Computer Vision

Lecture 10: Introduction to dynamic vision; image motion

Last lecture

- Seeing with two eyes
- Stereo geometry
- The correspondence problem

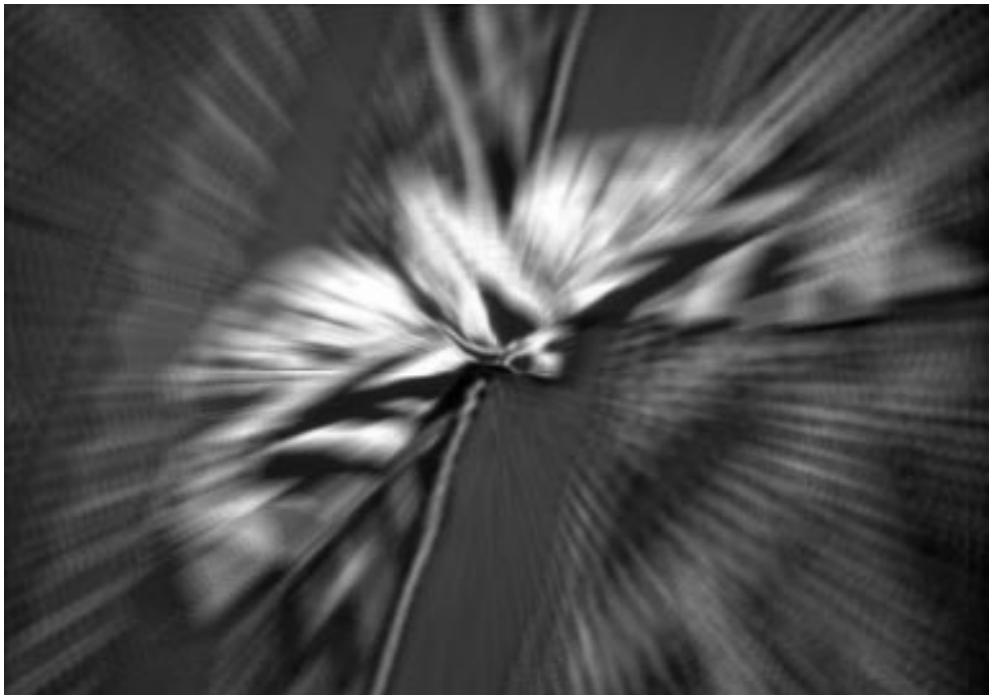
This lecture

- Introduction to visual motion and dynamic vision
- The optic flow field
- Detecting image motion
- Measuring optic flow

Introduction to visual motion

Why study motion in vision?

- **Motion** provides more information than can a static image. Traditionally, it has been seen as a powerful method for inferring 3-D layout, like stereo, but it offers much more than this.
- People and animals use motion information. Indeed, some psychologists regard motion as the foundation of vision.
- Technological advance: memory and processors now allow image sequences to be manipulated.
- Robotics, autonomous vehicles and surveillance applications require real-time predictive information.
- More than 50% of papers in major computer vision conferences now deal with motion in some way.
- The world is dynamic!

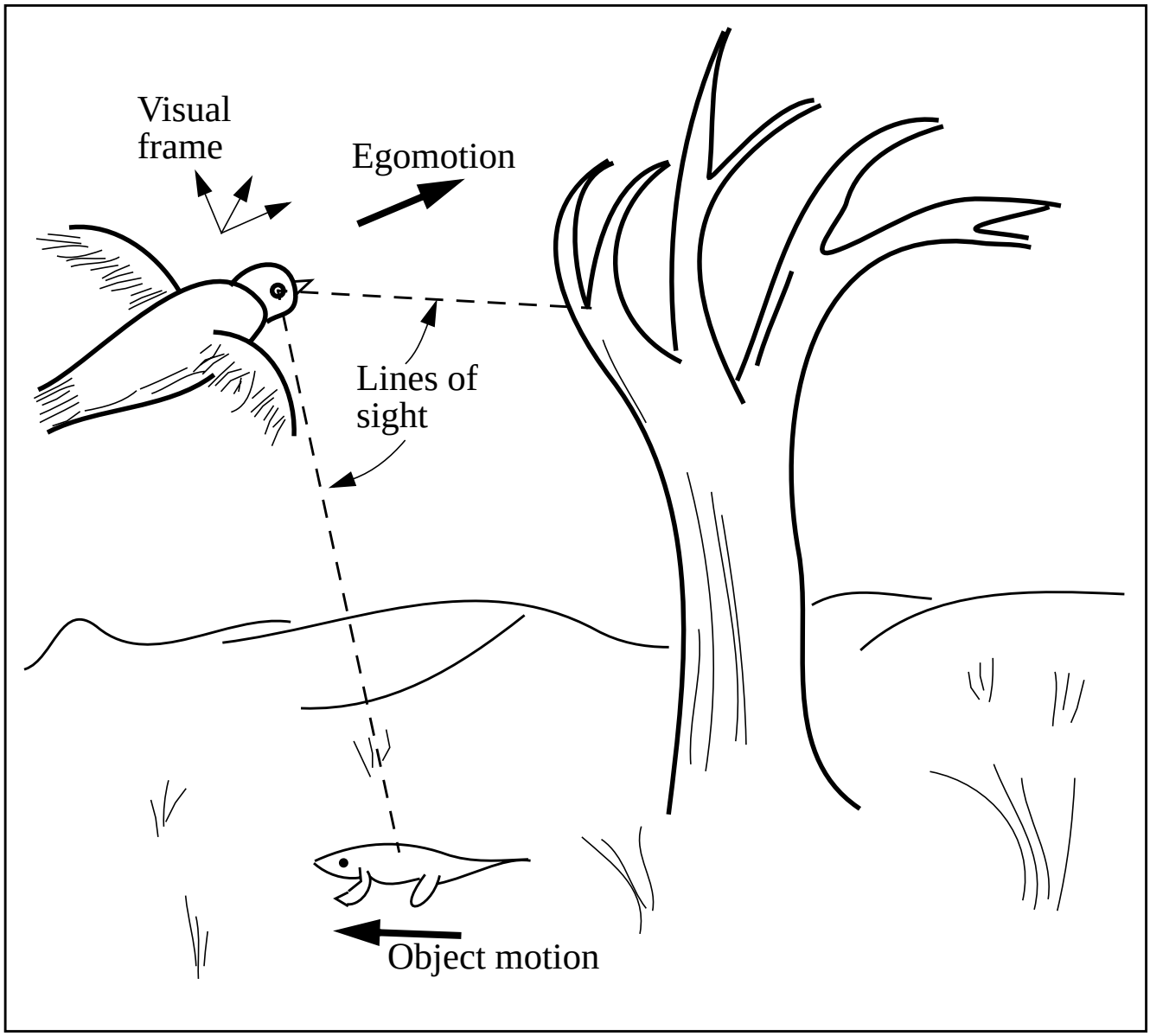


The optic flow field

The set of changing directions of **lines of sight**.

