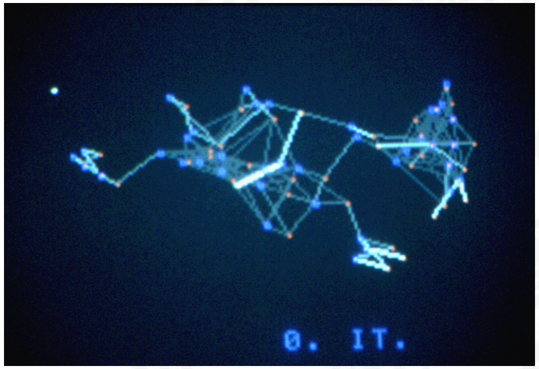
Correspondence by Iterative Relaxation

Basic scheme (Thompson and Barnard 81) modified by Dreschler and Nagel:

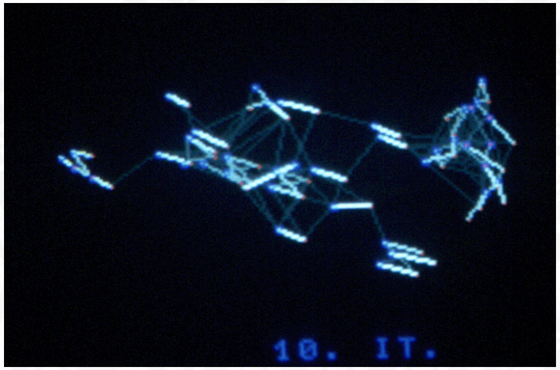
- initialize correspondence confidences between all pairs of interest points in 2 frames based on
  - similarity of greyvalue neighbourhoods
  - plausibility of distance (velocity)
- modify confidences iteratively based on
  - similarity of displacement vectors in the neighbourhood
  - confidence of competing displacement vectors



interest points of 2 frames (red and blue)



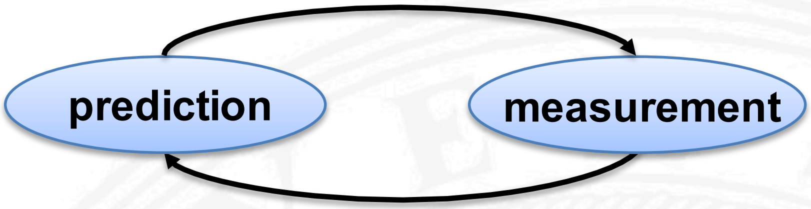
initialized confidences



confidences after 10 iterations

Kalman Filters I

A Kalman filter provides an iterative scheme for (i) predicting an event and (ii) incorporating new measurements.



Assume a *linear* system with observations depending *linearly* on the system state, and *white* Gaussian noise disturbing the system evolution and the observations:

$$\vec{x}_{k+1} = A_k \vec{x}_k + \vec{w}_k$$

$$\vec{z}_k = H_k \vec{x}_k + \vec{v}_k$$

- $\vec{x}_k$ quantity of interest ("state") at time k
- A_k model for evolution of $\underline{x}_k$
- $\vec{w}_k$ zero mean Gaussian noise with covariance Q_k
- $\vec{z}_k$ observations at time k
- H_k relation of observations to state
- $\vec{v}_k$ zero mean Gaussian noise with covariance R_k

Often, A_k, Q_k, H_k and R_k are constant.

What is the best estimate of $\vec{x}_k$ based on the previous estimate $\vec{x}_{k-1}$ and the observation $\vec{z}_k$?

Kalman Filters II

The best a priori estimate of $\vec{x}_k$ before observing $\vec{z}_k$ is:

$$\vec{x}_k' = A_{k-1}\vec{x}_{k-1}$$

After observing $\vec{z}_k$, the a priori estimate is updated by

$$\vec{x}_k'' = \vec{x}' + K_k\left(\vec{z}_k - H_k\vec{x}_k'\right)$$

K_k is Kalman <u>gain matrix</u>. $\mathrm{K_k}$ is determined to minimize the a posteriori variance P_k“ of the error $\vec{x}_k - \vec{x}_k''$. The minimizing K_k is

$$K_k = P_k'\left(H_k\right)^T\left(H_k P_k'\left(H_k\right)^T + R_k\right)^{-1} \quad \text{with } P_k' = A_k P_{k-1}''\left(A_k\right)^T + Q_{k-1} \quad \text{and: } P_k'' = \left(I - K_k H_k\right)P_k'$$

P_k´ is covariance of error $\vec{x}_k - \vec{x}_k'$ before observation of $\vec{z}_k$.

Iterative order of computations:

initialization: $\vec{x}_0$ → 1., P_0´ → 2.

1. $\vec{x}_k' = A_{k-1}\vec{x}_{k-1}$
2. $P_k' = A_k P_{k-1}''\left(A_k\right)^T + Q_{k-1}$

1. $K_k = P_k'\left(H_k\right)^T\left(H_k P_k'\left(H_k\right)^T + R_k\right)^{-1}$
2. $\vec{x}_k'' = \vec{x}' + K_k\left(\vec{z}_k - H_k\vec{x}_k'\right)$
3. $P_k'' = \left(I - K_k H_k\right)P_k'$

k := k+1

Kalman Filter Example

Track positions p_k and velocities v_k of an object moving along a straight line.

