Image motion

Václav Hlaváč

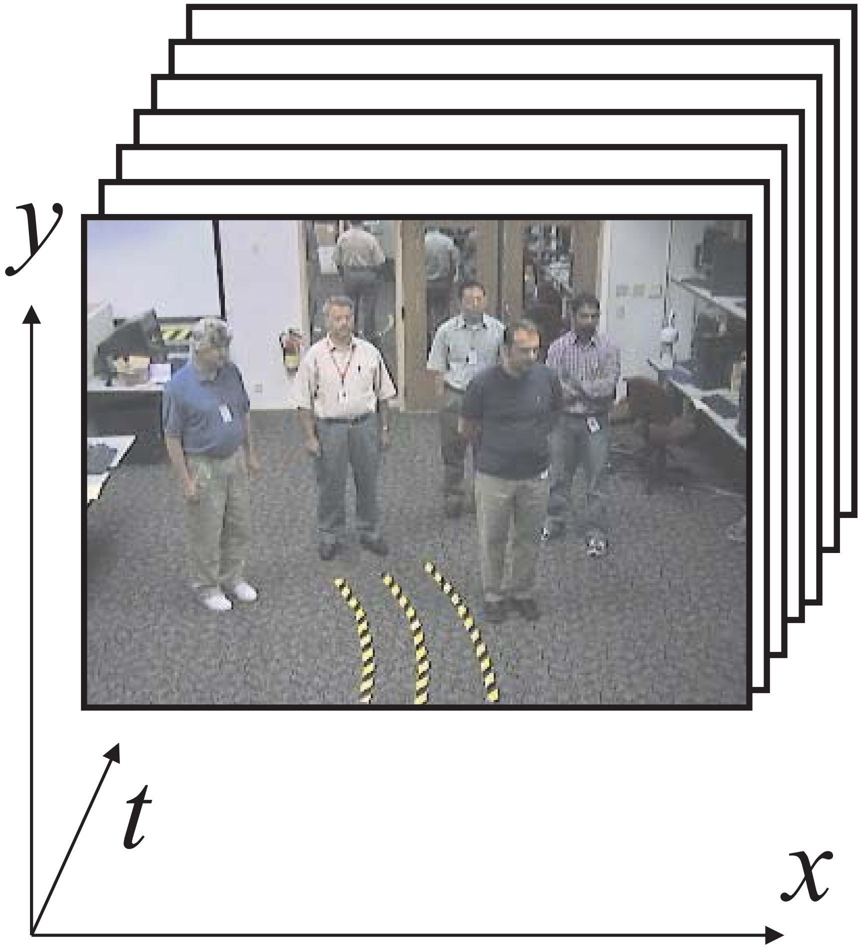
Czech Technical University in Prague, Faculty of Electrical Engineering
Department of Cybernetics, Center for Machine Perception
121 35 Prague 2, Karlovo nám. 13, Czech Republic

hlavac@fel.cvut.cz, http://cmp.felk.cvut.cz

Outline of the lecture:

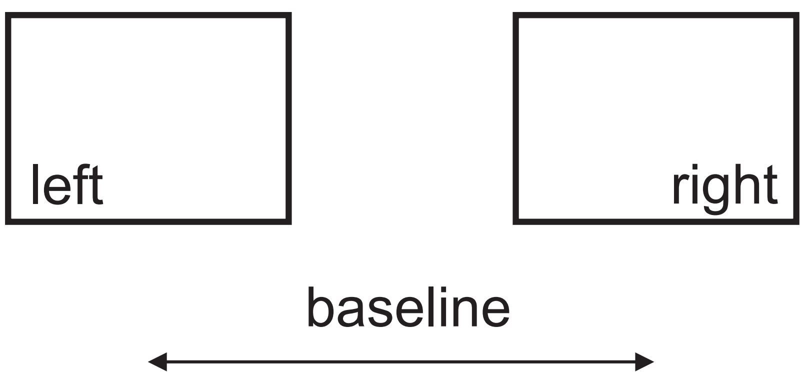
- Image motion analysis, task formulations.
- Motion field, apparent motion.
- Optic flow.
- Lucas-Kanade solution.
- Motion detection by frames differencing.
- Mogion detection, background models.

Dynamic scene analysis



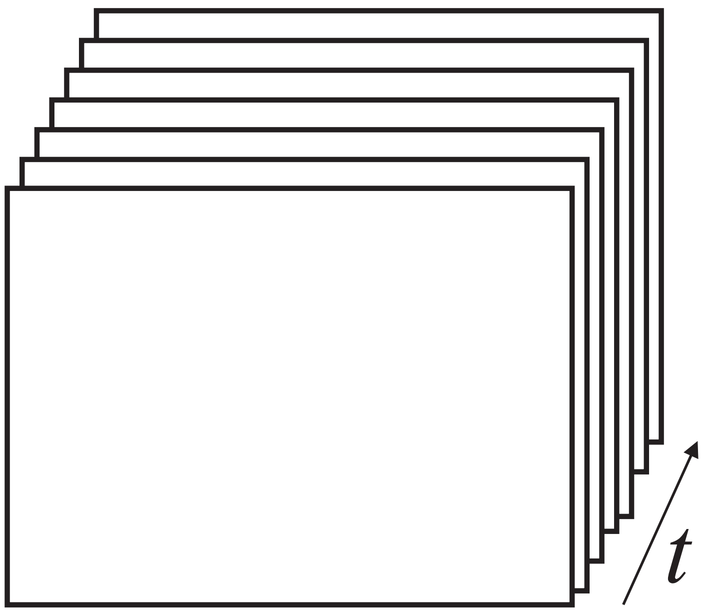
- The input to the dynamic scene analysis is a sequence of image frames $F(x, y, t)$ taken from the changing world.
- x, y are spatial coordinates. Frames are usually captured at fixed time intervals.
- t represents t-th frame in the sequence.

Motion analysis vs. stereo



Stereo

- The baseline is usually larger in stereo than in motion.
- Stereo images are captured at the same time.



Motion analysis

- Motion sequences ‘baselines’ are smaller than in stereo.
- The hope is that disparities are due to motion.

Camera, scene objects – static vs. moving

- Assuming that the scene illumination does not change, the image changes are due to relative motion between the scene objects and the camera (observer).
- There are three possibilities:
  - Stationary camera, moving objects.
  - Moving camera, stationary objects.
  - Moving camera, moving objects.

Motion analysis tasks

- Correspondence problem.

  Track corresponding pixels or interest points across frames.

- Segmentation problem.

  What are the regions of the image plane which correspond to different moving objects in the scene?

- Reconstruction problem, also called the shape from motion.

  Given a number of corresponding elements in the image plane of different video frames, and camera parameters, what can we say about the 3D motion and structure of the observed scene?

Involved tasks from the application point of view

- Motion detection. Often from a static camera. Common in surveillance systems. Often performed on the pixel level only (due to speed constraints).
- Object localization. Focuses attention to a region of interest in the image. Data reduction. Often only interest points found which are used later to solve the correspondence problem.
- Motion segmentation. Images are segmented into region corresponding to different moving objects.
- Three-dimensional shape from motion. Called also structure from motion. Similar problems as in stereo vision.
- Object tracking. A sparse set of features is often tracked, e.g., corner points.
- Pursuit. Consider two flies playing catch ...