Line of sight: straight line from **point of observation** to **surface point**.

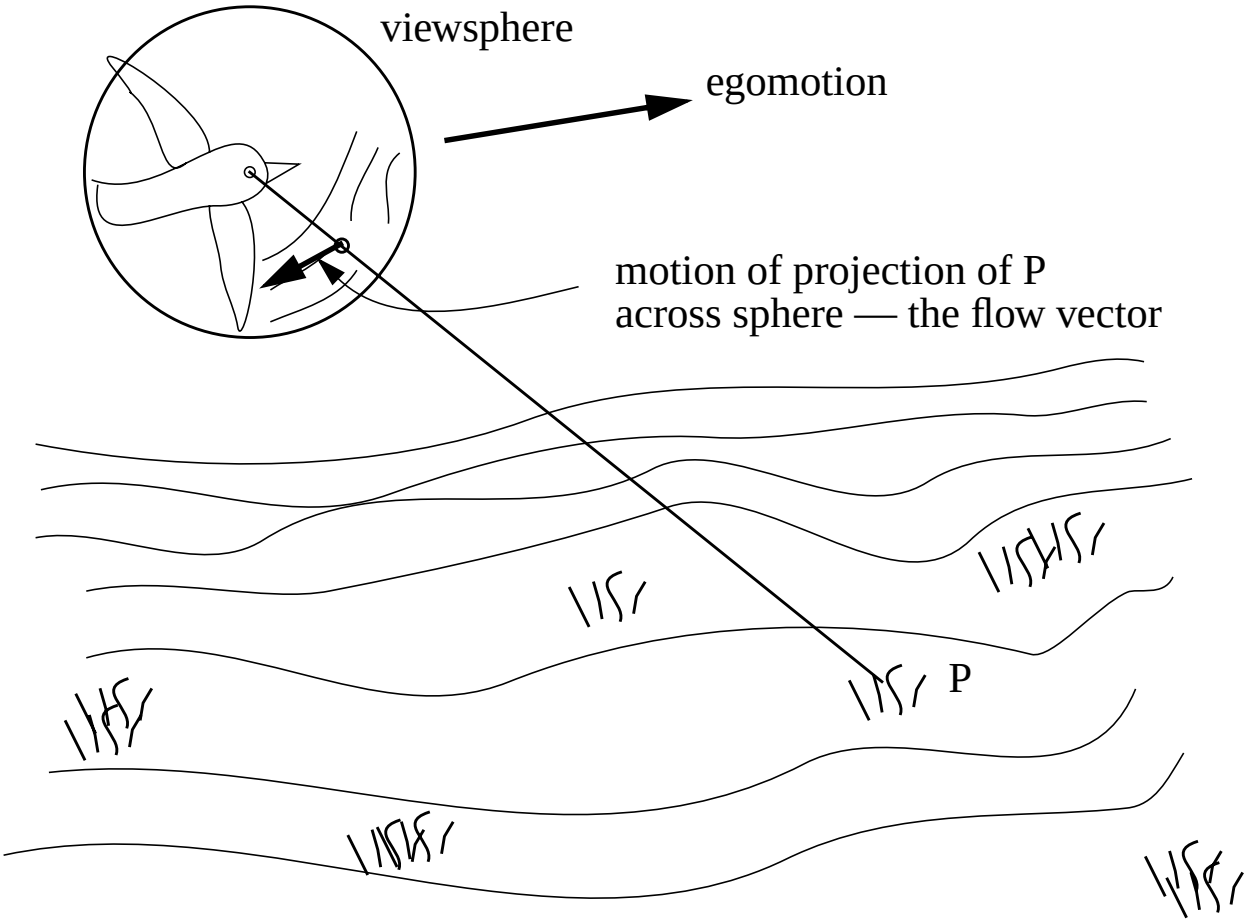
The direction is measured in a **frame of reference** attached to the eye or camera. The direction changes as a result of either **egomotion** or **object motion** (or both).



The motion of each point is described using an **optic flow vector**.

Mathematically, this is the rate of change of the unit vector along the line of sight to the surface point.

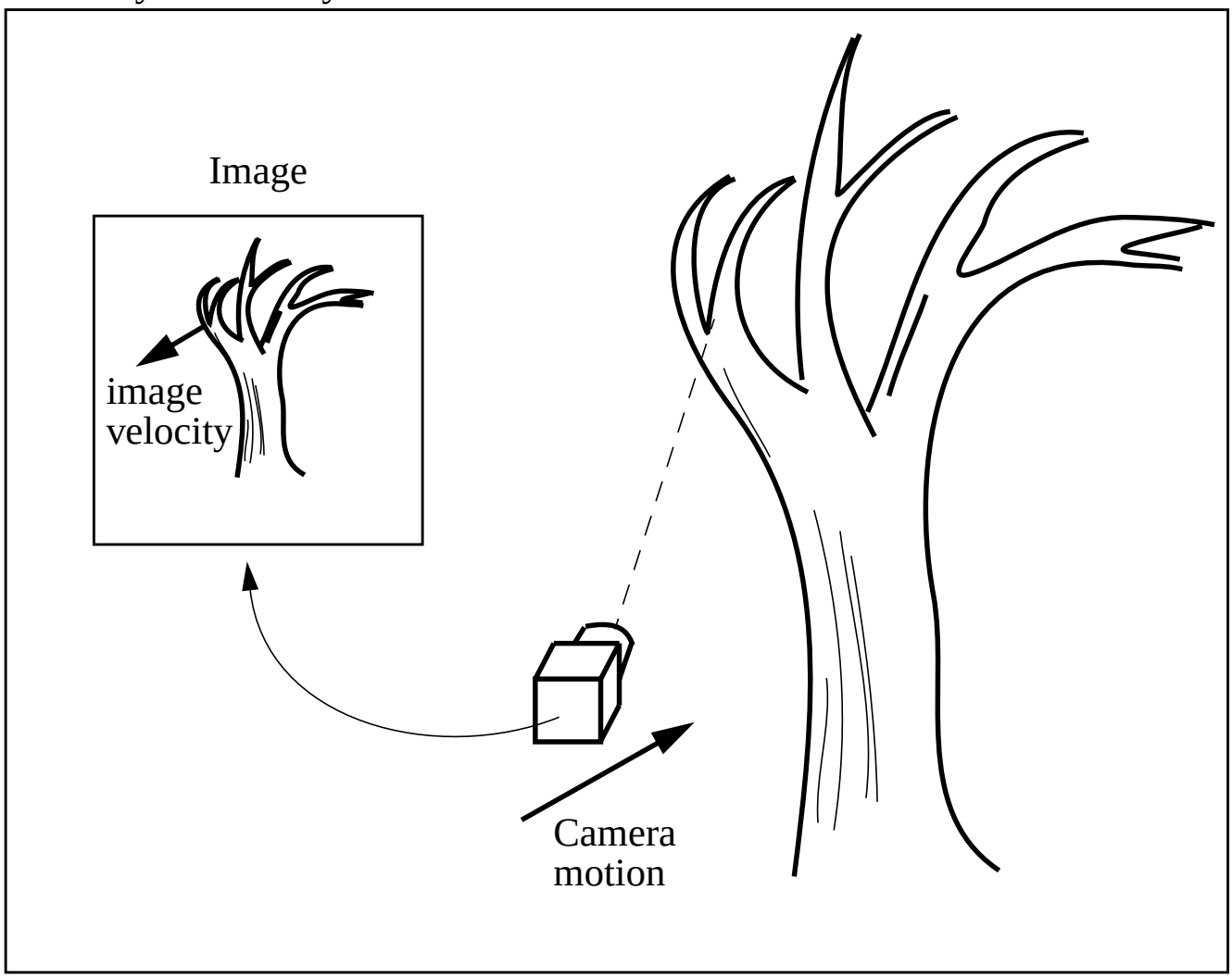
We can think of this as the motion of the intersection of the line of sight with the viewsphere (which moves with the eye or camera).