Assume unknown accelerations a_k with probability density $N(0, q^2)$ and measurements of positions p_k corrupted by white noise b_k with probability density $N(0, r^2)$.

$$\vec{x}_{k+1} = A_k \vec{x}_k + \vec{w}_k \Rightarrow \begin{pmatrix} p_{k+1} \\ v_{k+1} \end{pmatrix} = \begin{pmatrix} 1 & T \\ 0 & 1 \end{pmatrix} \begin{pmatrix} p_k \\ v_k \end{pmatrix} + \begin{pmatrix} \frac{T^2}{2} \\ T \end{pmatrix} a_k$$

T is time increment

$$\vec{z}_k = H_k \vec{x}_k + \vec{v}_k \Rightarrow \begin{pmatrix} z_k \\ 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} p_k \\ v_k \end{pmatrix} + \begin{pmatrix} b_k \\ 0 \end{pmatrix}$$

$$z_k = p_k + b_k$$

Initialization (here: position and velocity values are known with certainty)

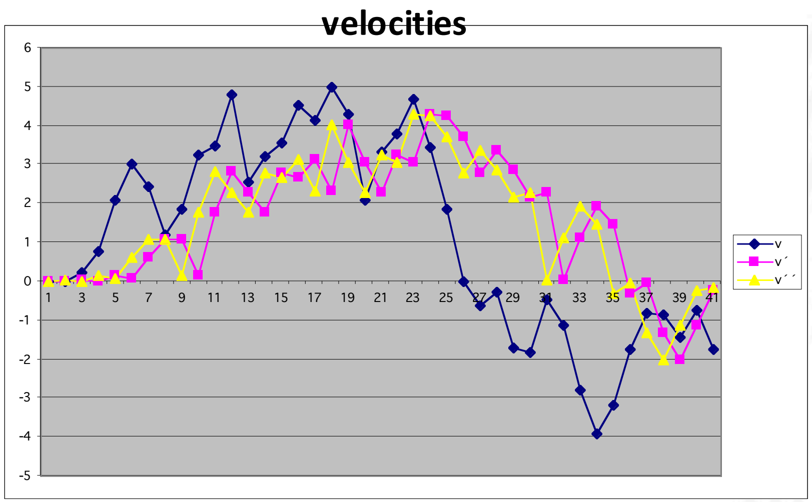
$$\vec{x}_0 = \begin{pmatrix} p_0 \\ v_0 \end{pmatrix} \quad P_0 = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} \quad K_0 = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} \quad \vec{x}_0'' = \begin{pmatrix} p_0 \\ v_0 \end{pmatrix} \quad P_0'' = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$$

First iteration

$$\vec{x}_1' = \begin{pmatrix} 1 & T \\ 0 & 1 \end{pmatrix} \begin{pmatrix} p_0 \\ v_0 \end{pmatrix} = \begin{pmatrix} p_0 + v_0 T \\ v_0 \end{pmatrix} \qquad P_1' = q^2 \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}$$

$$K_1 = \frac{q^2}{q^2 + r^2} \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \qquad \vec{x}_1'' = \begin{pmatrix} p_0 + v_0 T \\ v_0 \end{pmatrix} + \frac{q^2}{q^2 + r^2} \begin{pmatrix} z_1 - (p_0 + v_0 T) \\ 0 \end{pmatrix} \qquad P_1'' = \frac{q^2}{q^2 + 1} \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}$$

Diagrams for Kalman Filter Example I



$T = 1$ time step

$q = 1$ standard deviation of acceleration bursts

$r = 20$ standard deviation of position sensor

$p_0 = 0$ initial position

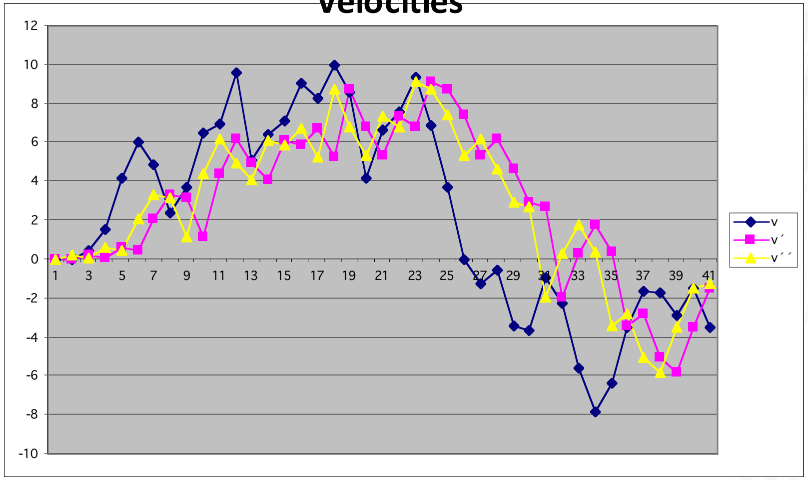
$v_0 = 0$ initial velocity

The standard deviation of the estimated position p is around 12 before observing $\vec{z}$ and around 10 after observing $\vec{z}$.

