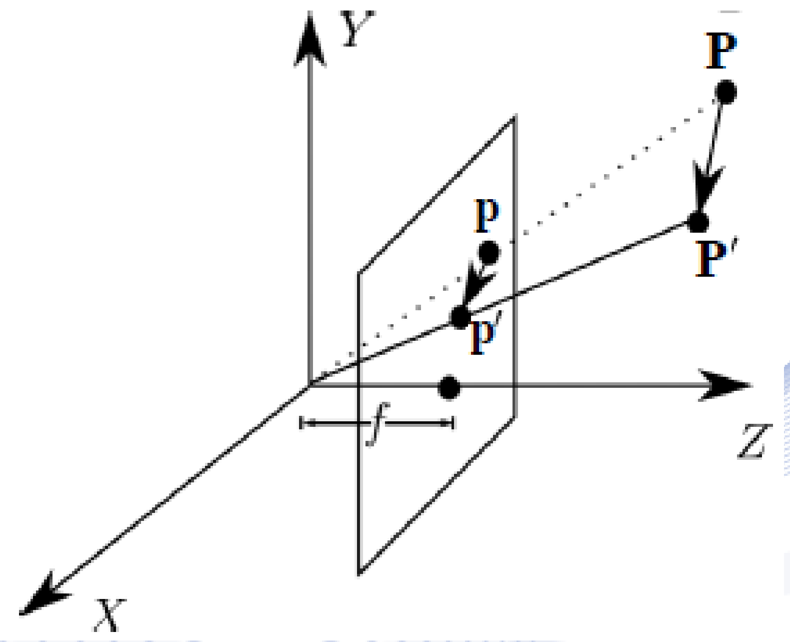
# Motion Estimation summary

Prof. Ioannis Pitas, S. Papadopoulos
Aristotle University of Thessaloniki
pitas@csd.auth.gr
www.aiia.csd.auth.gr
Version 3.3

Artificial Intelligence & Information Analysis Lab

# Two dimensional motion and apparent motion

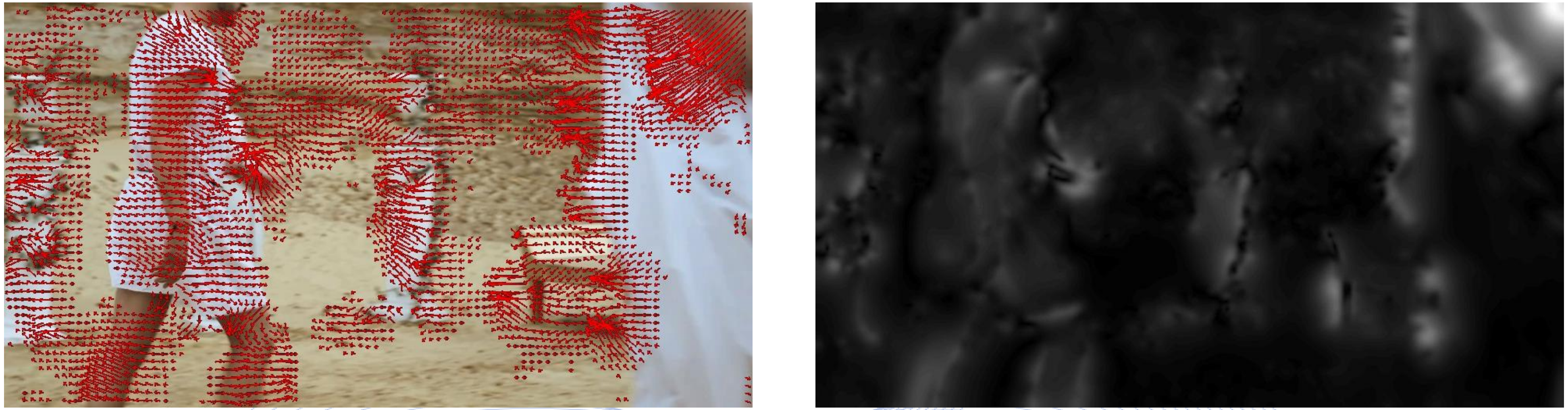
- 2D motion or projected motion is the perspective projection of the 3D motion on the image plane.
- Object point $\mathbf{P}$ at time $t$ moves to point $\mathbf{P}'$ at $t'$ and its perspective projection in the image plane from $\mathbf{p}$ to $\mathbf{p}'$.