The collection of all the flow vectors, distributed across the viewsphere, is called the **optic flow field**.

In computer vision, we often use a camera model to analyse optic flow. Then an optic flow vector is represented by the **image velocity** of a feature in the 2-D image plane.

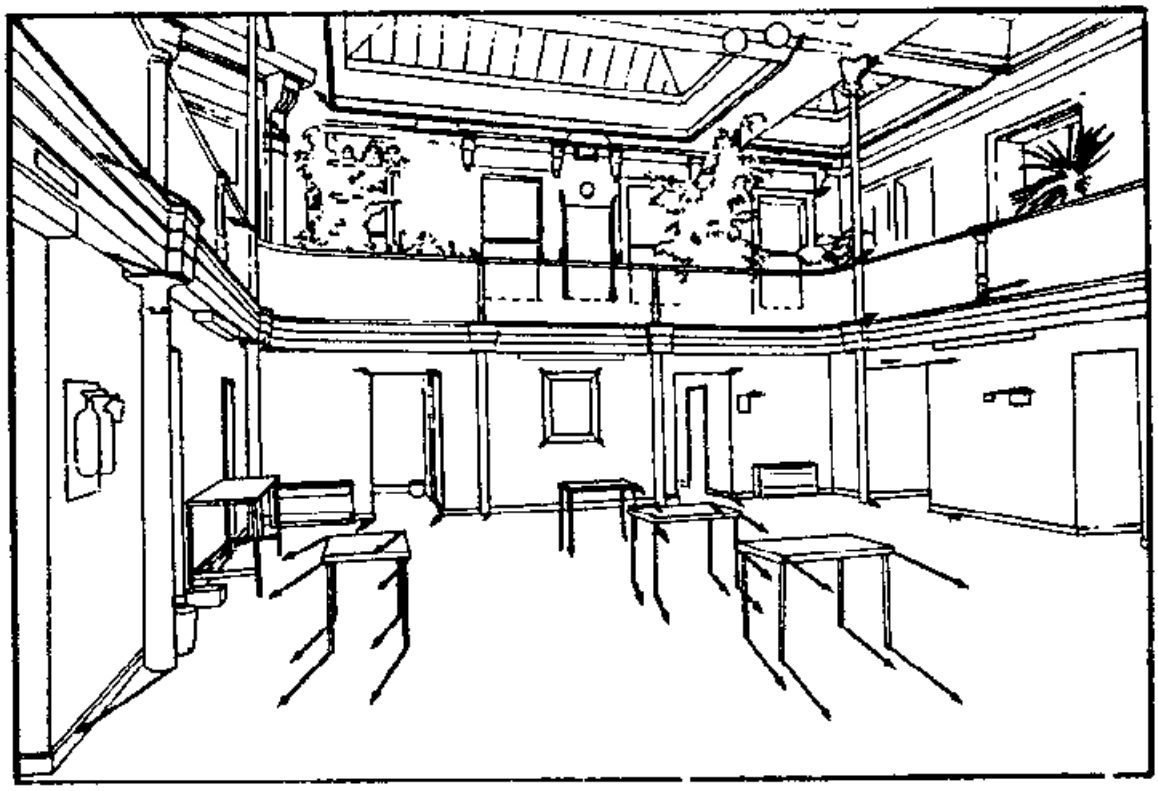
This is an approximation to the true optic flow because the correspondence between the image feature and the surface point may change with time. But it is usually sufficiently accurate.



(Note that some authors use “optic flow” to mean the observable part of the image velocity.)

Optic flow fields can be *sparse* (vectors defined only for specified features) or *dense* (vectors defined everywhere).

Here is an example of a sparse optic flow field for pure egomotion through an interior environment.



Flow vectors are indicated by arrows. The camera is moving towards the square object at the rear of the room.

In general, optic flow fields are more complex than this.

Using optic flow

To be useful in computer vision, an optic flow field must be

- Measured

  From an one or more images separated in time, estimates of some optic flow vectors must be obtained.

- Interpreted

  Information must be obtained about some of

  — the motion of the camera

  — the motion of objects

  — the orientation and position of surfaces relative to the camera

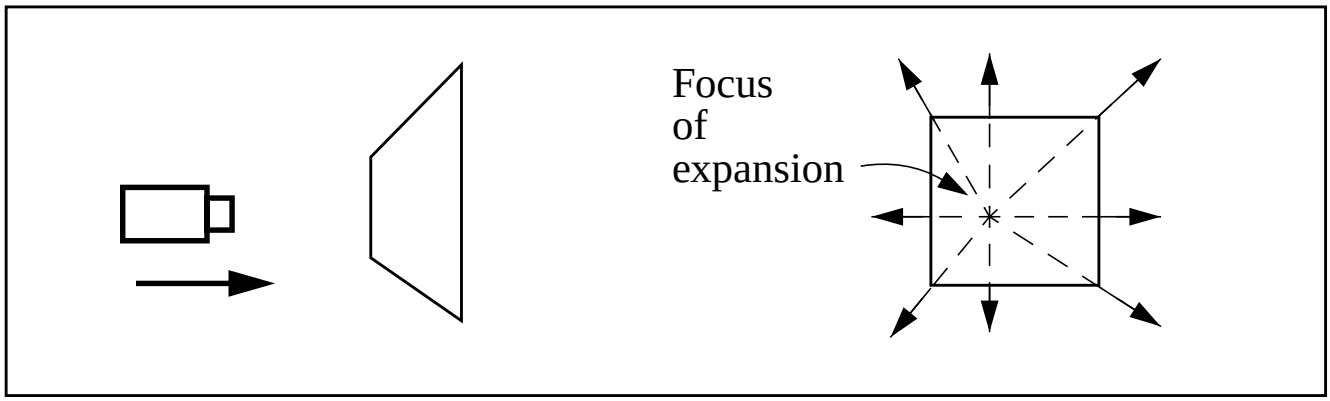
  This information might be **predictive**: it might help in the control of actions which need to anticipate events.

