Dr. George Karraz, Ph. D.

Motion tracking

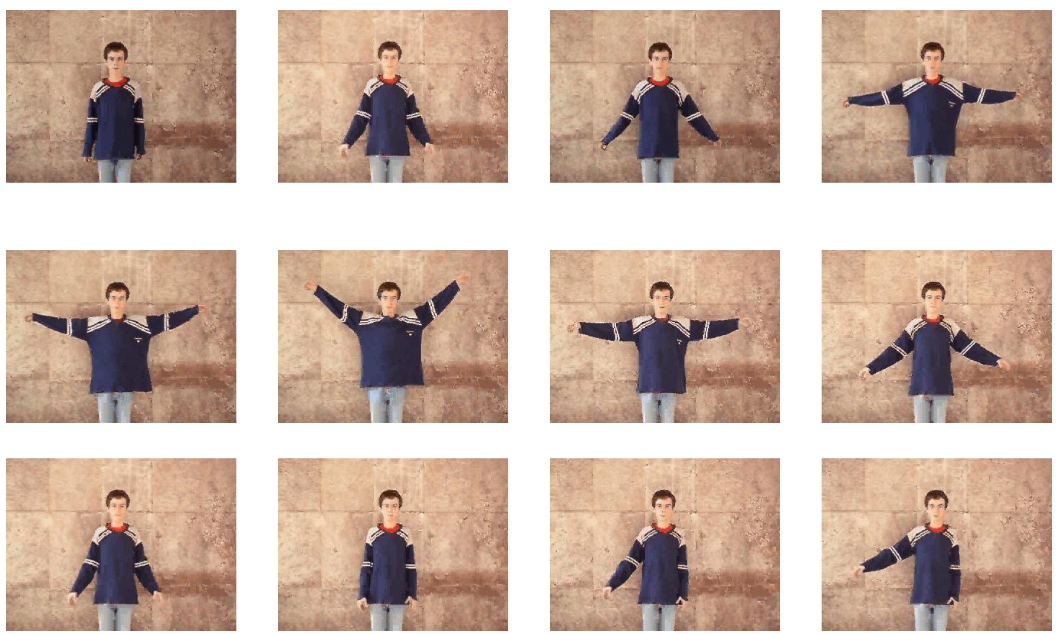
Dr. George Karraz, Ph. D.

Contents:

- The Problem
- Goals
- Approaches
- The Optical Flow Method
- Algorithm

The Problem

- Given a set of images in time which are similar but not identical, derive a method for identifying the motion that has occurred (in 2d) between different images.



Goals

- Input:
  - an image sequence
  - captured with a fixed camera
  - containing one or more moving objects of interest
- Processing goals: determine the image regions where significant motion has occurred
- Output: an outline of the motion within the image sequence

Motion Detection and Estimation

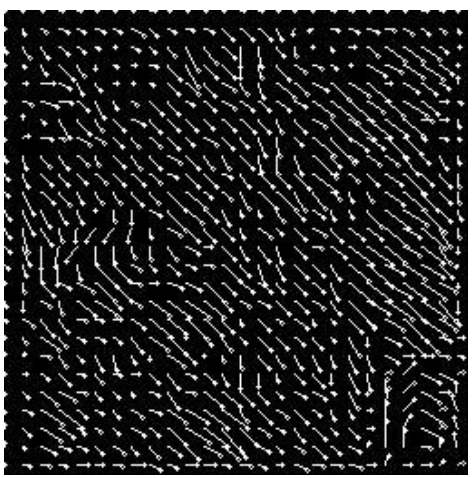
- Image differencing
  - based on the thresholded difference of successive images
  - difficult to reconstruct moving areas
- Background subtraction
  - foreground objects result by calculating the difference between an image in the sequence and the background image (previously obtained)
  - remaining task: determine the movement of these foreground objects between successive frames
- Block motion estimation
  - Calculates the motion vector between frames for sub-blocks of the image
  - mainly used in image compression
  - too coarse
- Optical Flow

What Is Optical Flow?