Motion field

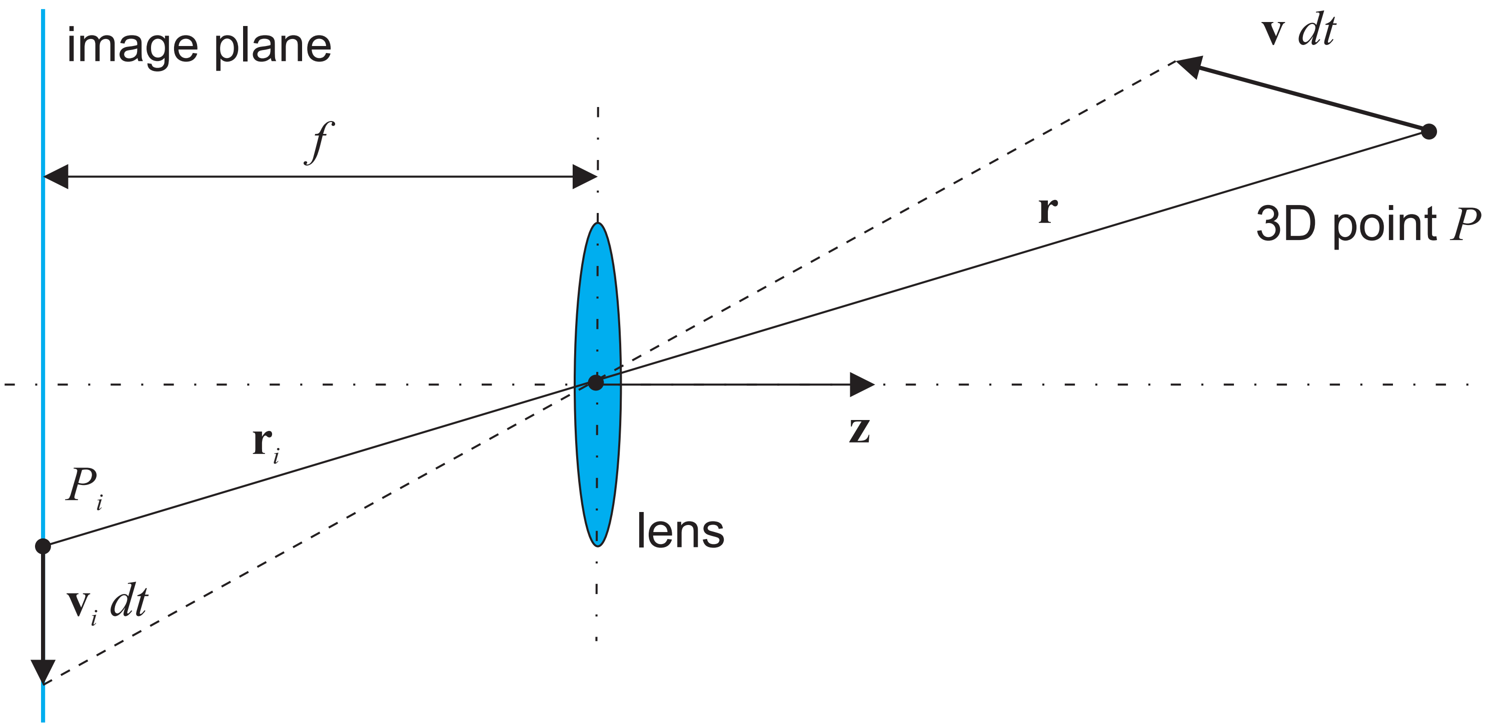
- Motion field is a 2D representation in the image plane of a (generally) 3D motion of points in the scene (typically on surfaces of objects).
- Each point is assigned a velocity vector corresponding to the motion direction and velocity magnitude:
  - In the 3D scene: vector $\mathbf{v}$.
  - In the image plane after the projection: vector $\mathbf{v}_i$.

- *Note: Inherent problem – the relation 3D scene $\rightarrow$ 2D image often needs image interpretation (i.e., segmentation, knowledge what objects are). This information is not available in general in the 'intrinsic image sequence'.*

Role of matching in the image plane

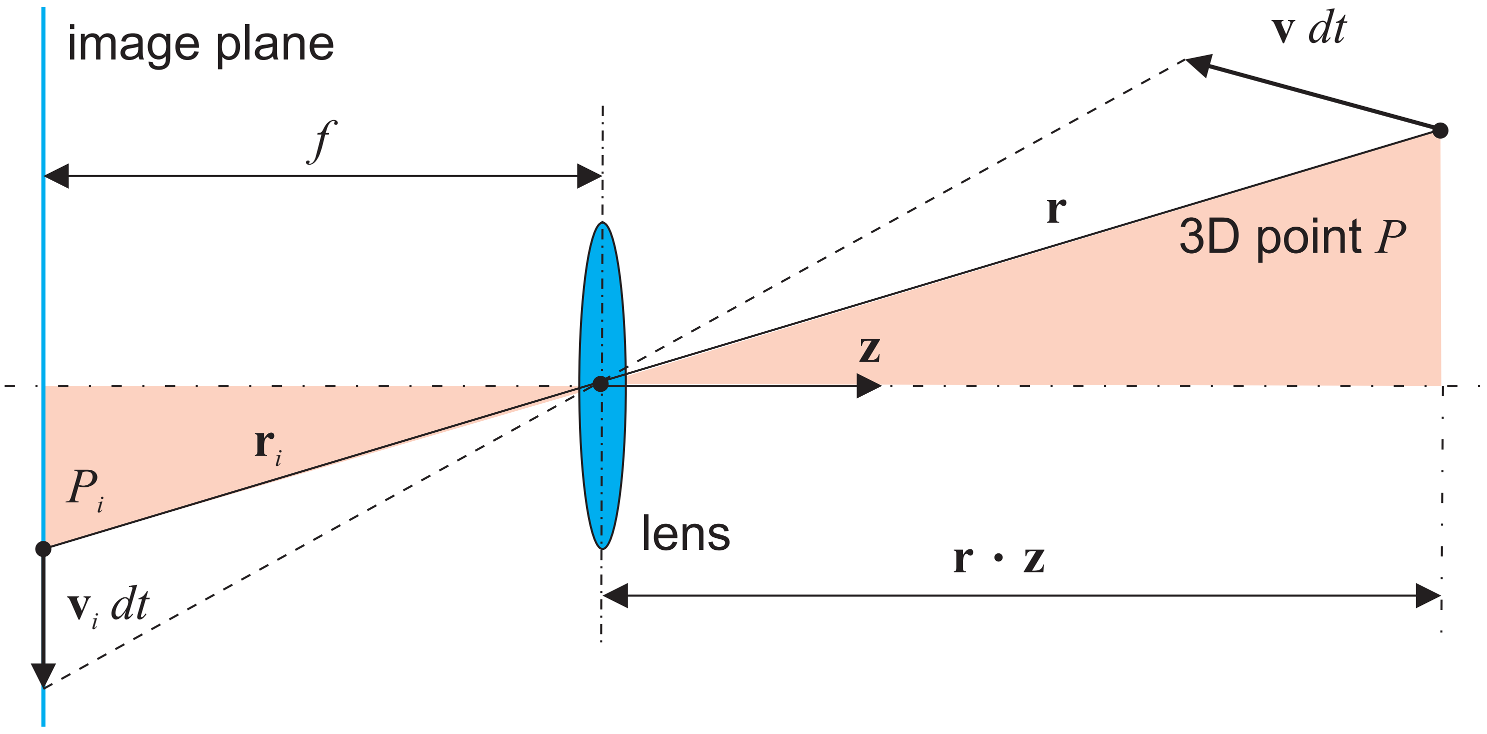
- Match region of image to region of image (as in stereo). However, in motion:
  - Motion (baseline in stereo terminology) is small.
  - The epipolar constraint is unknown.
- Estimating motion field, differently formulated tasks:
  1. Matching of objects. Interpretation needed. A sparse motion field.
  2. Matching of interest points. A bit denser motion field. A more difficult interpretation. Problems with repetitive structures.
  3. Optic flow, a differential technique matching intensities on a pixel level, exploring spatial and time variations. Ideally, a dense motion field. Problems for textureless parts of images due to aperture problem (to be explained).

Motion field, derivation



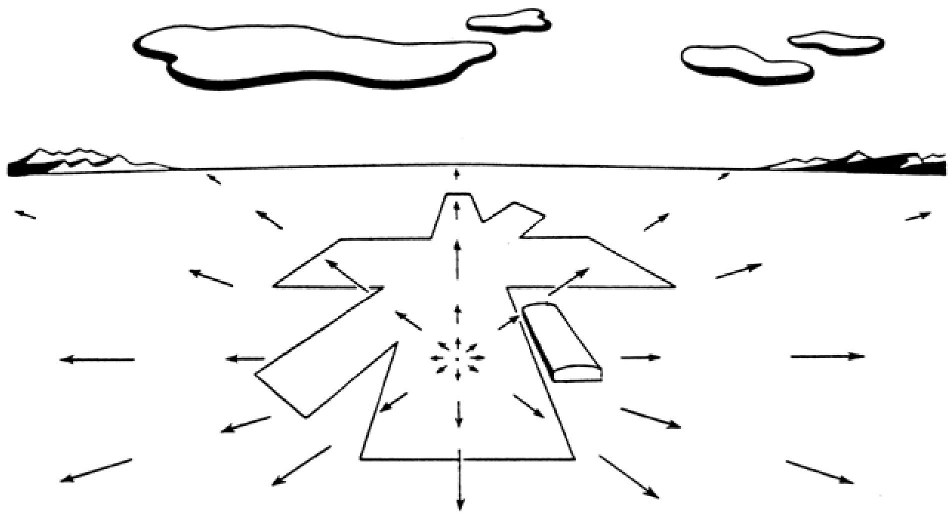
- $\mathbf{z}$ – unit vector in the axis z direction.
- f – focal length of a lens.
- $\mathbf{v}_i = \frac{\mathrm{d}\,\mathbf{r}_i}{\mathrm{d}\,t}$ – velocity in the image plane.
- $\mathbf{v} = \frac{\mathrm{d}\,\mathbf{r}}{\mathrm{d}\,t}$ – velocity in the 3D space.