There is much research on both of these aspects.

In some systems, the two stages are combined.

A simple interpretation ...

is possible in the simple case of camera movement. The **focus of expansion** indicates the direction of motion, and the **rate of expansion** gives estimated time-to-collision.



Optic flow in humans and animals

Our own visual system is extremely good at **motion segmentation**. We are very sensitive to relative motion in the visual field.

There is some evidence that people and animals use the relationship between rate of expansion and time-to-collision to control interceptive actions.

Observations of **gannets diving** (large sea birds) indicate that they may fold their wings when the rate of expansion reaches a trigger level. The acceleration due to gravity allows different strategies to be tested.

Similar observations of **people punching balls** indicate that people may use rate of expansion to synchronise their movements.

Experiments in the **swinging room** show that people use flow for balance.

Studies of **flies landing** have provided further evidence.

There are some suggestions that neurons sensitive to expansion can be found.

Dynamic vision

Optic flow is an important element of dynamic vision. But it is not the whole story.

- Image motion is richer than the optic flow field.

  We have defined the **instantaneous optic flow field** which corresponds to image velocities. It can be estimated from 2 frames of a sequence. But there is useful information in image **accelerations**, and in general in **feature trajectories** which need a larger number of frames to estimate. These possibilities have received relatively little attention.

- Vision is only part of the system.

  A key element of true dynamic vision is that it is part of a **perception-action cycle**. Our own motions give rise to visual feedback, both from egomotion and manipulation. Understanding vision as part of a control process is only just beginning. The technology – active camera heads and mobile robots – is only now facilitating research.

  Dynamic vision may also involve higher order interpretation – perhaps of intentions and plans of other agents.

  Purposive vision and **active vision** are other terms used (with some variation of meaning) to describe the use of vision in a dynamic environment.

Applications of dynamic vision

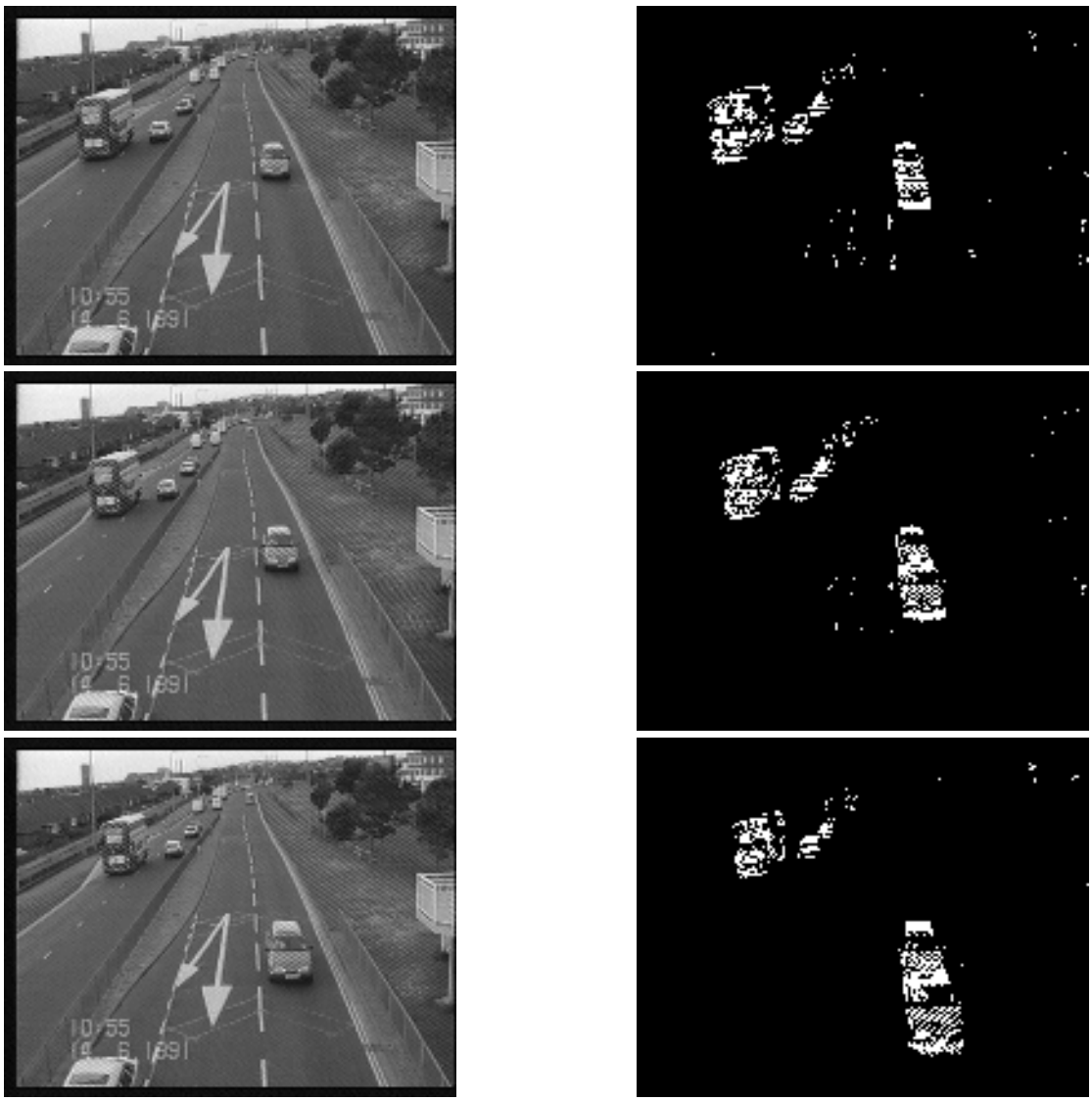
Applications span most of the areas in which computer vision can be employed. Examples include

- Surveillance – road traffic monitoring, intruder detection etc.
- Mobile robots and autonomous vehicles – obstacle avoidance, and also the general layout of the environment for navigation.
- Robot manipulators – guiding robot arms to a correct grasp on an object.