Diagrams for Kalman Filter Example II

positions

velocities



$T = 1$	time step
$q = 2$	standard deviation of acceleration bursts
$r = 20$	standard deviation of position sensor
$p_0 = 0$	initial position
$v_0 = 0$	initial velocity

The standard deviation of the estimated position p is around 15 before observing $\vec{z}$ and around 12 after observing $\vec{z}$.

Kalman Filter Demo

https://www.youtube.com/watch?v=sG-h5ONsj9s

Differential Approaches

Idea:

Instead of searching and tracking features, derive the change within the sequence by means of differential (flow) fields.

We destinguish between:

- local approaches,
  which derive the flow field only using local sequence information
- global approaches,
  which use global contraints to derive the flow field
- hybrid approaches,
  which combine the former two approaches

The correspondence problem does not need to be solved. But the dependency between flow field and motion has to be questioned!

Most approaches are based on the Optical Flow Constraint Equation!

Optical Flow Constraint Equation

Optical flow is the displacement field of surface elements of a scene during an incremental time interval dt ("velocity field").

Assumptions:

- Observed brightness is constant over time (no illumination changes)
- Nearby image points move similarly (velocity smoothness constraint)

For a continuous image $g(x, y, t)$ a linear Taylor series approximation gives

$$g(x+dx,\ y+dy,\ t+dt) \approx g(x,\ y,\ t) + g_x dx + g_y dy + g_t dt = 0 \quad \text{with: } \nabla^T g = \begin{pmatrix} g_x & g_y & g_t \end{pmatrix}$$

For motion without illumination change we have

$$g(x+dx,\ y+dy,\ t+dt) = g(x,\ y,\ t)$$

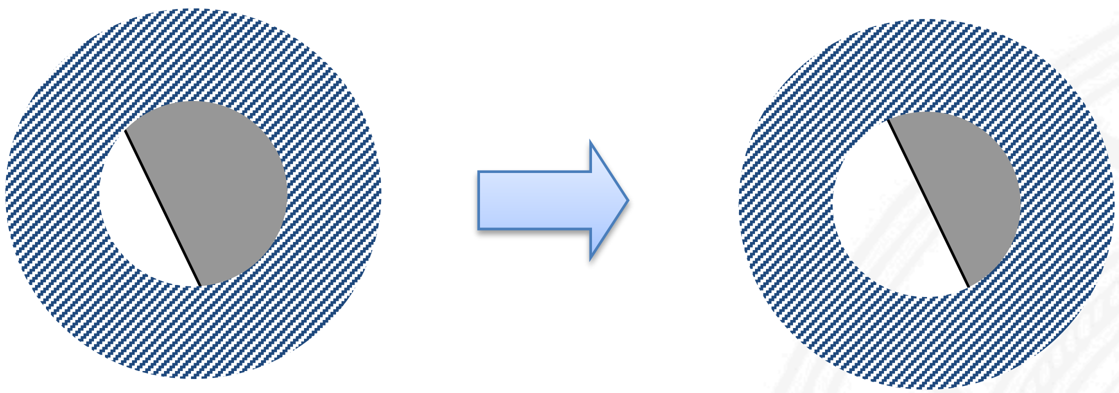
Hence $\frac{dx}{dt} g_x + \frac{dy}{dt} g_y = g_x u + g_y v = -g_t$

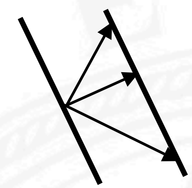
Optical Flow Constraint Equation (OFCE)
with: u, v velocity components

$g_x \approx \Delta g/\Delta x,\ \ g_y \approx \Delta g/\Delta y,\ \ g_t \approx \Delta g/\Delta t$ may be estimated from the spatial and temporal surround of a location (x, y), hence the optical flow constraint equation provides **one** equation for the **two** unknowns u and v.

Aperture Effect

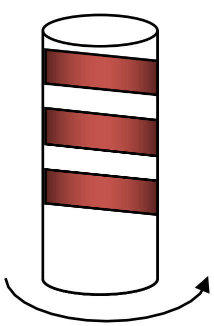
The optical flow constraint allows for ambiguous motion interpretations. This can be illustrated by the aperture effect.





In which direction has the edge moved?

Compare with the barber pole effect:
https://www.youtube.com/watch?v=EkmdGcPnRXQ



Due to the linear approximation of the image function, the velocity vector cannot be determined uniquely from a local neighbourhood.

Local Smoothness Constraint

One method to solve the Aperture Problem:
Assume a locally **equal** flow →**Lucas and Kanade approach**

$$\underset{(x,y)\in D_{ij}}{\forall} g_x(x,y,t)u + g_y(x,y,t)v = -g_t(x,y,t) \text{ with } D \text{ a local neighborhood around } (i,j)$$

Instead of one equation for two unknowns we now get $\mathrm{card}(D)$ equations! Hence the linear system of equations becomes overdetermined.

Solution via minimization of the squared error (MSE):

$$\begin{pmatrix} \sum_{(x,y)\in D_{ij}} g_x(x,y)^2 & \sum_{(x,y)\in D_{ij}} g_x(x,y)g_y(x,y) \\ \sum_{(x,y)\in D_{ij}} g_x(x,y)g_y(x,y) & \sum_{(x,y)\in D_{ij}} g_y(x,y)^2 \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix} = -\begin{pmatrix} \sum_{(x,y)\in D_{ij}} g_x(x,y)g_t(x,y) \\ \sum_{(x,y)\in D_{ij}} g_x(x,y)g_t(x,y) \end{pmatrix}$$

The above LSE can be computed efficiently by implementing the analytical solving procedure explicitly.