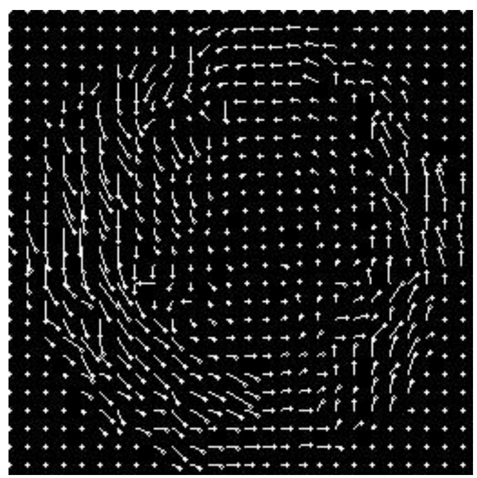
- **Optical flow** is the displacement field for each of the pixels in an image sequence.
- For every pixel, a velocity vector $\left(\frac{dx}{dt}, \frac{dy}{dt}\right)$ is found which says:
  - how quickly a pixel is moving across the image
  - the direction of its movement.

Optical Flow Examples

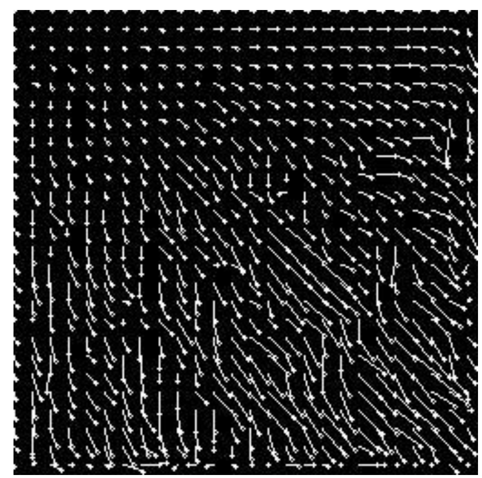
Translation



Rotation



Scaling



Algorithm

- Optical flow: maximum one pixel large movements
- Optical flow: larger movements
- Morphological filter
- Contour detection (demo purposes)

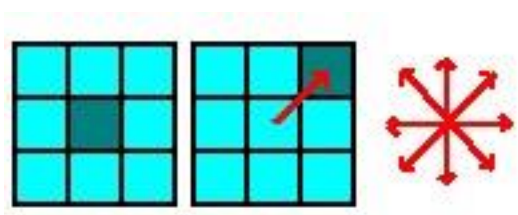
Optical Flow: maximum one pixel large movements

- The optical flow for a pixel (i, j) given 2 successive images k and $k+1$:

$$\overline{m_k(i, j)} = (x, y) \quad \text{so that}$$

$$\left| I_k(i, j) - I_{k+1}(i + x, j + y) \right| \qquad (1)$$

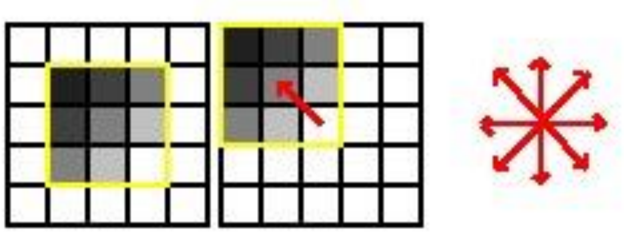
is minimum for $-1 \le x \le 1, -1 \le y \le 1$



k k+1

Optical Flow: maximum one pixel large movements (2)

- More precision: consider a 3×3 window around the pixel:



- Optical flow for pixel (i, j) becomes:

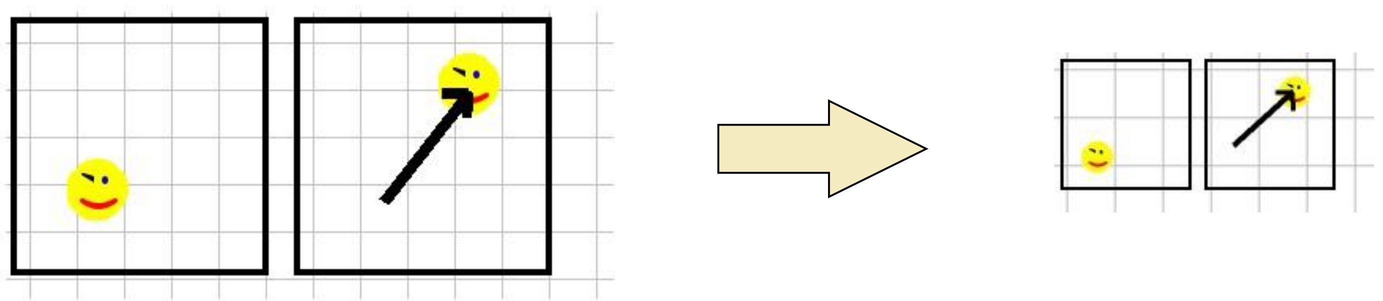
$$\overline{m_k(i, j)} = (x, y) \quad \text{so that}$$

$$\left| \sum_{u=-1}^{1} \sum_{v=-1}^{1} I_k(i+u, j+v) - \sum_{u=-1}^{1} \sum_{v=-1}^{1} I_{k+1}(i+u+x, j+v+y) \right| \qquad (2)$$

is minimum for $-1 \le x \le 1, -1 \le y \le 1$

Optical Flow: larger movements

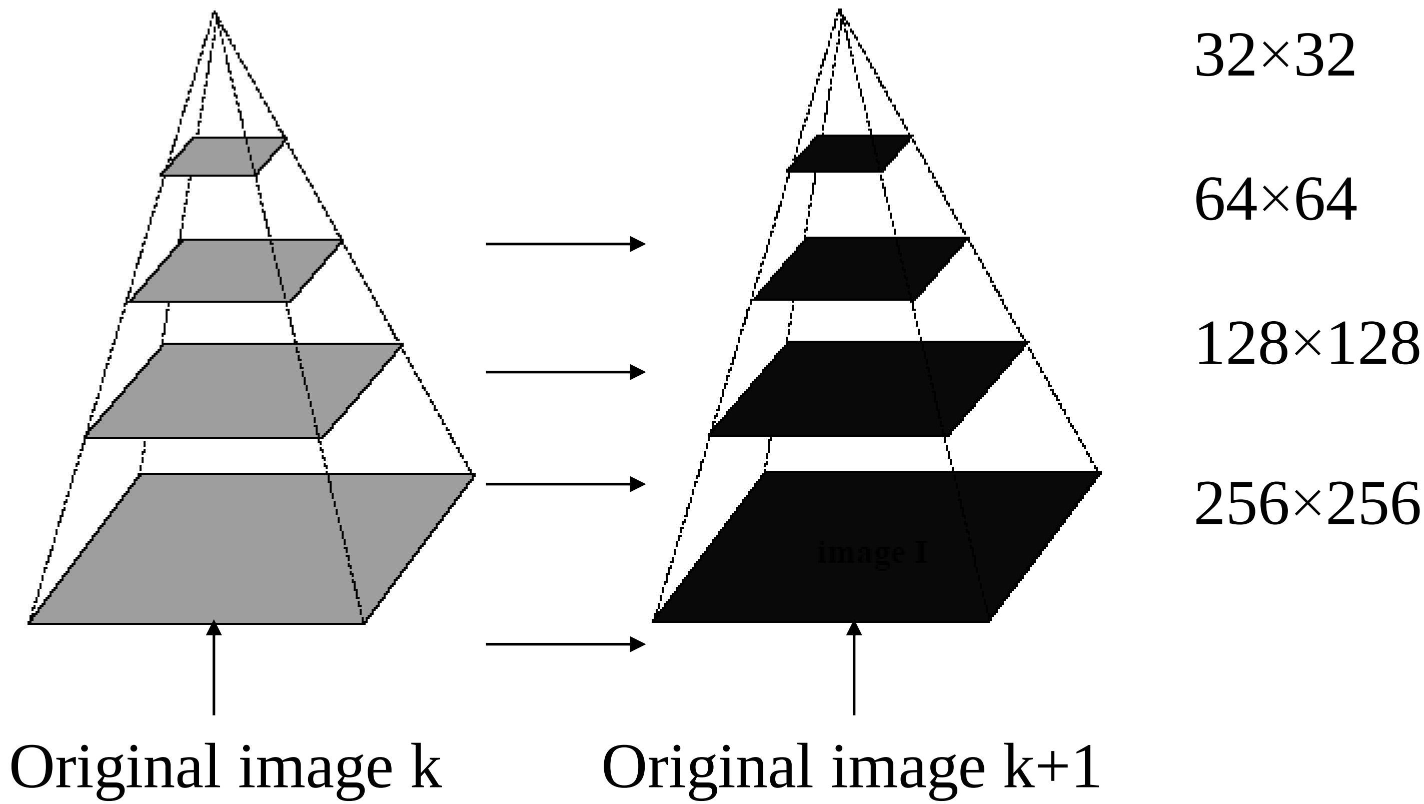
- Reduce the size of the image
  => reduced size of the movement