Motion field, derivation 2

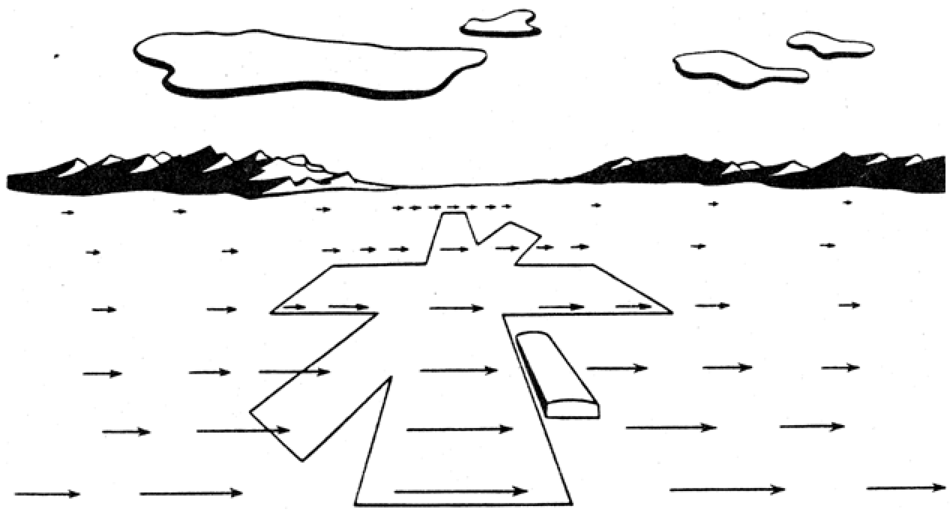


- Similar triangles $\frac{\mathbf{r}_i}{f} = \frac{\mathbf{r}}{\mathbf{r}\cdot\mathbf{z}}$
- Temporal derivative $\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\mathbf{r}_i}{f}\right) = \frac{\mathbf{v}_i}{f} = \frac{(\mathbf{r}\cdot\mathbf{z})\cdot\mathbf{v} - (\mathbf{r}\cdot\mathbf{z})\cdot\mathbf{r}}{(\mathbf{r}\cdot\mathbf{z})^2} = \frac{(\mathbf{r}\times\mathbf{v})\times\mathbf{z}}{(\mathbf{r}\cdot\mathbf{z})^2}$

Example of motion fields 1



(a)



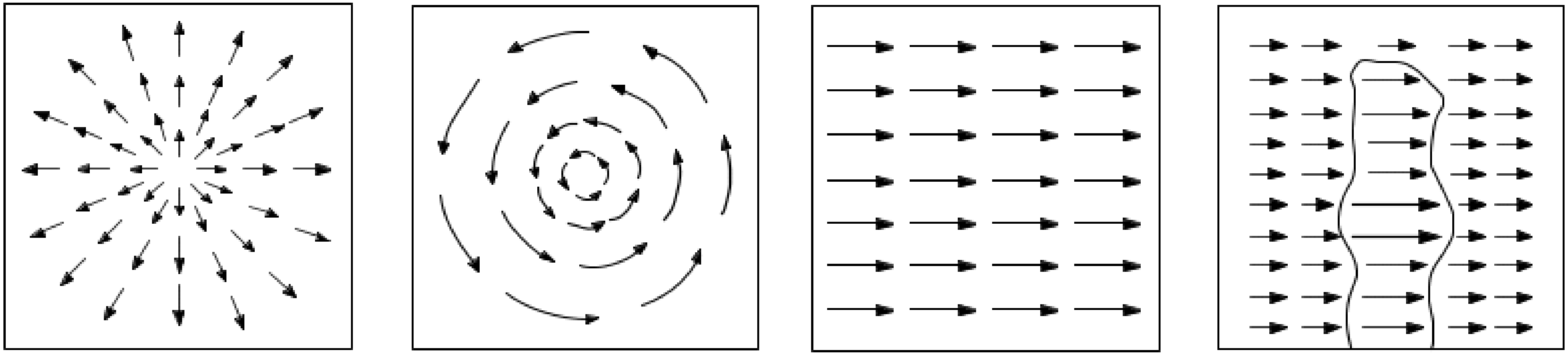
(b)

Motion field of a pilot

1. looking straight ahead while approaching a fixed point on a landing strip.
2. looking to the right in the level flight.

Courtesy: Cordelia Fernmüller, U of Maryland

Example of motion fields 2



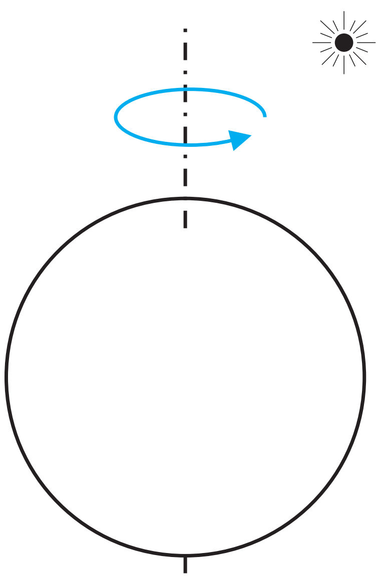
(a) (b) (c) (d)

1. Translation perpendicular to a surface.
2. Rotation about axis perpendicular to image plane.
3. Translation parallel to a surface at a constant distance.
4. Translation parallel to an obstacle in front of a more distant background.

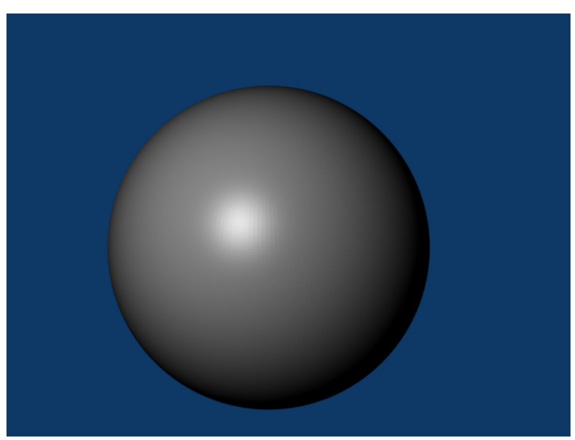
Courtesy: Cordelia Fernmüller, U of Maryland

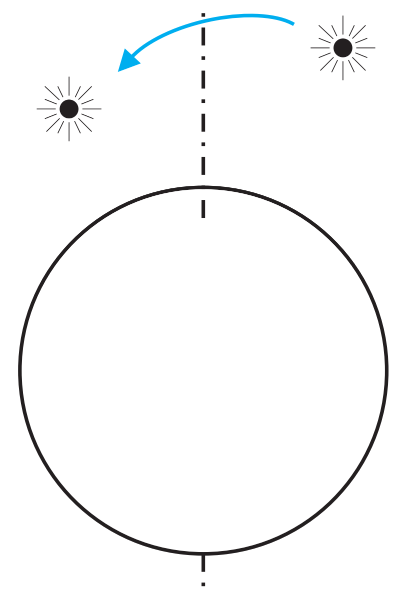
Inherent problem = apparent motion

- The image sequence informs only about intensity changes which are not necessarily related to the motion of points in the 3D scene.
- This is the apparent motion only, see B. Horn's counterexample below:



Sphere rotates.
Illuminant is still.





Sphere is still.
Illuminant moves.

Optic flow, where do pixels move to?

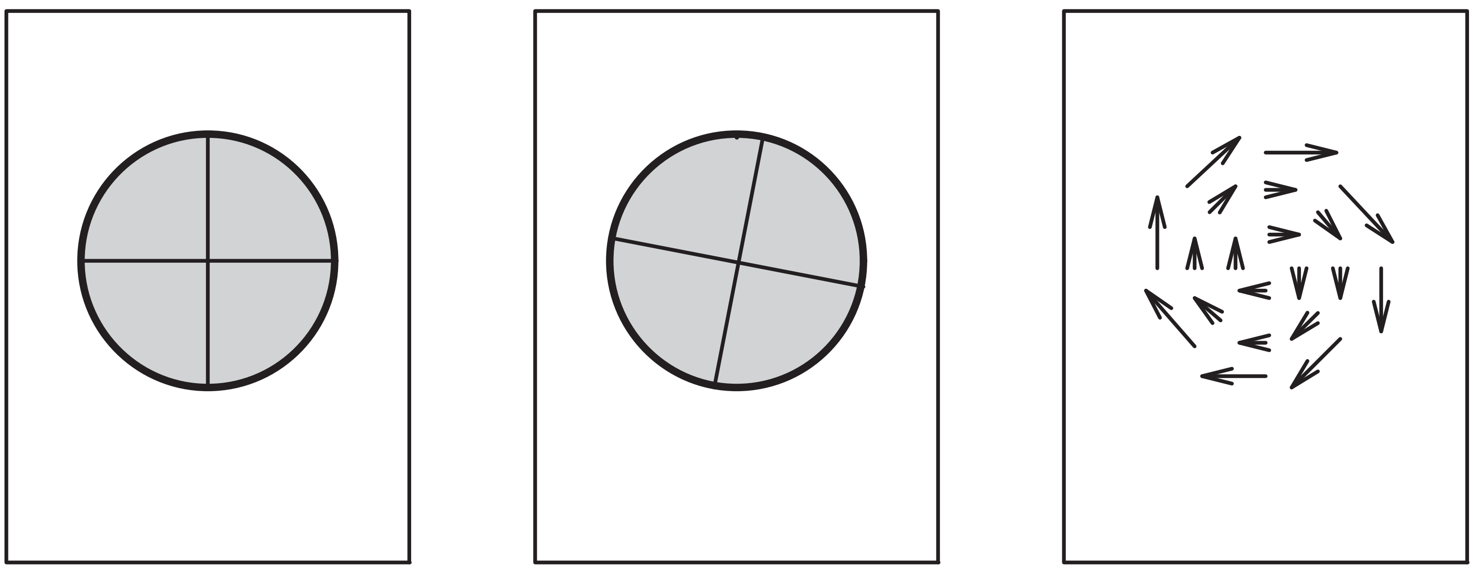
The optic flow problem can be inherently very difficult even if there is a causal relation between intensity changes and motion field, for example:

A birch in the wind.

Idea: A presentation of M. Pollefeys, ETH Zürich. Video: V. Hlavac.

Optic flow, initial problem formulation

- Optic flow = apparent motion of the same (similar) intensity patterns.
- Ideally, the optic flow can approximate projections of the three-dimensional motion (i.e., velocity) vectors onto the image plane.
- Hence, the optic flow is estimated instead of the motion field (since the latter is unobservable in general).



Time t_1 Time t_2 Optic flow

Optic flow

- Optic flow in an approximation to the motion field which explores spatial and temporal changes of the intensity in a temporal sequence of images.
- The approximation error is low at pixels with the high spatial gradient of the intensity (under some additional simplifying assumptions).
- Ideally, the optical flow corresponds to the projection of 3D motion vectors on some highly textured surfaces.
- Optic flow is sensitive to the aperture problem (will be explained shortly), illumination changes and motion of unimportant objects (e.g., shadows).
- Optic flow computation is usually a first step followed by a subsequent higher level processing.

Image intensity, constancy assumption

- $f(x, y, t)$ is the image intensity at a location x, y at a time t.
- Intensity constancy assumption $f(x, y, t) = f(x + \mathrm{d}x, y + \mathrm{d}y, t + \mathrm{d}t)$.
- Taylor series approximation of order 1,

$$f(x, y, t) = f(x, y, t) + \frac{\partial f}{\partial x}\mathrm{d}x + \frac{\partial f}{\partial y}\mathrm{d}y + \frac{\partial f}{\partial t}\mathrm{d}t + \textit{approx. error}\ . \tag{1}$$

- Expected no change of intensity f implies the total derivative $\frac{\mathrm{d}f}{\mathrm{d}t} = 0$, which can be obtained by dividing equation (1) by $\mathrm{d}t$. It writes as

$$\frac{\mathrm{d}f(x, y, t)}{\mathrm{d}t} \approx \frac{\partial f}{\partial x}\frac{\mathrm{d}x}{\mathrm{d}t} + \frac{\partial f}{\partial y}\frac{\mathrm{d}y}{\mathrm{d}t} + \frac{\partial f}{\partial t} = 0\ .$$

- The $\frac{\partial f}{\partial t}$ is the derivative of the intensity across frames.

Optic flow constraint
image intensity constancy equation

- Rewriting from the previous page,

$$\frac{\partial f}{\partial x}\frac{\mathrm{d}x}{\mathrm{d}t}+\frac{\partial f}{\partial y}\frac{\mathrm{d}y}{\mathrm{d}t}+\frac{\partial f}{\partial t}=0\,;$$

- Introducing optic flow vector $[u, v] = \left[\frac{\mathrm{d}x}{\mathrm{d}t}, \frac{\mathrm{d}y}{\mathrm{d}t}\right]$;
- Provides image intensity constraint equation

$$\frac{\partial f}{\partial x}u+\frac{\partial f}{\partial y}v+\frac{\partial f}{\partial t}=0\,,$$

which is one equation of two unknowns u, v.

- Consequently, the optic flow calculation from image intensities changes is underconstraint and generates so called aperture problem.

Optic flow constraint, matrix form

- The image intensity constancy equation at a single image pixel x, y writes as

$$\frac{\partial f(x,y)}{\partial x} u + \frac{\partial f(x,y)}{\partial y} v + \frac{\partial f(x,y)}{\partial t} = 0\,.$$

- The previous equation can be rewritten into a matrix form,

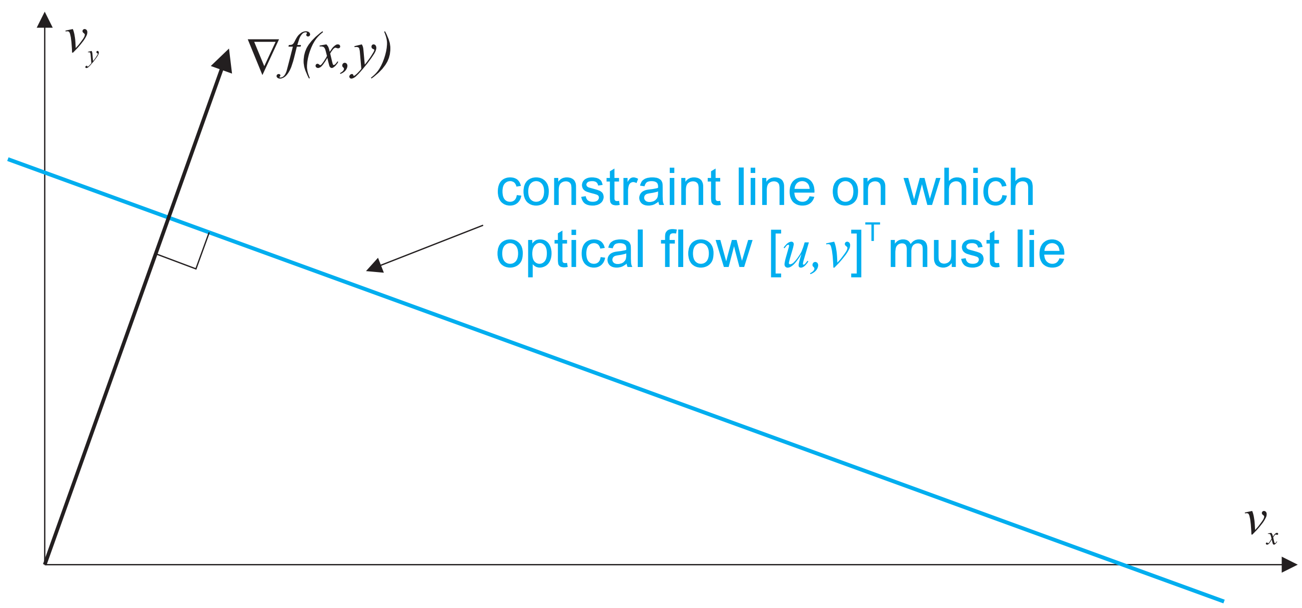
$$\left[\frac{\partial f(x,y)}{\partial x}, \frac{\partial f(x,y)}{\partial y}\right] \cdot \begin{bmatrix} u \\ v \end{bmatrix} + \frac{\partial f(x,y)}{\partial t} = 0\,.$$

- After introducing the spatial gradient within the frame $\nabla f(x,y) = \left[\ \frac{\partial f(x,y)}{\partial x}, \frac{\partial f(x,y)}{\partial y}\ \right]$,
the optic flow constraint equation writes as

$$\nabla f(x,y) \cdot \begin{bmatrix} u \\ v \end{bmatrix} + \frac{\partial f(x,y)}{\partial t} = 0\,.$$