Note: More sophisticated methods apply local smoothness instead of local equality constraints → Structure Tensor approach

Global Smoothness Constraint

For dynamic scenes one can often assume that the velocity field changes smoothly in a spatial neighbourhood → **Horn and Schunck approach**:

- large objects
- translational motion
- observer motion, distant objects

Hence, as an additional constraint, one can minimize a smoothness error:

$$e_s = \iint \left(u_x^2 + u_y^2\right) + \left(v_x^2 + v_y^2\right) dx\, dy \qquad \text{with } u_x = \frac{\partial u}{\partial x}\ etc.$$

One also wants to minimize the error in the optical flow constraint equation:

$$e_c = \iint \left(g_x u + g_y v + g_t\right)^2 dx\, dy$$

Using a Lagrange multiplier λ, both constraints can be combined into an error functional, to be minimized by the calculus of variations:

$$e = e_c + \lambda e_s = \iint \left(g_x u + g_y v + g_t\right)^2 + \lambda\left(u_x^2 + u_y^2 + v_x^2 + v_y^2\right) dx\, dy$$

Optical Flow Algorithm

(Horn & Schunck 81)

The solution for optical flow with smoothness constraint is given in terms of a pair of partial differential equations:

$$2g_x(g_x u + g_y v + g_t) - 2\lambda u_{xx} - 2\lambda u_{yy} = 0$$

$$2g_y(g_x u + g_y v + g_t) - 2\lambda v_{xx} - 2\lambda v_{yy} = 0$$

The equations can be solved by a Gauss-Seidel iteration based on pairs of consecutive images (Horn & Schunck 81). At each iteration $n+1$ we get:

$$u^{n+1} = \bar{u}^n - g_x \frac{g_x \bar{u}^n + g_y \bar{v}^n + g_t}{\lambda + g_x^2 + g_y^2}$$

$$v^{n+1} = \bar{v}^n - g_y \frac{g_x \bar{u}^n + g_y \bar{v}^n + g_t}{\lambda + g_x^2 + g_y^2}$$ with $\bar{u}, \bar{v}$ mean velocities of the local neighborhood

Note:

- The partial derivatives are usually pre-computed for the image series.
- Initialization vectors may be non-zero.
- The iterations may be restricted by count or by threshold (Sonka):

$$e^k = \sum_i \sum_i \left(g_x(i,j)u^k(i,j) + g_y(i,j)v^k(i,j) + g_t(\mathrm{i,j})\right)^2 + \lambda\left(\left(u_x^k(i,j)\right)^2 + \left(u_y^k(i,j)\right)^2 + \left(v_x^k(i,j)\right)^2 + \left(v_y^k(i,j)\right)^2\right) < \varepsilon$$

Optical Flow Improvements

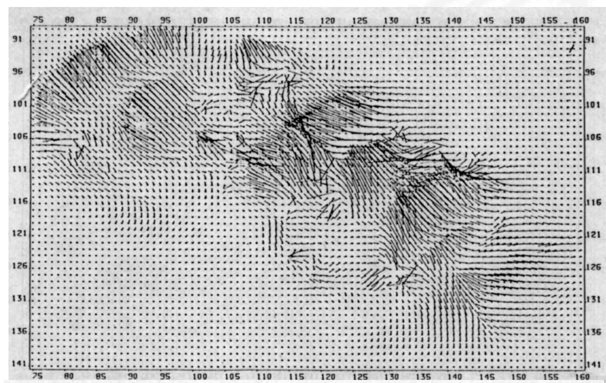
(from Nagel and Enkelmann 86)

Several improvements of the Horn & Schunck optical flow computation have been suggested. For example, Nagel (1983) introduced the "oriented smoothness constraint" which does not enforce smoothness across edges.