- Solution: **multi-resolution analysis** of the images
- Advantages: **computing efficiency, stability**

Multi-resolution Analysis

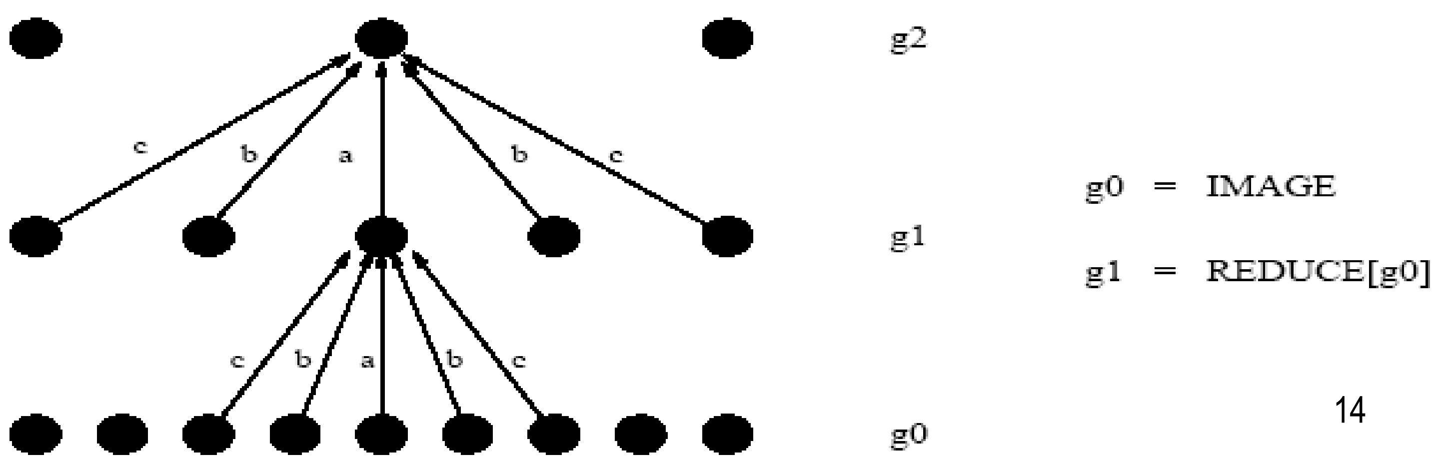
- Coarse to fine optical flow estimation:



Gaussian Pyramid

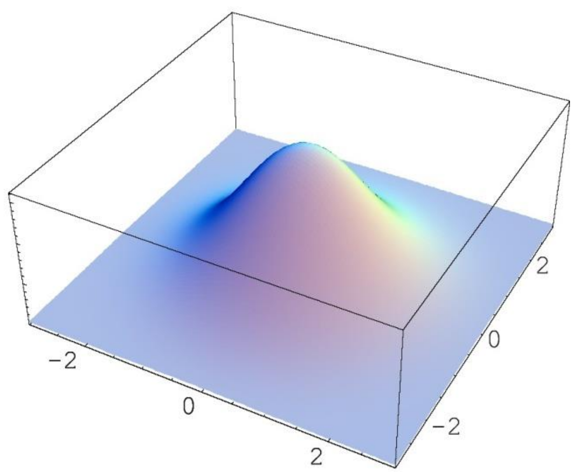
- Lowest level g_0 - the original image
- Level g_l - the weighed average of values in g_{l-1} in a 5×5 window:

$$g_l(i,j)=\sum_{m=-2}^{2}\sum_{n=-2}^{2}w(m,n)g_{l-1}(2i+m,2j+n) \qquad (3)$$



Gaussian Pyramid (2)