Detecting motion

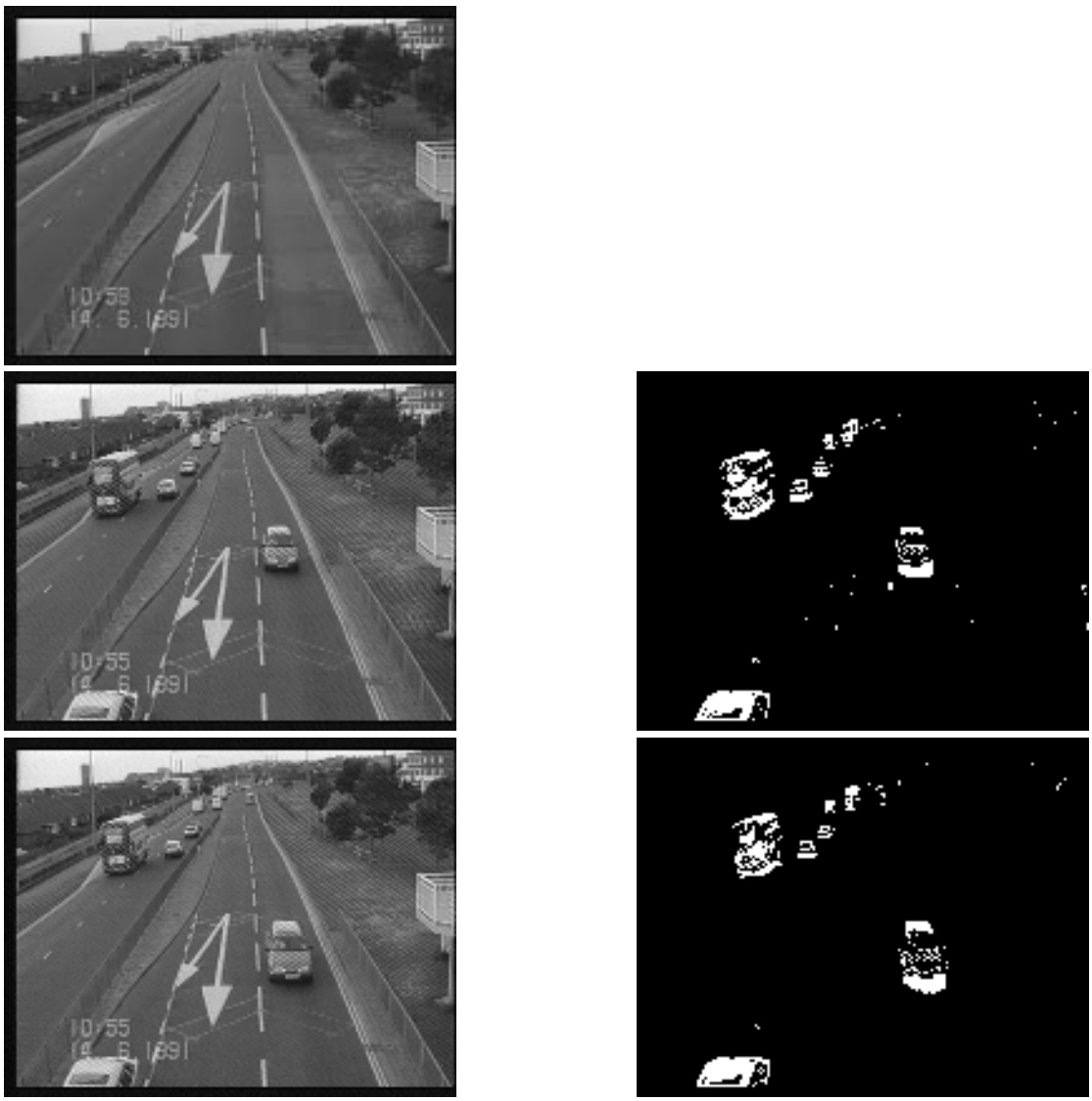
When the camera is static but objects are moving, one simple and often-used way to detect motion is **image differencing**.

Successive image are subtracted pixel by pixel, and differences exceeding some threshold are recorded. In regions of motion, the grey level changes, and these are highlighted.



This results in confusion between the start and end point of an object's motion.

It is better to take differences relative to a **reference image**. Constructing the reference image may be difficult: it may or may not be possible to find a time when all moving objects are absent. Also the reference image must be updated to taken account of slow changes, such as changing illumination. A system of **temporal averaging** can be used to overcome these problems.



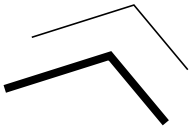
Measuring optic flow

Basic method: track the motion of a point in the image between successive frames.

Many different kinds of image feature are possible – the choice depends on various trade-offs.

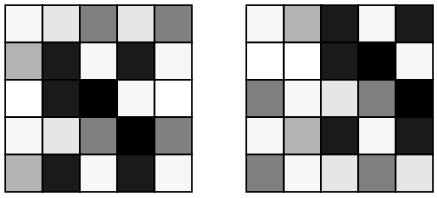
- edges

- corners



- “blobs”

- local neighbourhoods



- high-level features

Three main principles are used

- Similar features match.

  A feature in one image must correspond to the matching feature in the other, using some characteristics such as

  — geometrical (shape) similarity

  — similarity of grey levels

  The assumption is that the appearance of a scene feature does not change significantly between frames.

- The image moves **coherently**.

  — The flow is often locally uniform. Discontinuities may occur at scene boundaries. Local flow vectors that are similar reinforce one another.

  — In active vision, the flow will not be uniform close to the fixation point, but will still approximate locally to a simple smooth pattern.

  The assumption is that the scene is made up of extended reasonably smooth surfaces, which move rigidly or at least distort smoothly when moving.

- The optic flow vectors are small.

  The amount of motion between frames is small compared to the size of the image.

  The actual limit depends on the matching method. Some (gradient methods) require motions to be less than a few pixels between frames. Matching very high level features may allow much larger velocities.

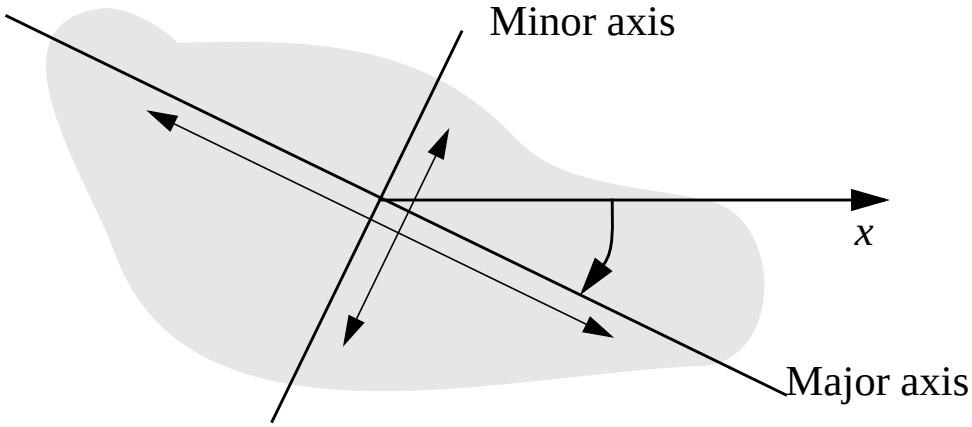
Some simple algorithms

An image sequence for testing:

Algorithm 1: Match blobs

1. Smooth with a Gaussian (or other) smoothing mask.
2. Threshold at a suitable level (determined by image statistics).

3. Find 8-connected regions above the threshold: “blobs”.
4. Remove blobs smaller than some limit.
5. Describe each blob’s shape using lengths of major and minor axes and orientation (principal component analysis of pixel coordinates).