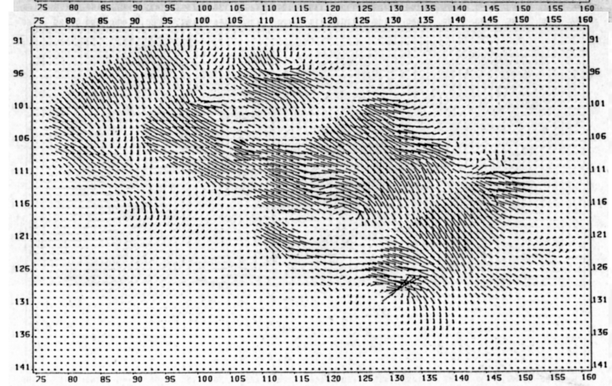
2 frames of the taxi sequence

frame 11

frame 12

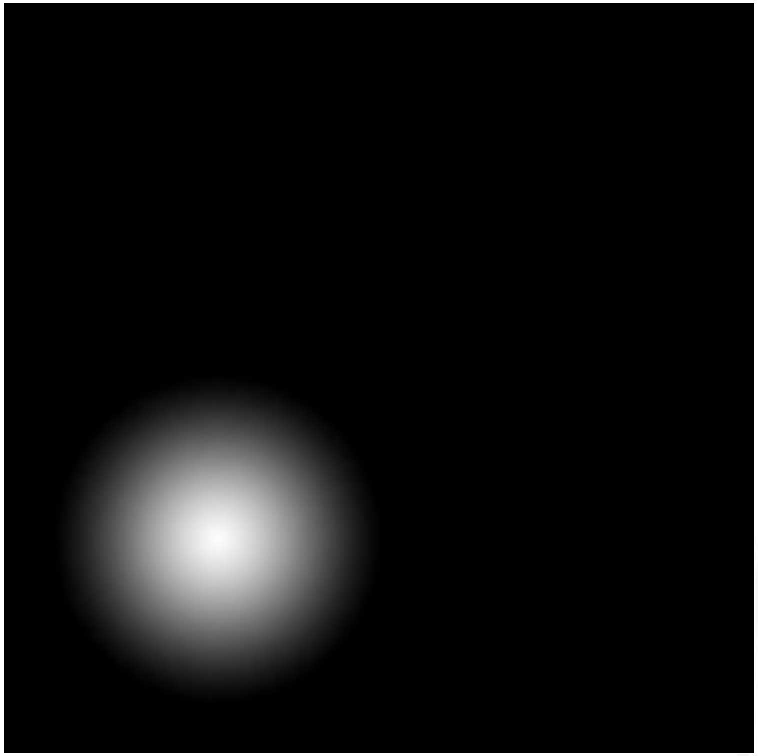


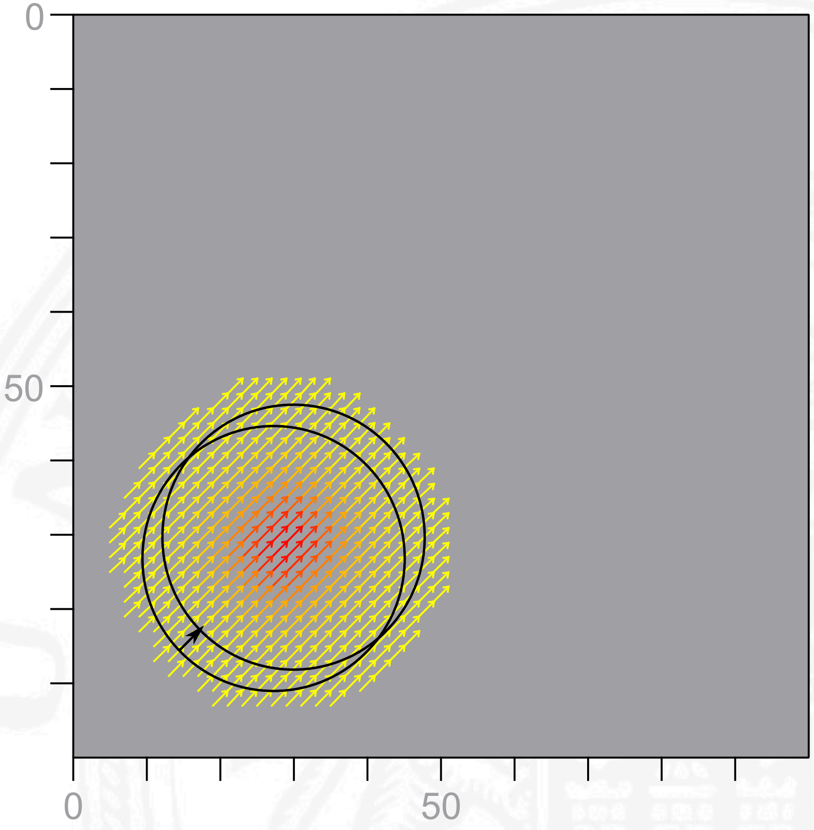
needle diagram of optical flow for taxi motion with isotropic smoothness constraint after 30 iterations



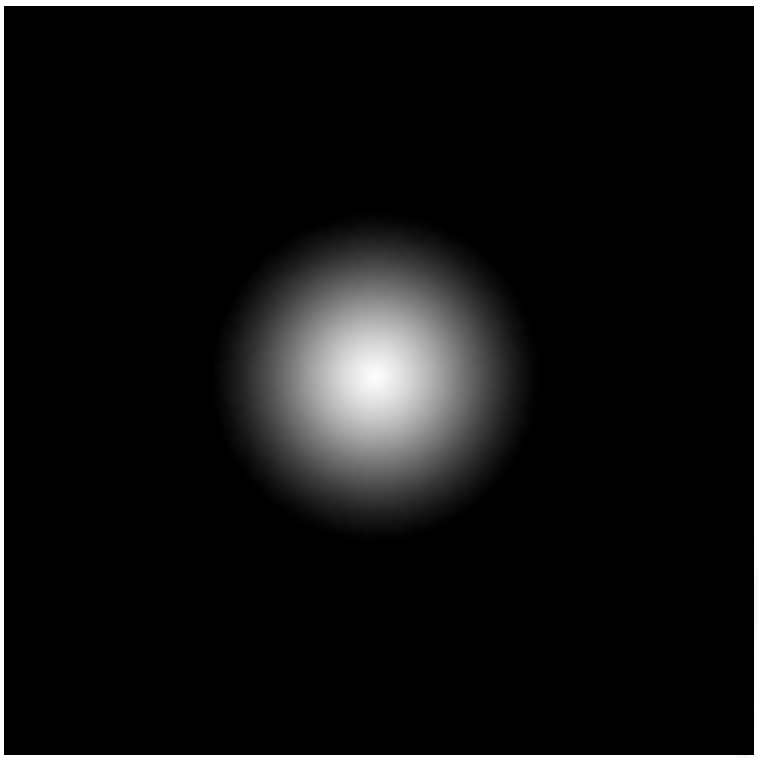
the same with oriented smoothness constraint