# Two dimensional motion and apparent motion

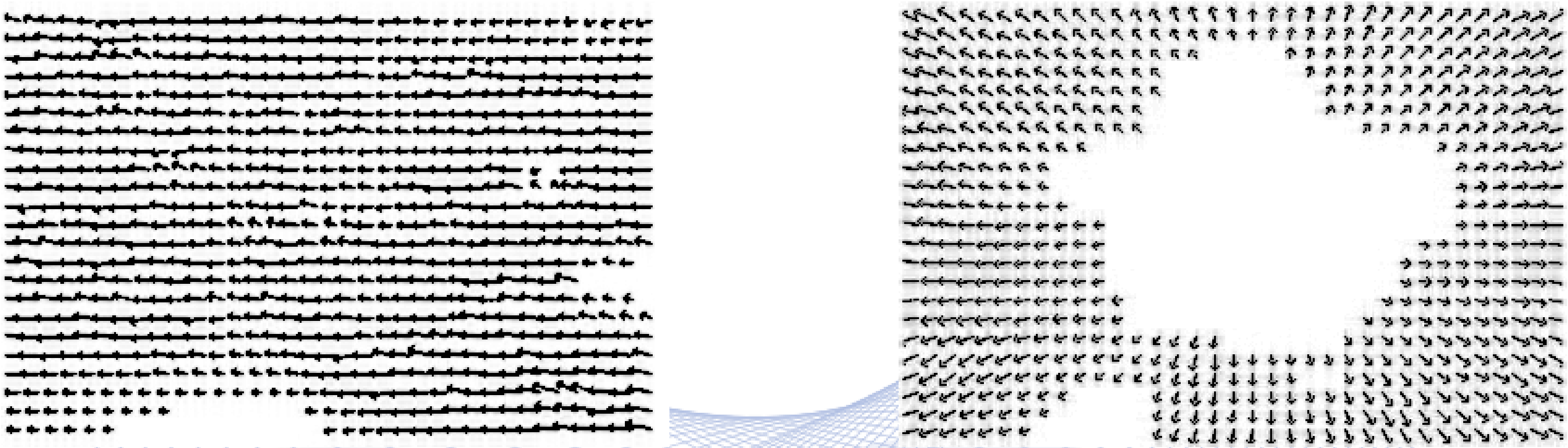
- ***Optical flow*** vector: the derivative of the correspondence vector: $[v_x, v_y]^T = [dx/dt\,, dy/dt]^T$.
- It describes the spatiotemporal changes of luminance $f_a(x, y, t)$.
- **Motion speed**: magnitude of the motion vector.
- The correspondence or optical flow vectors determine the apparent motion.

# Two dimensional motion and apparent motion



a) Motion field; b) motion speed.

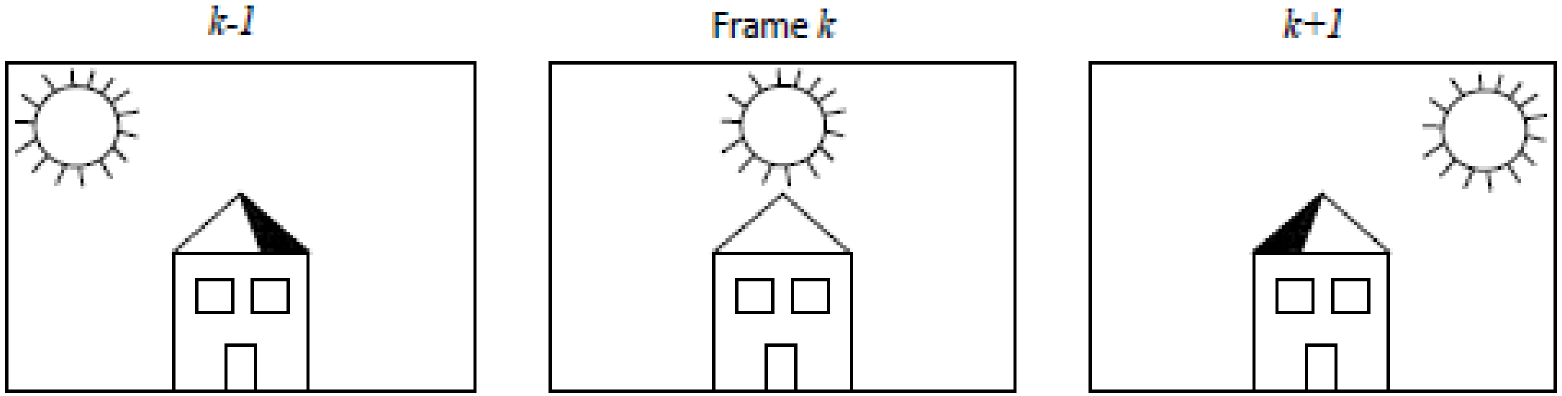
# Two dimensional motion and apparent motion



Global optical flow generated by a) camera pan and b) zoom.