6. Take a blob in image 1 and compare it with each blob in image 2. Define the matching blob as the one with the maximum similarity. An example of a blob similarity measure, using the statistics chosen, is

$$\frac{1}{\frac{\text{orientation difference}}{360} + \frac{\text{major axis diff}}{\text{mean major axis}} + \frac{\text{minor axis diff}}{\text{mean minor axis}} + w\text{distance moved}^2}$$

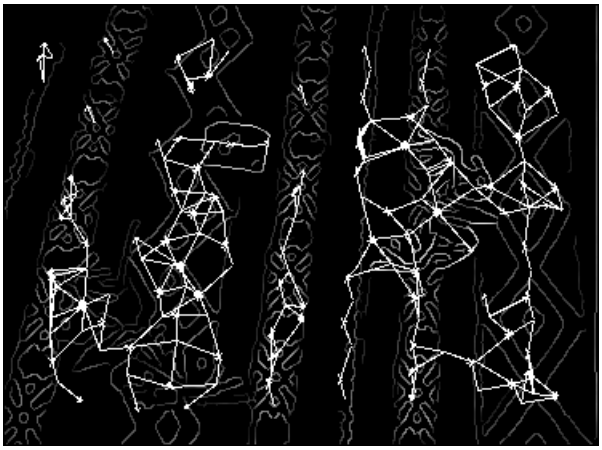
   Note that this also takes account of the distance moved by the blob centre: a small movement is preferred. Here w is a weight which determines the acceptable distance for the image motion.

7. Repeat for each blob in image 1.

Algorithm 2: Find local support for the flow

1. Convolve with a Gaussian smoothing mask.
2. Find local maxima of the grey level (pixels where all 8 neighbours have a lower value). These serve as structureless features – we use their position and nothing else.

3. For each feature in image 1, find all the features within some radius in image 2. These give all the possible matches, assuming some maximum flow velocity.



4. For each possible match, accumulate the **support** that it receives from other nearby possible matches.

$$\sum_{\text{matches for nearby features}} \left(\frac{w_1}{w_1 + \text{distance}^2} \right) \left(\frac{w_2}{w_2 + |\text{motion difference}|^2} \right)$$

   where w_1 and w_2 are constant weights, *distance* is the distance between the two features in one image, and *motion difference* is the absolute difference between the vector match we are assessing and the vector for the other match.

   The time is quadratic in the number of possible flow vectors, and so the calculation may be restricted to nearby features.

5. For a given feature, select the match with the highest support, if this exceeds a threshold. May also check that the same result is obtained if the images are exchanged.

Variations on the simple algorithms

The algorithms described are only examples. In general, methods need to be tailored to the problem.

Matching algorithms can be used with many kinds of local image structure.

Grey level neighbourhoods are often matched using a sum-of-squared-differences measure. This leads to **correlation** algorithms, which can give good results but which can be computationally expensive.

Image differencing may be used to find regions to match when the background is static and objects are moving.

Higher order features, such as recognised objects, can easily be matched. There is then usually little ambiguity.

When objects are matched across several frames, the algorithms are usually referred to as **tracking algorithms** rather than optic flow algorithms.

Flow support algorithms have many variations.

Other simple features can be used. **Corners** are popular, using e.g. Plessey and Oxford corner detectors.

A **Hough transform** can be used to accumulate evidence for the overall flow in a region. This can be good for finding background motion, and is efficient.

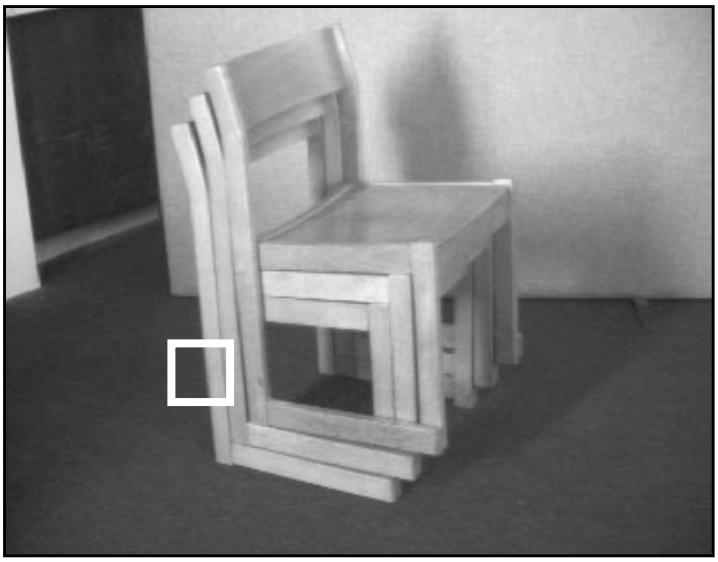
A **scale space** approach using a **resolution pyramid** can be effective. Approximate flow vectors are found at low resolution (few pixels, little ambiguity) and used to guide the search for matches at higher resolutions.

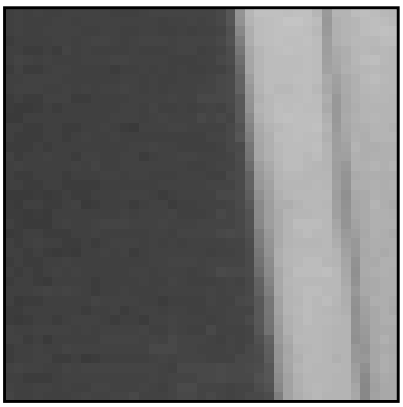
The two approaches can be combined to give a range of powerful methods.

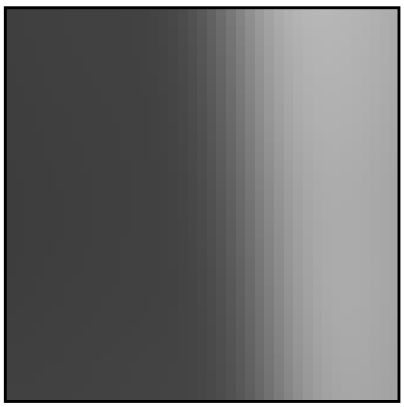
Gradient algorithms

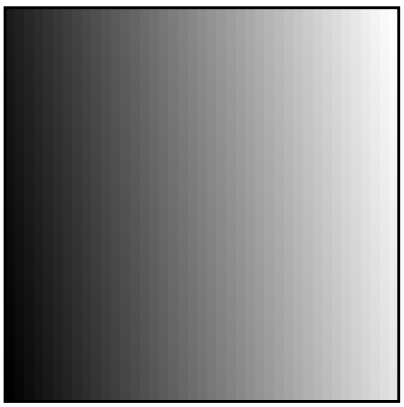
When the motion between frames is small compared with the scale of image structure, it is possible to avoid explicit matching by using a local model of grey-level structure.