- The mask $G(m,n)$ is an approximation of the 2D Gaussian:



0.003	0.013	0.022	0.013	0.003
0.013	0.060	0.098	0.060	0.013
0.022	0.098	0.162	0.098	0.022
0.013	0.060	0.098	0.060	0.013
0.003	0.013	0.022	0.013	0.003

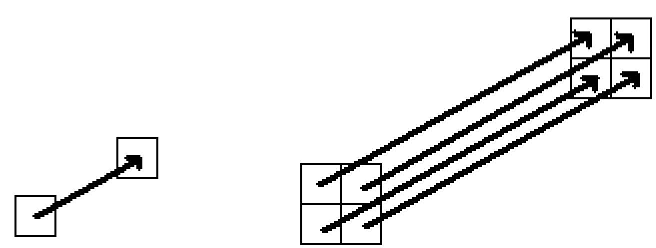
- The mask is symmetric and separable:

$$G(m,n) = G_r(m) * G_c(n) \qquad (4)$$

Optical Flow: Top-down Strategy

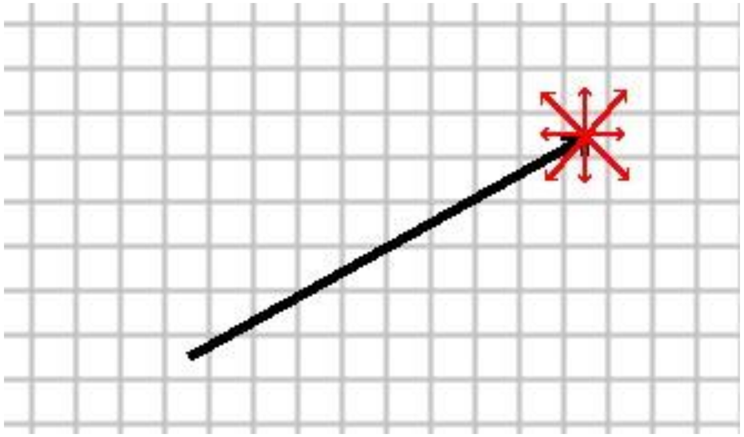
Algorithm (1/4 scale of resolution reduction)**:**

- Step 1: compute optical flow vectors for the highest level of the pyramid l (smallest resolution)
- Step 2: double the values of the vectors
- Step 3: first approximation: optical flow vectors for the (2i, 2j), (2i+1, 2j), (2i, 2j+1), (2i+1, 2j+1) pixels in the l-1 level are assigned the value of the optical flow vector for the (i,j) pixel from the l level



Optical Flow: Top-down Strategy (2)

- Step 4:
  - adjustment of the vectors of the l-1 level in the pyramid
  - method: detection of maximum one pixel displacements around the initially approximated position



- Step 5: smoothing of the optical flow field (Gaussian filter)

Dr. George Karraz, Ph. D.

Filtering the Size of the Detected Regions

- Small isolated regions of motion detected by the optical flow method are classified as **noise** and are eliminated with the help of **morphological operations:**
- Step 1: Apply the **opening:**
- Step 2: Apply the **closing:**

Contour Detection

- For demonstration purposes, the **contours** of the moving regions detected are outlined
- Method: the **Sobel edge detector**:
  - Compute the intensity gradient: $\nabla f(x,y) = \left(\dfrac{\partial f}{\partial x}, \dfrac{\partial f}{\partial y}\right) = \left(f_x, f_y\right)$ (5)

    using the Sobel masks:

    $$G_x = \frac{1}{4}\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, \quad G_y = \frac{1}{4}\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \quad (6)$$

  - Compute the magnitude of the gradient:

    $$M(x,y) = \left\|\nabla f(x,y)\right\| = \sqrt{f_x^{\;2} + f_y^{\;2}} \quad (7)$$

  - **if** $M(x,y) > threshold$ **then** *edge pixel*

    else *non-edge pixel*

A Block Diagram of the System