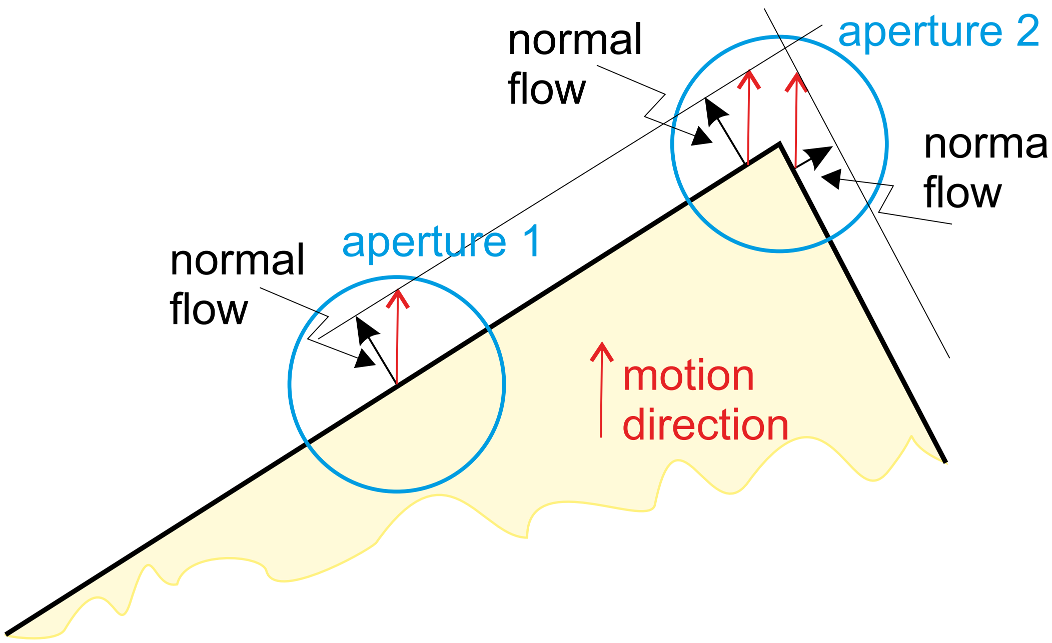
The aperture problem geometric interpretation

- The optic flow constraint equation $\nabla f(x,y) \cdot \begin{bmatrix} u \\ v \end{bmatrix} + \frac{\partial f(x,y)}{\partial t} = 0$ is the equation of a straight line in the velocity space v_x, v_y.



- The optic flow velocity has to lie in the straight line perpendicular to the direction of the intensity gradient $\nabla f(x,y)$ locally at each point x, y.
- Nothing is known about optical flow in the direction perpendicular to the straight constraint line.

Aperture problem, the illustration



Solving the optic flow equation

- Optic flow estimation is an ill-posed problem.
- No correspondence can be established at occlusions (covered/uncovered regions).
- Due to the aperture problem, the optical flow provides only one constraint equation for two independent variables u, v.

  Consequently, number of unknown variables is twice number of pixels.
- Additional assumptions or modeling is needed.
  - Parametric models of motion: translational, affine (planar object, orthography), perspective, bilinear.
  - Non-parametric models aiming at dense motion estimation: smoothness or uniformity constraints.

Spatial motion models

Assume a parametric motion $\mathbf{u} = [u, v]^\top$ at the pixel $[x, y]^\top$.

- Translational (2 parameters a_x, a_y): $\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} a_x \\ a_y \end{bmatrix}$
- Affine (6 parameters $a_1, \ldots, a_6$): $\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} a_1 & a_2 \\ a_3 & a_4 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} a_5 \\ a_6 \end{bmatrix}$
- Perspective (8 parameters) . . .
- Bilinear (8 parameters) . . .

Translational model, SSD solution

- Registration of images $f(x, y, t)$ and $f(x, y, t+1) = f(x + a_x, y + a_y, t)$.
- Optimization task, minimization of the sum of square differences (SSD) criterion:

$$E(a_x, a_y) = \sum_{x,y} \left(f(x + a_x, y + a_y, t) - f(x, y, t+1)\right)^2 \ .$$

- Simple SSD algorithm:
  - For each offset a_x, a_y calculate $E(a_x, a_y)$,
  - Select such a_x, a_y which minimizes $E(a_x, a_y)$.
- Problems: inefficient calculation, no subpixel accuracy.

Lucas-Kanade method 1981 (1)

- Compensates missing equations due to aperture problem by assuming that optic flow is smooth locally.
- This is implemented as a constant flow requirement over an image patch (typically 5×5 window W).

$$0 = \frac{\partial f(\mathbf{p}_i)}{\partial x} u + \frac{\partial f(\mathbf{p}_i)}{\partial y} v + \frac{\partial f(\mathbf{p}_i)}{\partial t} .$$

- The constraint provides 25 linear equations per pixel $\mathbf{p}_i = [x_i, y_i]^\top$, $i = 1 \dots 25$ in the window W.

$$\begin{bmatrix} \frac{\partial f(\mathbf{p}_1)}{\partial x}, & \frac{\partial f(\mathbf{p}_1)}{\partial y} \\ \frac{\partial f(\mathbf{p}_2)}{\partial x}, & \frac{\partial f(\mathbf{p}_2)}{\partial y} \\ \vdots, & \vdots \\ \frac{\partial f(\mathbf{p}_{25})}{\partial x}, & \frac{\partial f(\mathbf{p}_{25})}{\partial y} \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = - \begin{bmatrix} \frac{\partial f(\mathbf{p}_1)}{\partial t} \\ \frac{\partial f(\mathbf{p}_2)}{\partial t} \\ \vdots \\ \frac{\partial f(\mathbf{p}_{25})}{\partial t} \end{bmatrix} .$$

Written in matrix form: $A\mathbf{u} = -\mathbf{b}$.

Lucas-Kanade method (2)

- The goal is to minimize optimization criterion $E(u,v)$ over a window W,

$$E(u,v) = \sum_{\mathbf{p}\in W} \left(\frac{\partial f}{\partial x} u + \frac{\partial f}{\partial y} v + \frac{\partial f}{\partial t} \right)^2 = \| A\mathbf{u} = -\mathbf{b} \| \,.$$

- The least squares solution to the overdetermined system of linear equations $A\mathbf{u} + \mathbf{b} = 0$ is sought

$$\underbrace{A^\top A}_{2\times 2} \underbrace{\mathbf{u}}_{2\times 1} = \underbrace{A^\top \mathbf{b}}_{2\times 1} \quad \Rightarrow \quad \text{optic flow vector } \mathbf{u} = \left(A^\top A\right)^{-1} A^\top \mathbf{b}\,.$$

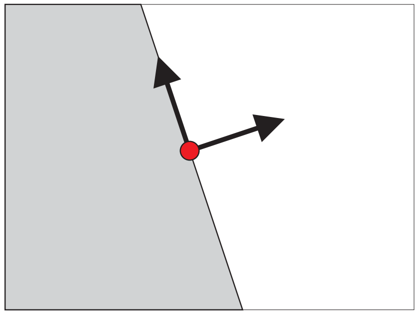
$$A^\top A \text{ (which is a Hessian) } = \begin{bmatrix} \frac{\partial f^2(\mathbf{p})}{\partial x^2}, & \frac{\partial f^2(\mathbf{p})}{\partial x\, \partial y} \\ \frac{\partial f^2(\mathbf{p})}{\partial x\, \partial y}, & \frac{\partial f^2(\mathbf{p})}{\partial y^2} \end{bmatrix} .$$

Notice: that the same Hessian was used in Harris corner detector.

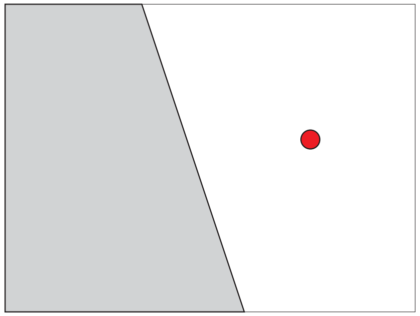
Behavior of Lucas-Kanade method

Matrix $A^\top A$ has to be invertible $\Rightarrow$ full rank 2
$\equiv$ non-zero eigenvalues.

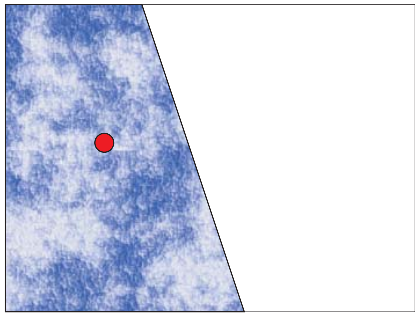
On the edge: $A^\top A$ becomes singular, has rank 1 because there is no intensity change along edge.



In a homogeneous region: $A^\top A$ has rank 0, because $\left[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right] \approx 0$.