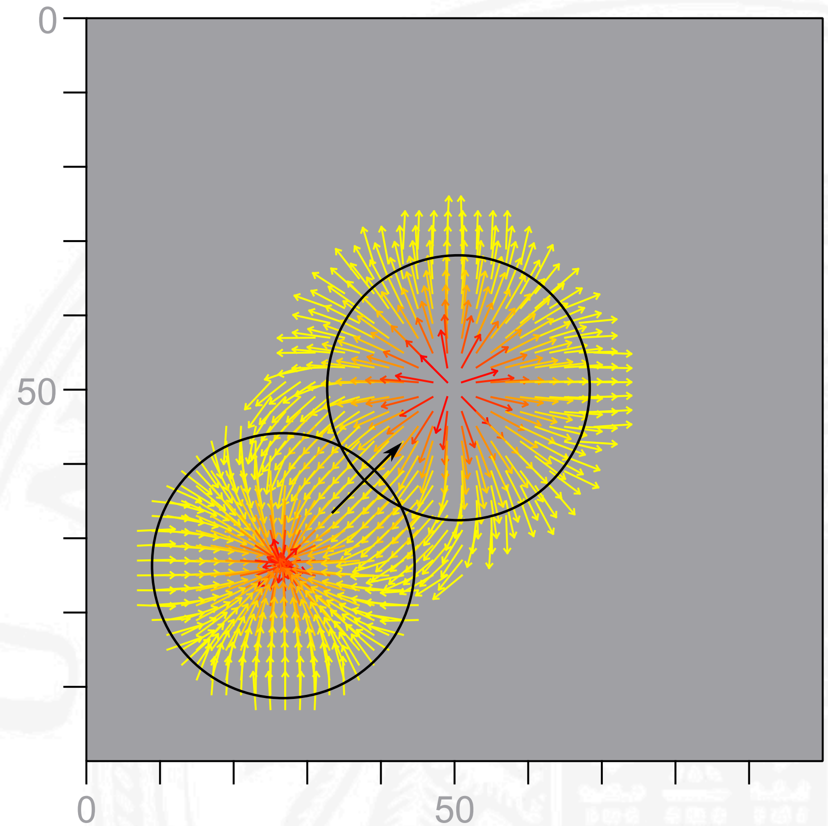
Small Spatio-temporal Distances



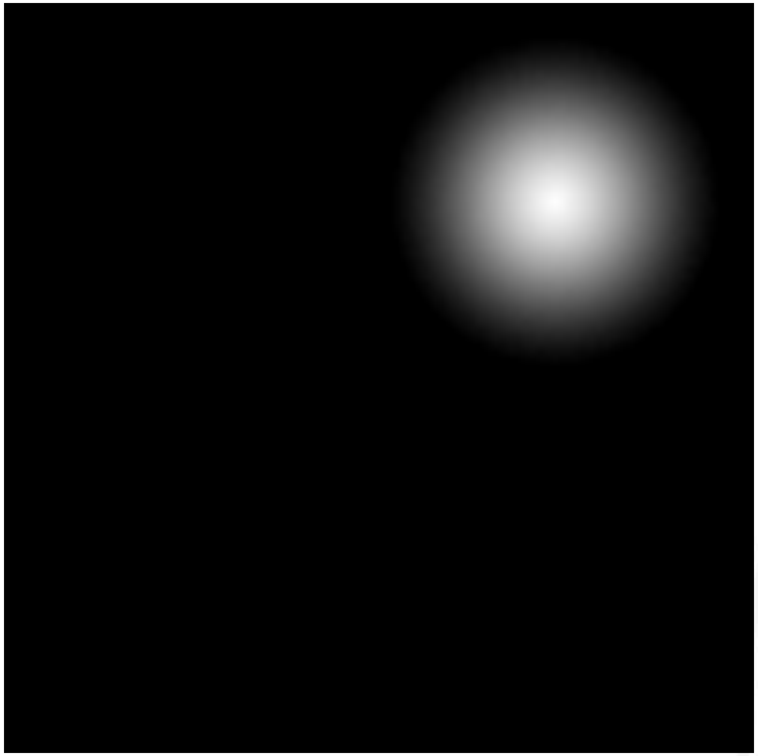


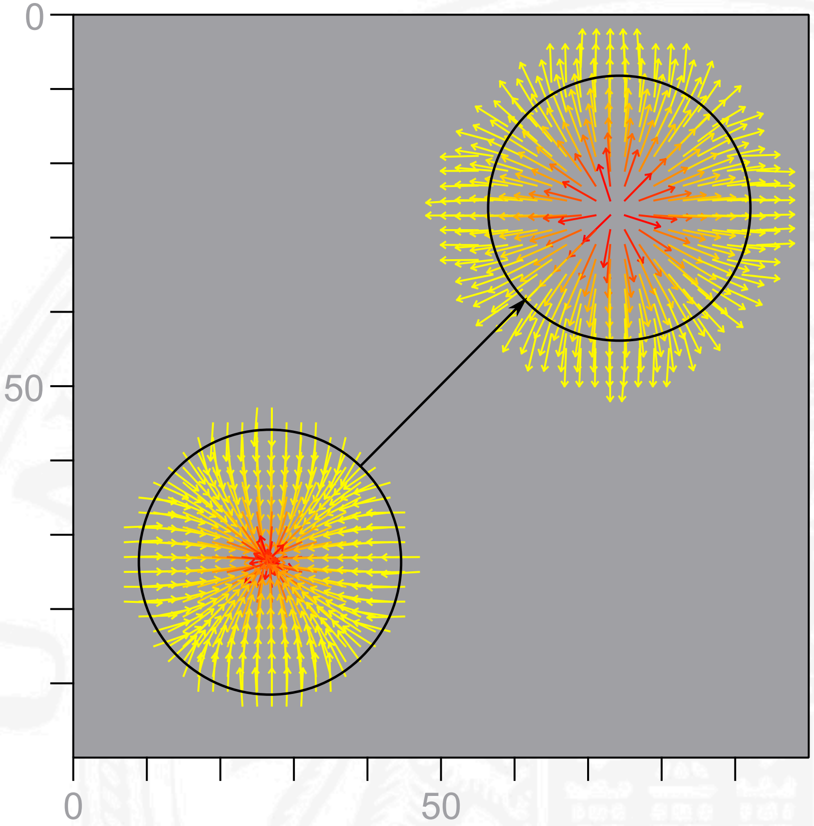
Larger Spatio-temporal Distances





Largest Spatio-temporal Distances





Optical Flow Oberservations

- For large spatio-temporal distances, the flow may not correspond to the motion.
- Typical solution approach:
  - use Image (resolution) pyramid to decompose motion at different scales
  - warp partial results and integrate the parts at the end.
- Observation:
  - Pyramids may not be applicable for non-linear motion decompostion
  - Inter-level warping may introduce new errors

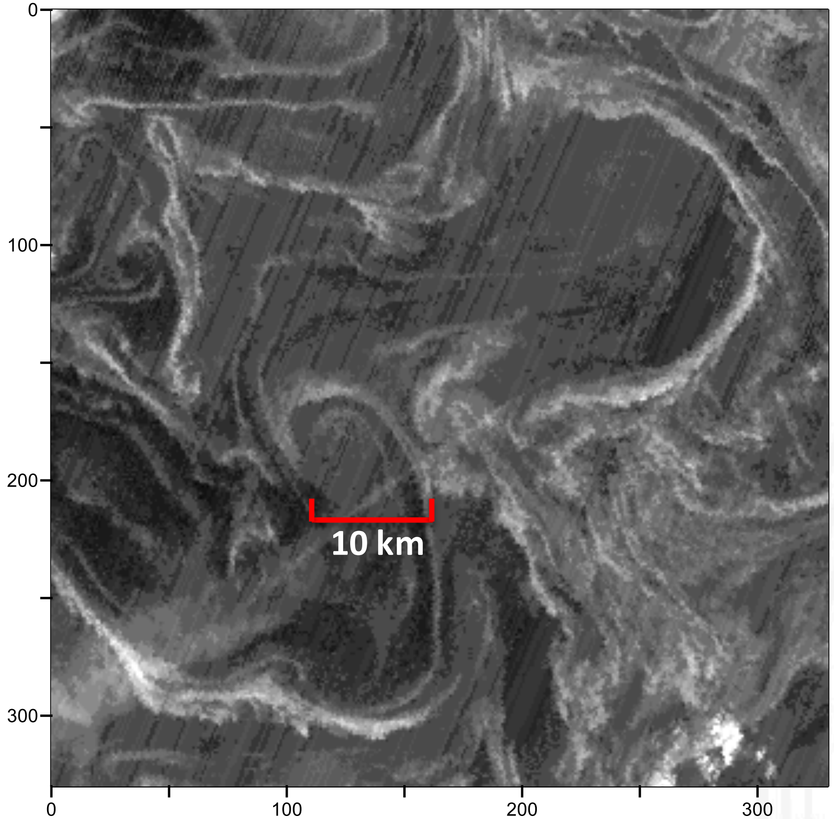
Optical Flow Improvements

(from Seppke 2013)