The simplest model is the local **grey-level gradient**. We assume that locally, the image can be approximated by a grey-level function that is linear in the image coordinates.



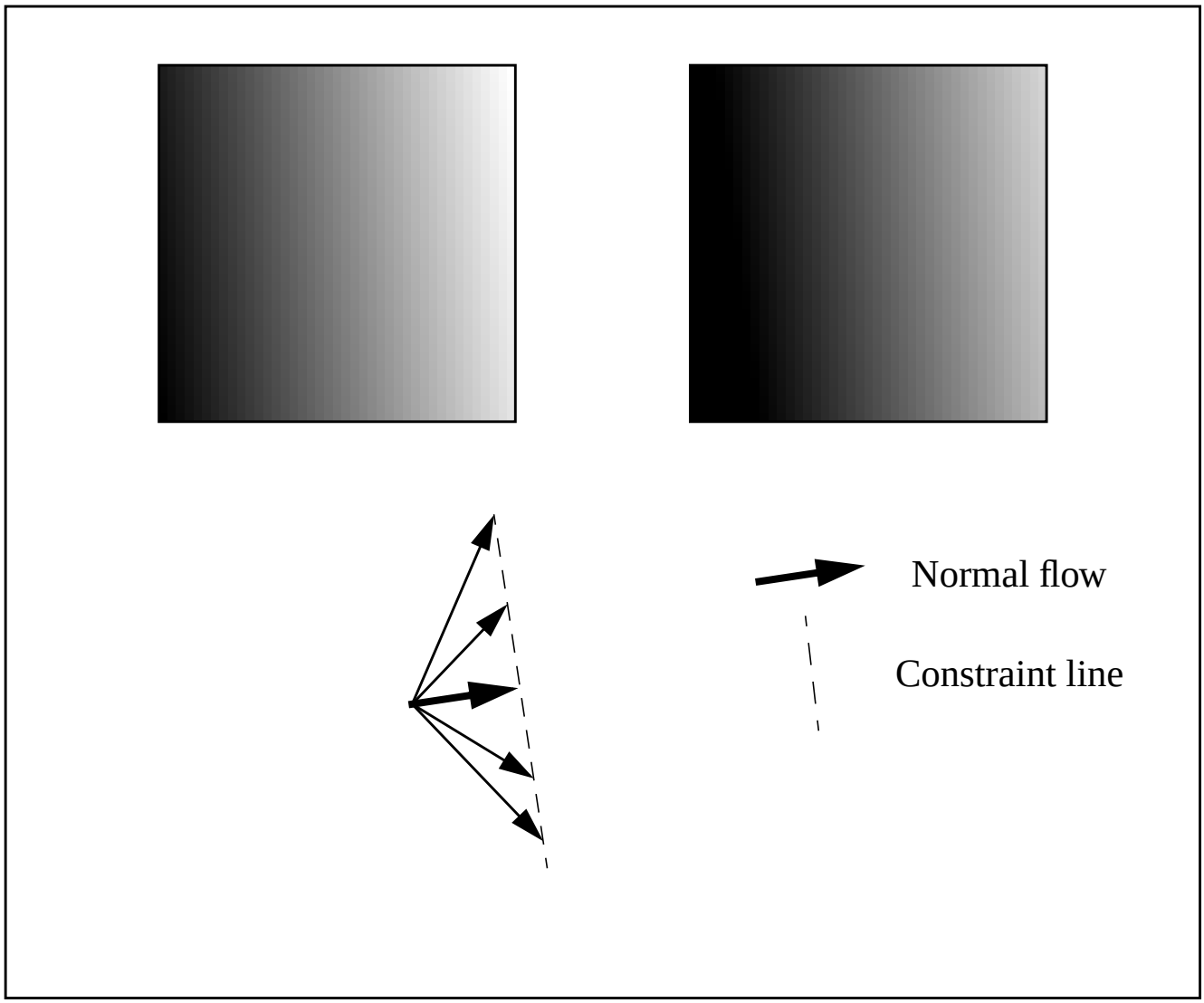






In the next frame, image motion will have caused a change in the grey levels of the local model.

Varous flow vectors are consistent with this change. The **normal flow**, the component of flow along the grey level gradient, is exactly determined. Other possible vectors lie on a **constraint line** perpendicular to the gradient.



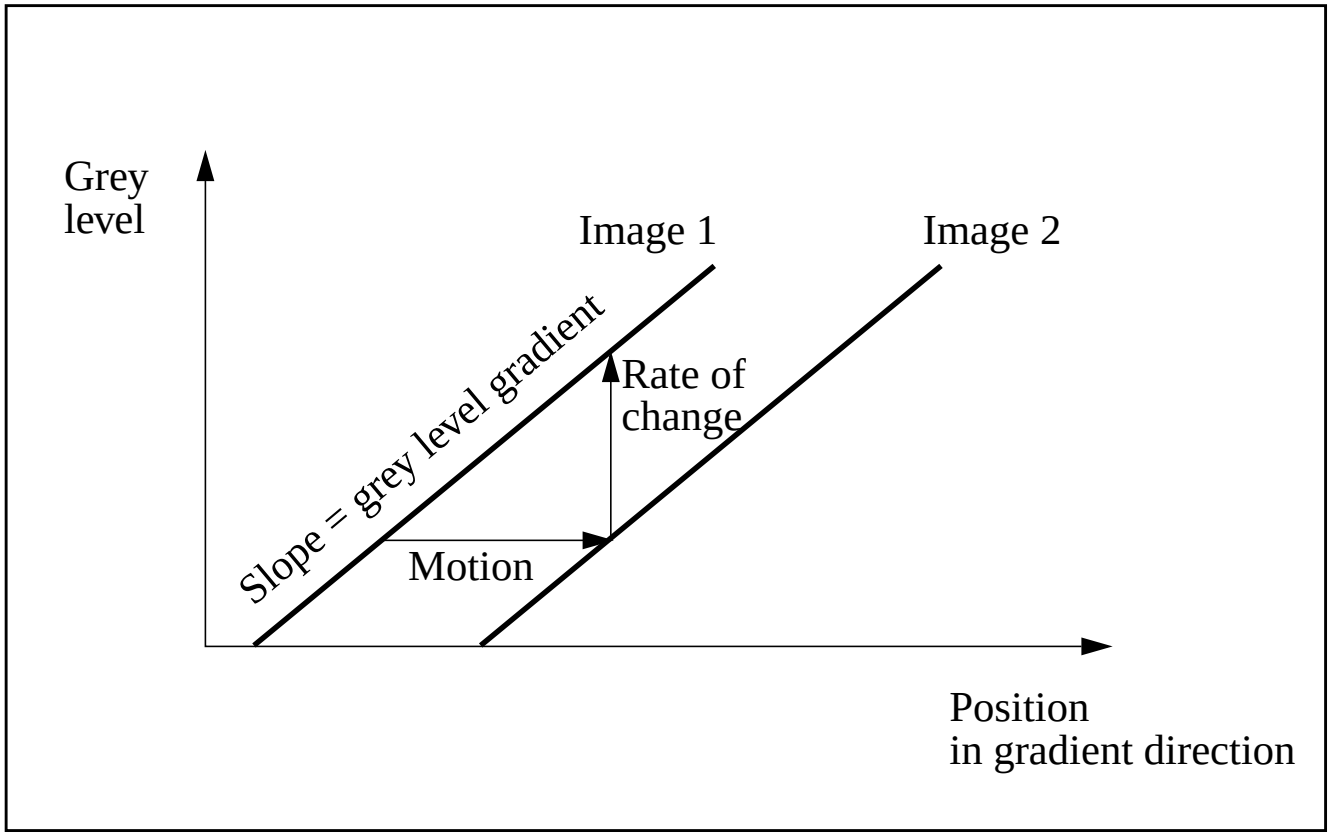
The grey-level gradient (direction and magnitude) can be estimated using standard static image processing techniques: smoothing and differencing along *x* and *y*.