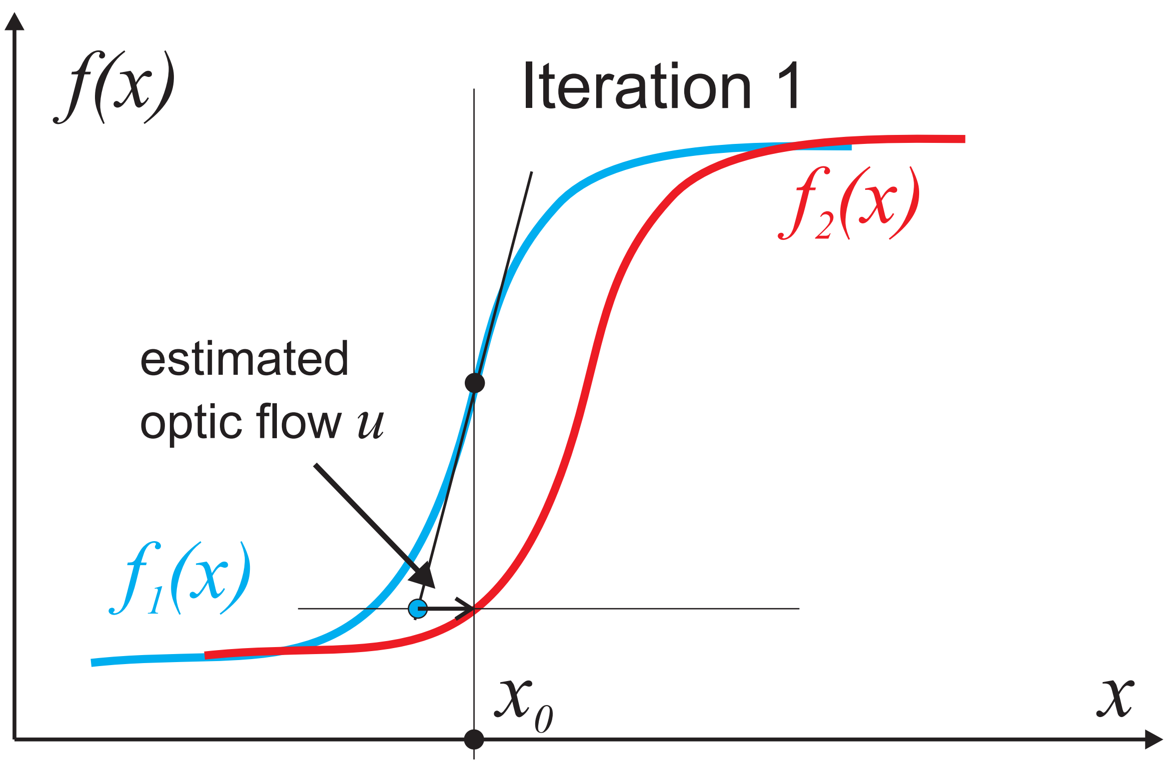


In a highly textured region: $A^\top A$ is regular, both eigenvalues are (safely) non-zero.



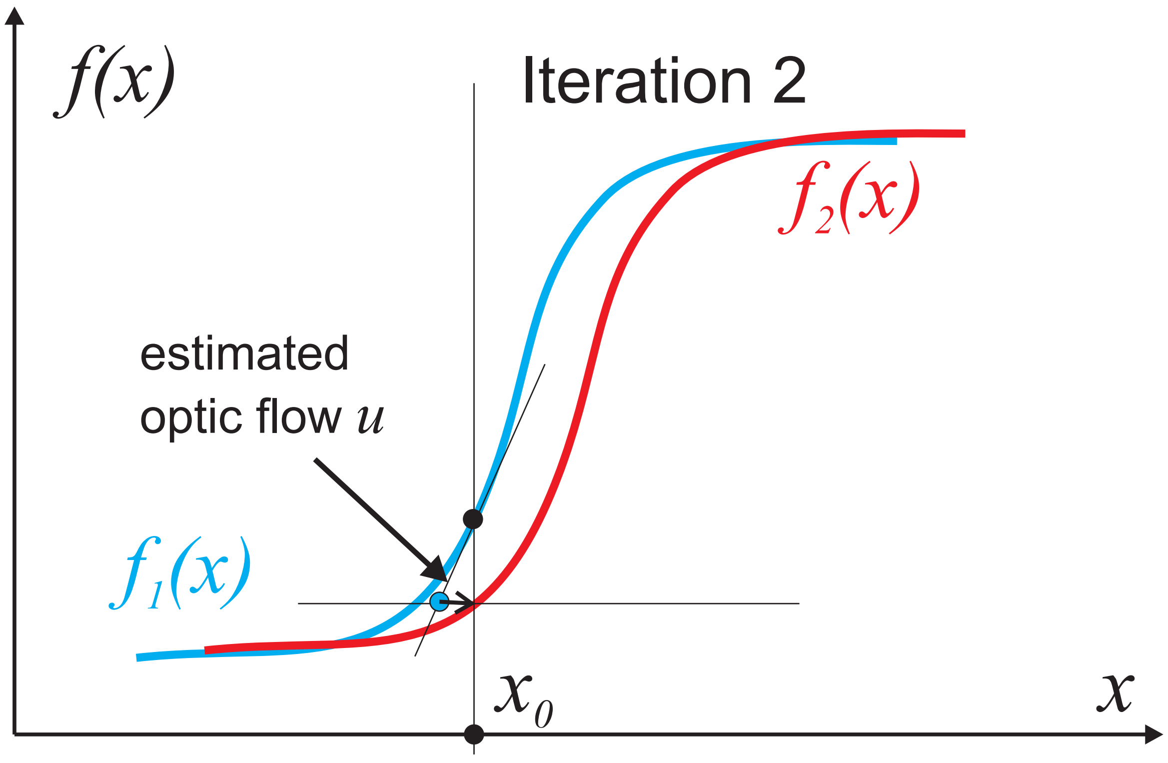
Iterative matching refinement (1)

- Estimate velocity at each pixel using one iteration of Lucas-Kanade flow field, $\mathbf{u} = (A^\top A)\, A^\top \mathrm{b}$.
- Warp older image frame to the newer one using the estimated flow field $\mathbf{u}$ (actually, it is not easy in practice).
- Refine the estimate by repeating the iterations.



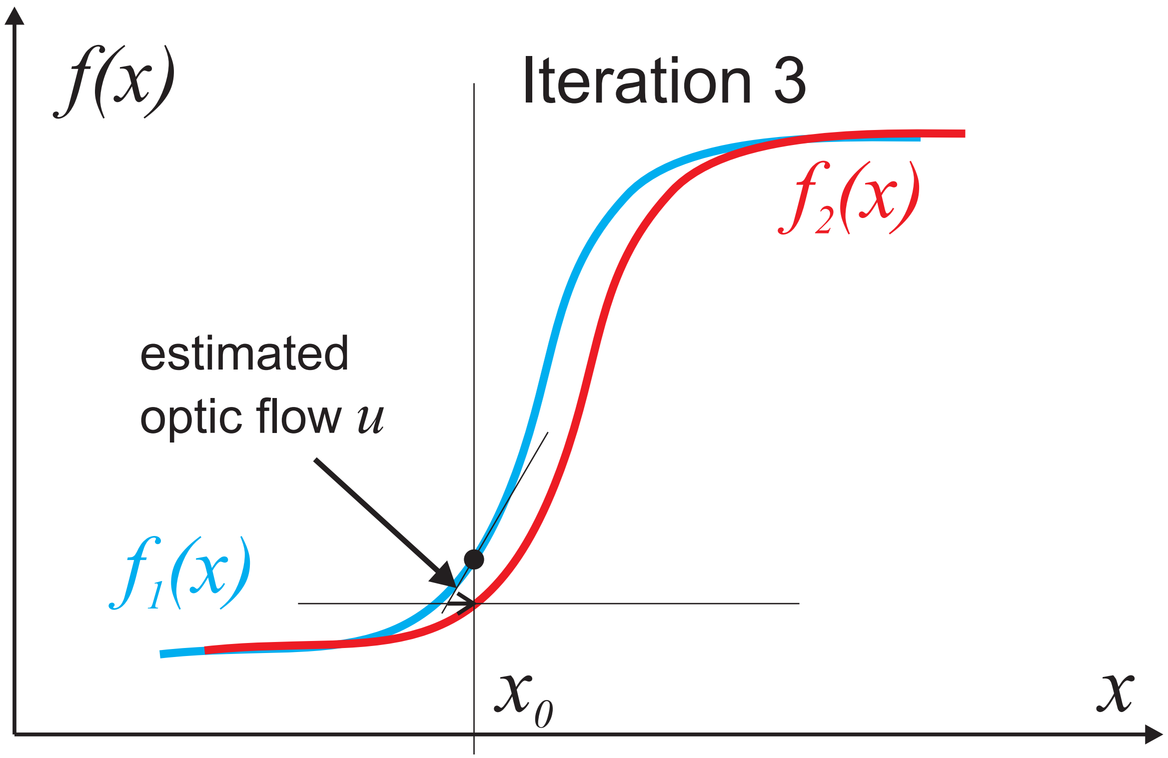
Iterative matching refinement (2)

- Estimate velocity at each pixel using one iteration of Lucas-Kanade flow field, $\mathbf{u} = (A^\top A)\, A^\top \mathrm{b}$.
- Warp older image frame to the newer one using the estimated flow field $\mathbf{u}$ (actually, it is not easy in practice).
- Refine the estimate by repeating the iterations.