# Two dimensional motion and apparent motion

- The optical flow field may be different from the 2D displacement field:
  - When the image has insufficient spatial information, the actual motion field is not observable.
  - Illumination changes alter luminance value of a static object.



# Three-dimensional motion models

- 3D solid object motion can be described by the affine transformation:

$$\mathbf{X}' = \mathbf{R}\mathbf{X} + \mathbf{T},$$

where $\mathbf{T}$ is a $3 \times 1$ translation vector:

$$\mathbf{T} = \begin{bmatrix} T_X \\ T_Y \\ T_Z \end{bmatrix}$$

and $\mathbf{R}$ is a $3 \times 3$ rotation matrix (various forms).

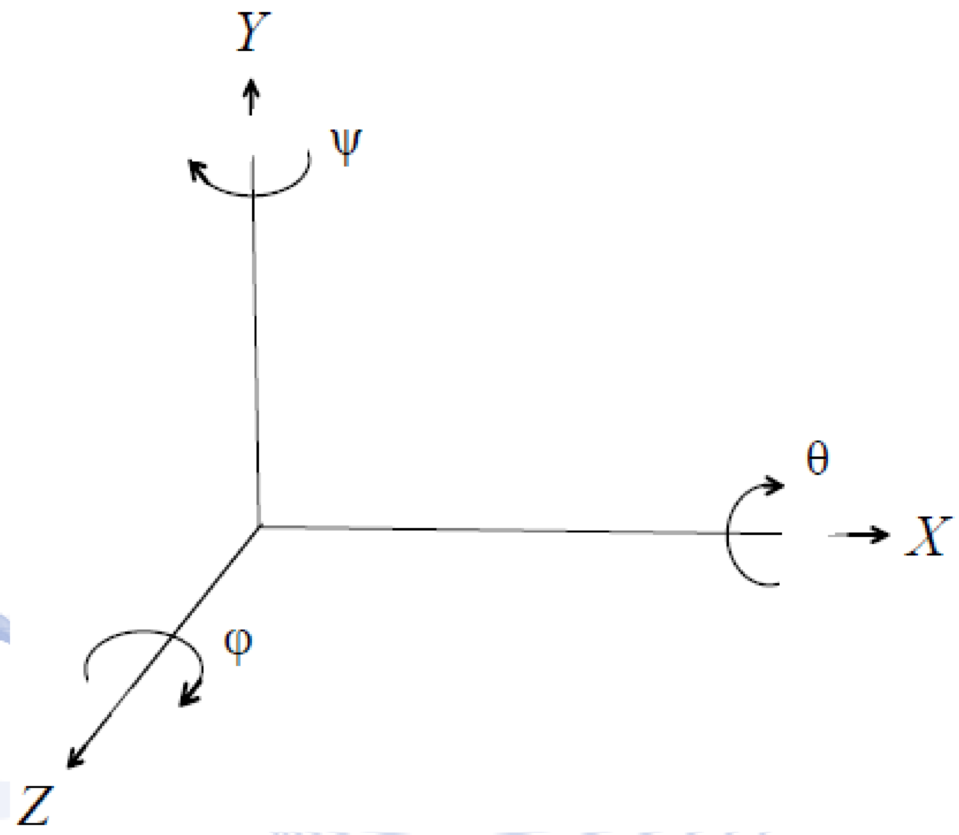
# Three-dimensional motion models

- In Cartesian coordinates, $\mathbf{R}$ can be described:
  - either by the Euler rotation angles about the three coordinate axes $X, Y, Z$.
  - or by a rotation axis and a rotation angle about this axis.
- The matrices describing the clockwise rotation around each axis in the three dimensional space, are given by:

$$\mathbf{R} = \mathbf{R}_Z \mathbf{R}_Y \mathbf{R}_X.$$

- Their order **does matter**.
- $\mathbf{R}$ is orthonormal, satisfying $\mathbf{R}^T = \mathbf{R}^{-1}$ and $det(\mathbf{R}) = \pm 1$.