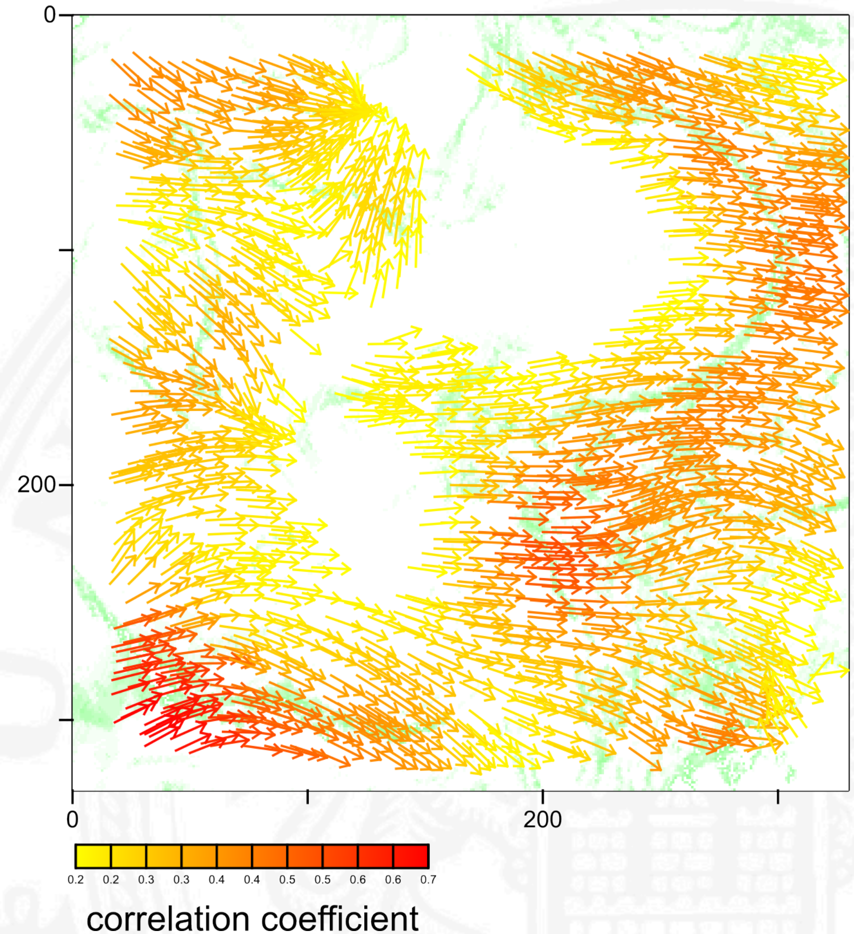
- To overcome the drawbacks, we propose the following solution:
  - Decompose motion into a (large) global and a (small) local motion part
  - Find a model for the local motion and apply e.g. automated registration methods to correct it.
  - Use the commonly known Optical Flow approaches to derive the local part of the motion
  - Combine both results for the final motion vectors
- Benefits:
  - Local/Global motion parts may provide a deeper insight into the processing underneath the motion.
  - Generic base instead of a new (over-)specialized algorithm
  - Approach is also applicable to enhance feature-based methods!

Flow Decomposition – Overall Flow

Deriving sea surface current fields from satellite images

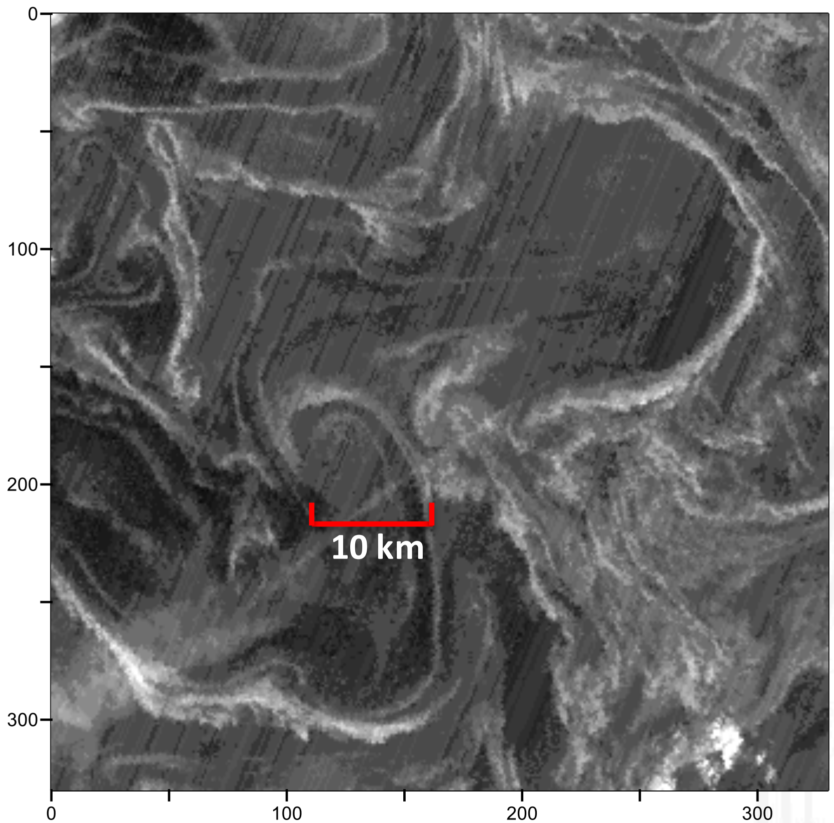


IRS-1C WiFS (Near Infrared)



Flow Decomposition – Local Flow Component

Deriving sea surface current fields from satellite images



IRS-1C WiFS (Near Infrared)