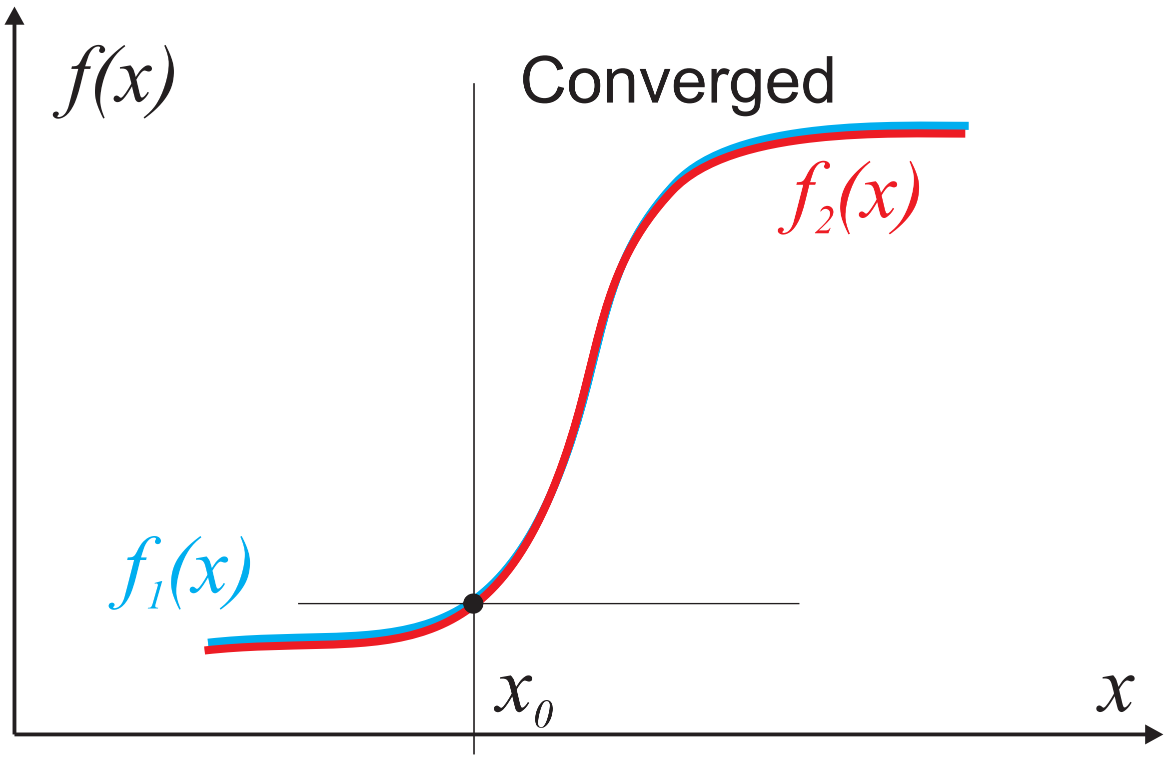
Iterative matching refinement (3)

- Estimate velocity at each pixel using one iteration of Lucas-Kanade flow field, $\mathbf{u} = (A^\top A)\, A^\top \mathrm{b}$.
- Warp older image frame to the newer one using the estimated flow field $\mathbf{u}$ (actually, it is not easy in practice).
- Refine the estimate by repeating the iterations.



Iterative matching refinement (4)

- Estimate velocity at each pixel using one iteration of Lucas-Kanade flow field, $\mathbf{u} = (A^\top A)\, A^\top \mathrm{b}$.
- Warp older image frame to the newer one using the estimated flow field $\mathbf{u}$ (actually, it is not easy in practice).
- Refine the estimate by repeating the iterations.



Iterative matching refinement implementation notes

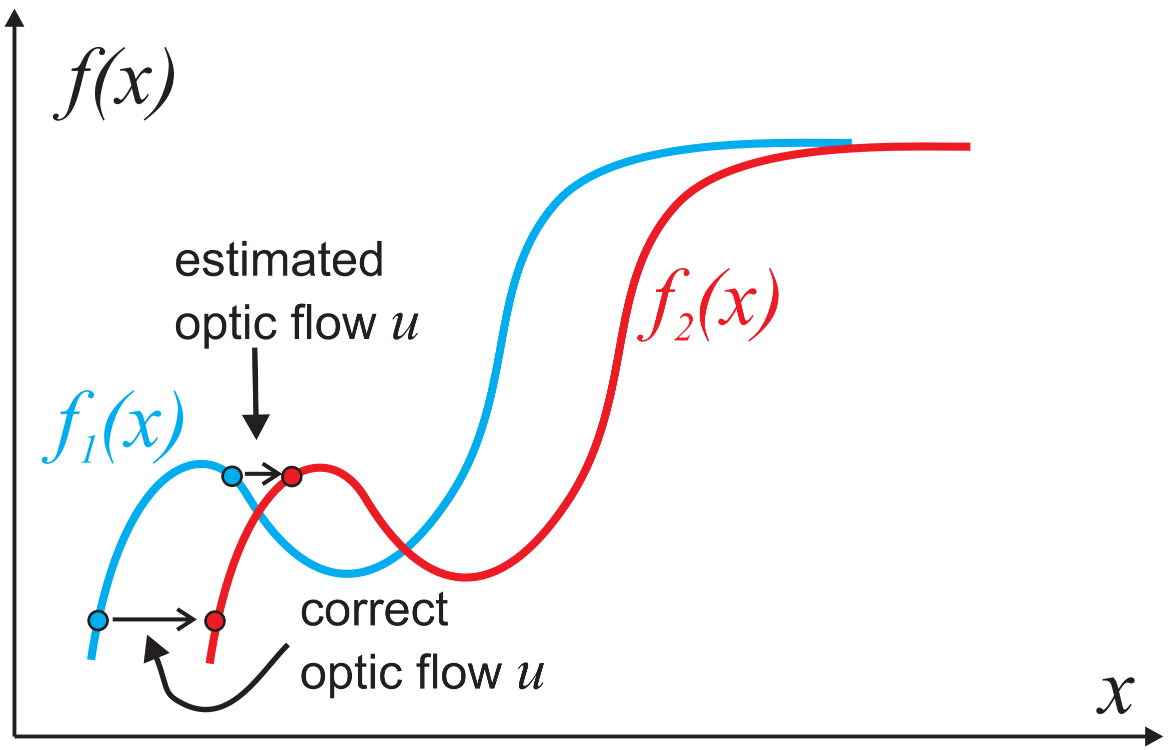
- Warping is not easy. It has to be checked that errors in warping are smaller than the estimate refinement.
- It is often useful to low-pass filter the images before motion estimation. The derivative estimation and linear approximations to image intensity become better.
- Warp older image, take derivatives of the newer one in order not to re-compute the gradient after each iteration.

Method pitfalls and way around them:

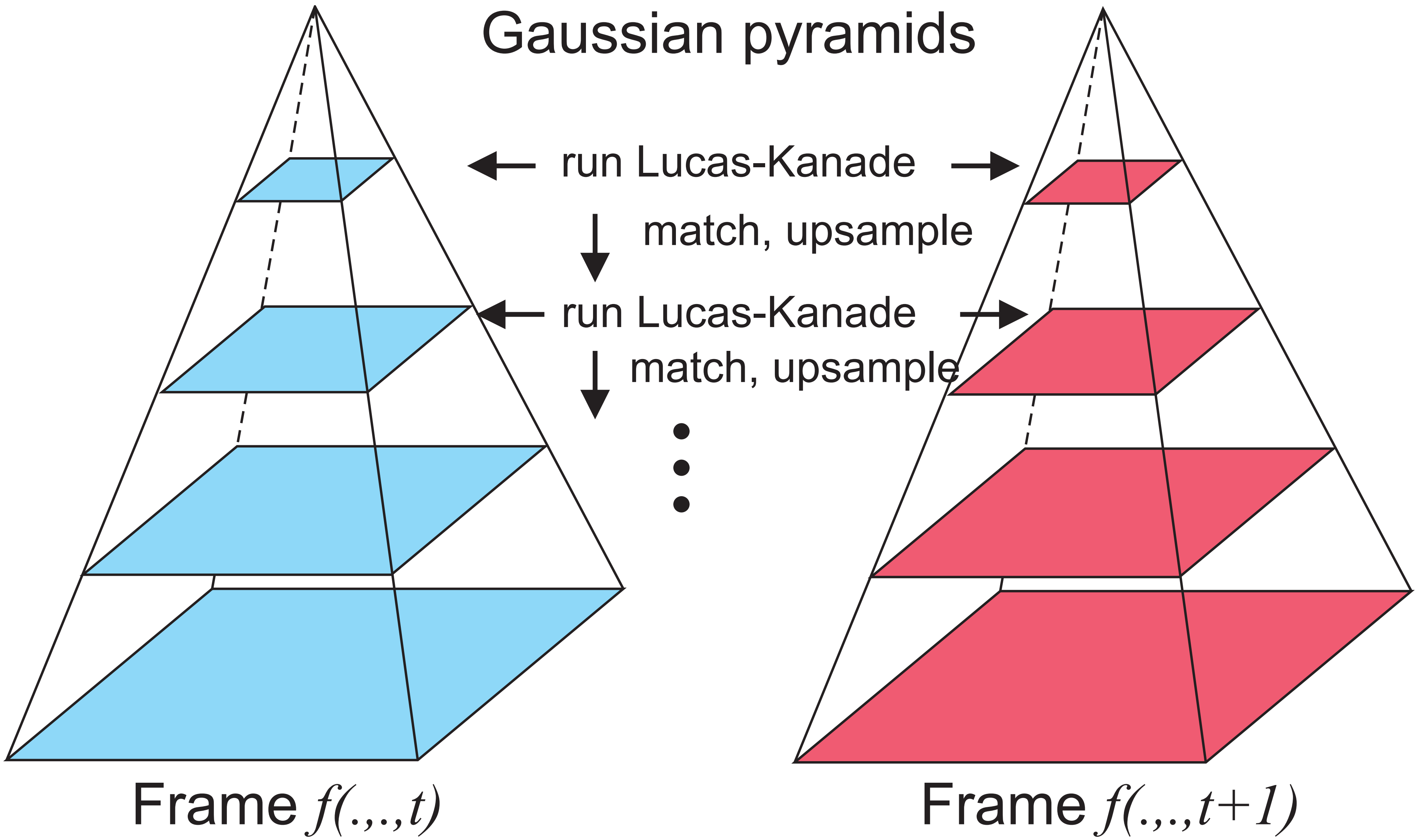
- Intensity constancy assumption does not work. ⇨ Use interest point-based method.
- A point does not move as its neighbors. ⇨ Use regularization-based algorithms (as Horn-Schunk and it followers).
- Motion is not small and/or aliasing problems. ⇨ Multiscale analysis.