The **direction** of the normal flow is along the grey-level gradient.

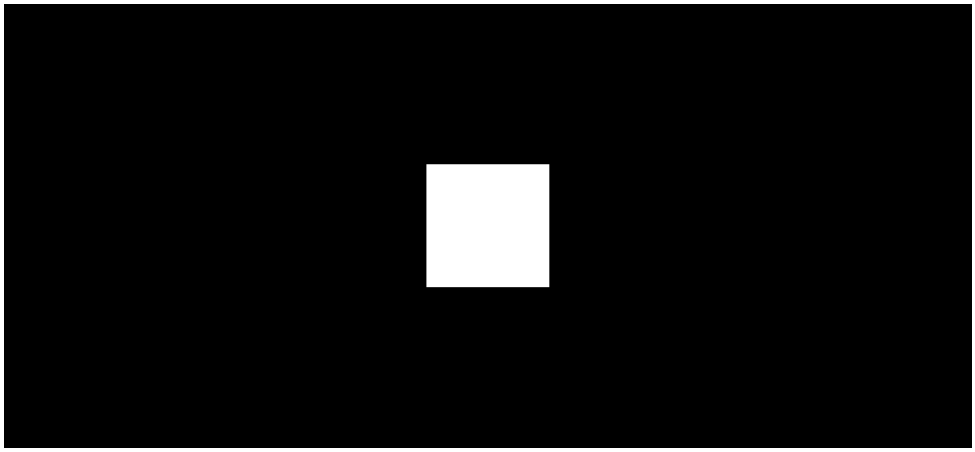
The **magnitude** of the normal flow is found from the grey-level gradient and the **rate of change of grey level** at the central pixel.

$$\text{Normal flow magnitude} = \frac{\text{Rate of change of grey level}}{\text{Grey level gradient}}$$

The rate of change of grey level can be estimated using smoothing and simple image differencing (subtracting values of corresponding pixels in the two frames). These operations are computationally inexpensive.



The problem that only the normal flow is estimated is called the **aperture problem** because when a gradient is seen locally (as if through an aperture) only the normal flow is visible.



Estimates of normal flow at different points are combined assuming local flow smoothness to estimate the full flow vectors. Algorithms are due to Horn & Schunck and to Hildreth. The **Horn-Schunck** algorithm is, in essence

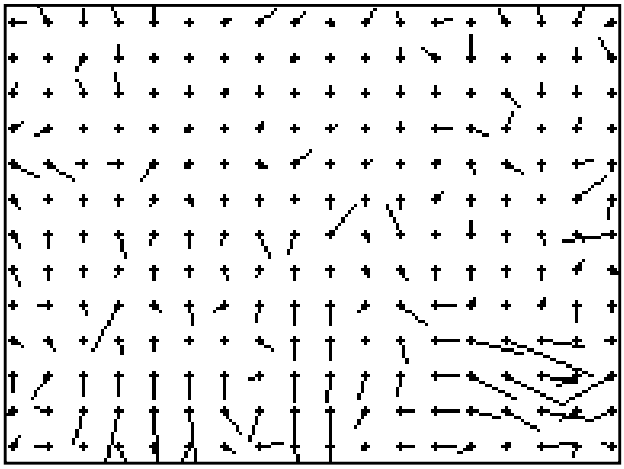
1. Estimate normal flow vectors and hence the constraint lines at many image points.
2. Initialise the flow field to the normal flow vectors.
3. Smooth the current flow field.
4. Move each flow vector closer to its constraint line.
5. If the changes are small, stop, else go to 3.

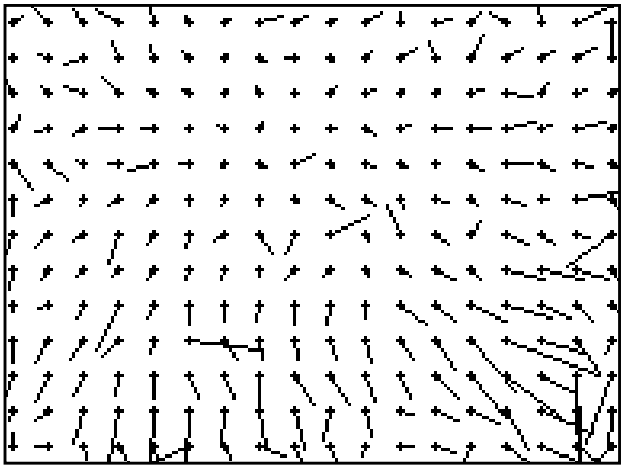
This amounts to iterative minimisation of a function combining

- flow smoothness
- constraints from estimates of the normal flow.

One problem is that **discontinuities** are smoothed over. More sophisticated algorithms can include a term that allows for line discontinuities.

As the iterations progress, the flow vectors move towards a flow field that satisfies the data but which is also smooth.





Spatio-temporal filtering

The normal flow is estimated using **spatial differencing** to estimate the grey-level gradient in the image, and **temporal differencing** to estimate the rate of change of grey-level between frames.

The differencing operators are implemented as filters (convolution masks). More complex spatio-temporal filters can be applied, giving more information about the local grey-level structure in space and time. The outputs of a family of such filters can be used to give optic flow estimates.

Such filters can be viewed as detecting gradients in the 3-D spatio-temporal image. This view of image sequences is very useful.