Temporal aliasing problems

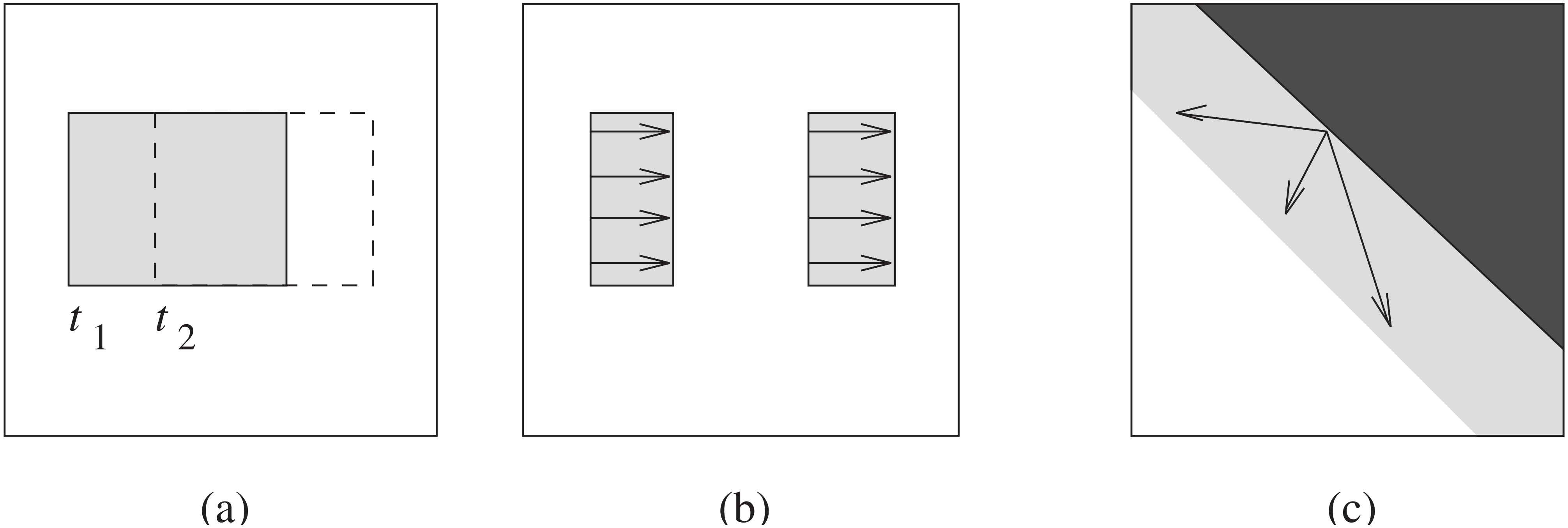
- Temporal aliasing problems occurs because many pixels can have same intensity.
- In matching, ambiguities occur because the source pixel can be matched to many target pixels.



Lukas-Kanade on the multi-scale



Problems of the differential motion analysis



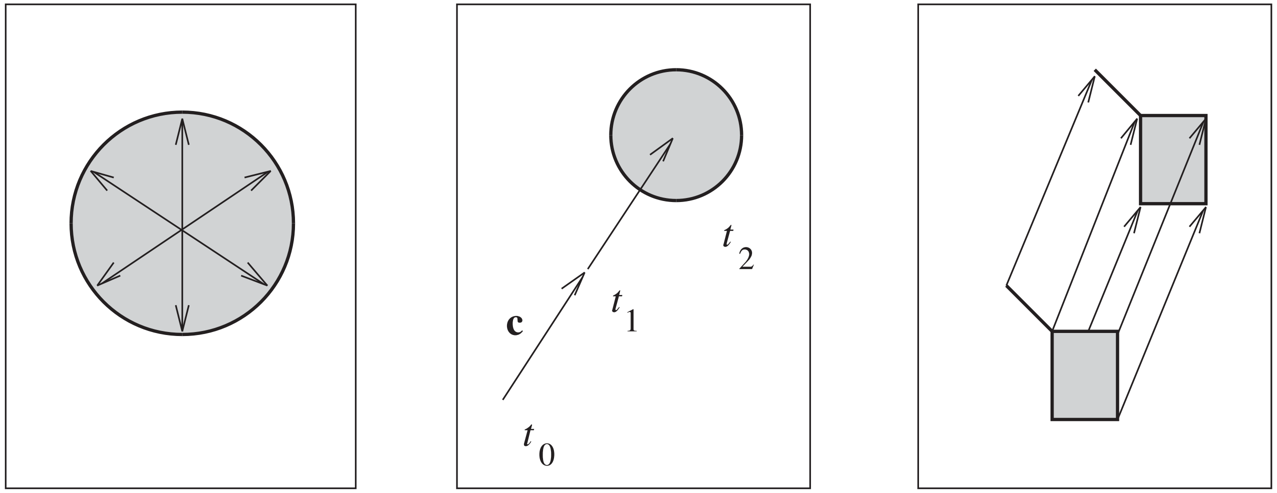
(a) A shifted rectangle.

(b) The observed difference.

(c) The aperture problem – ambiguous motion.

Simplifying object motion assumptions

- Maximum velocity (a).
- Small acceleration (b).
- Common motion.
- Mutual correspondence (c).



(a) (b) (c)

Differential motion detection

$$d(i,j) = \begin{cases} 0 & \text{if } |f_1(x,y) - f_2(x,y)| \le \varepsilon \\ 1 & \text{otherwise} \end{cases}$$

Problem: $d(i,j)$ does not show the direction of motion.

Differential analysis, example (1)

Frame 1 of 5

Frame 2 of 5

Differential analysis, example (2)

Frame 5 of 5

Difference between 1 and 2 (inverted)

Cumulative difference image

$$d_{cum}(i,j) = \sum_{k=1}^{n} a_k |f_1(i,j) - f_k(i,j)|$$

- Used if direction of motion has to be detected or trajectory of small object followed.
- f_1 is the first frame of the sequence (reference image).
- a_k is a weight coefficient.
- Resulting value shows how many times the gray value was different (if not weighted) in comparison to reference image.

Cumulative difference image, example

Problems with differential motion detection

- Motion detected only at places where objects were and newly appeared (ideally).
- Changing illumination causes changes in brightness too.
- Way around: updated reference image, often only averaging with forgetting.
- More advanced solution: detect moving blobs, treat the rest as background $\Rightarrow$ background subtraction (called also motion segmentation).

Background model

- Methods try to estimate the most probable background model.
- Each pixel is compared to the background model with a tolerance bound.
- Parameters of the background model and tolerance bounds are updated through the sequence.

Three prominent background models

- Three frames differencing, VSAM project, Carnegie-Mellon Robotics Institute, Collins 2000.
- Introducing chromaticity, McKenna 2000.
- Mixture of Gaussians, MIT Stauffer & Grimson 1999.