# Three-dimensional motion models




Euler rotation angles.

$$\mathbf{R}_X = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix},$$

$$\mathbf{R}_Y = \begin{bmatrix} \cos\psi & 0 & \sin\psi \\ 0 & 1 & 0 \\ -\sin\psi & 0 & \cos\psi \end{bmatrix},$$

$$\mathbf{R}_Z = \begin{bmatrix} \cos\varphi & -\sin\varphi & 0 \\ \sin\varphi & \cos\varphi & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

# Two-dimensional motion models

- In many occasions, it is difficult to distinguish between camera and visualized object motion.
  - We consider that the camera remains static and the scene objects move:

$$\begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + \begin{bmatrix} T_X \\ T_Y \\ T_Z \end{bmatrix}.$$

  - From the 12 relevant parameters, only 6 are independent (3 rotation parameters and the 3 translation vector components).

Artificial Intelligence & Information Analysis Lab

# Two-dimensional motion models

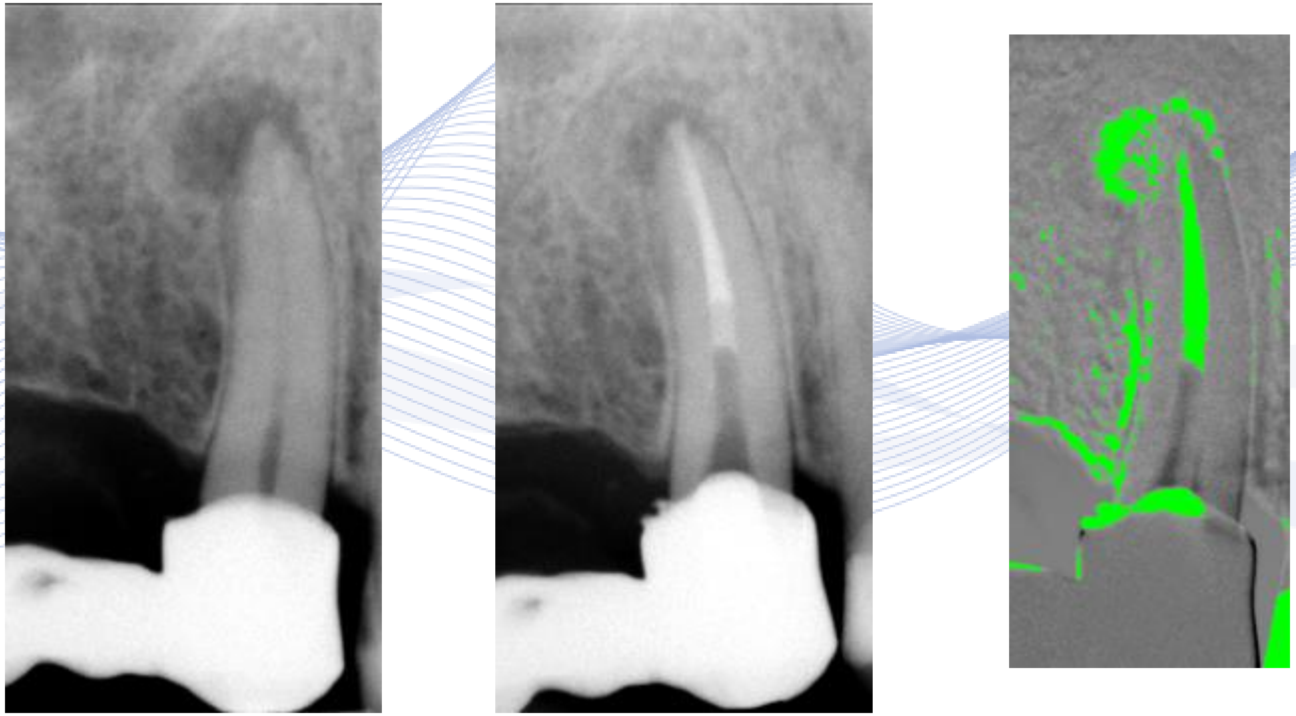
- The new image point coordinates must be calculated as projections of the world coordinates.
- Analytical expression of the new coordinates $[x', y']^T$ on the image plane as a function of the old position $[x, y]$ and depth $Z$:

$$x' = \frac{(r_{11}x+r_{12}y+r_{13}f)Z+T_Xf}{(r_{31}x+r_{32}y+r_{33}f)Z+T_Zf}, \quad y' = \frac{(r_{21}x+r_{22}y+r_{23}f)Z+T_Xf}{(r_{31}x+r_{32}y+r_{33}f)Z+T_Zf}.$$

# Two-dimensional motion models

- 2D affine mapping transformation: it describes 2D rotation, translation and scaling.
- It can be used for 2D image registration.

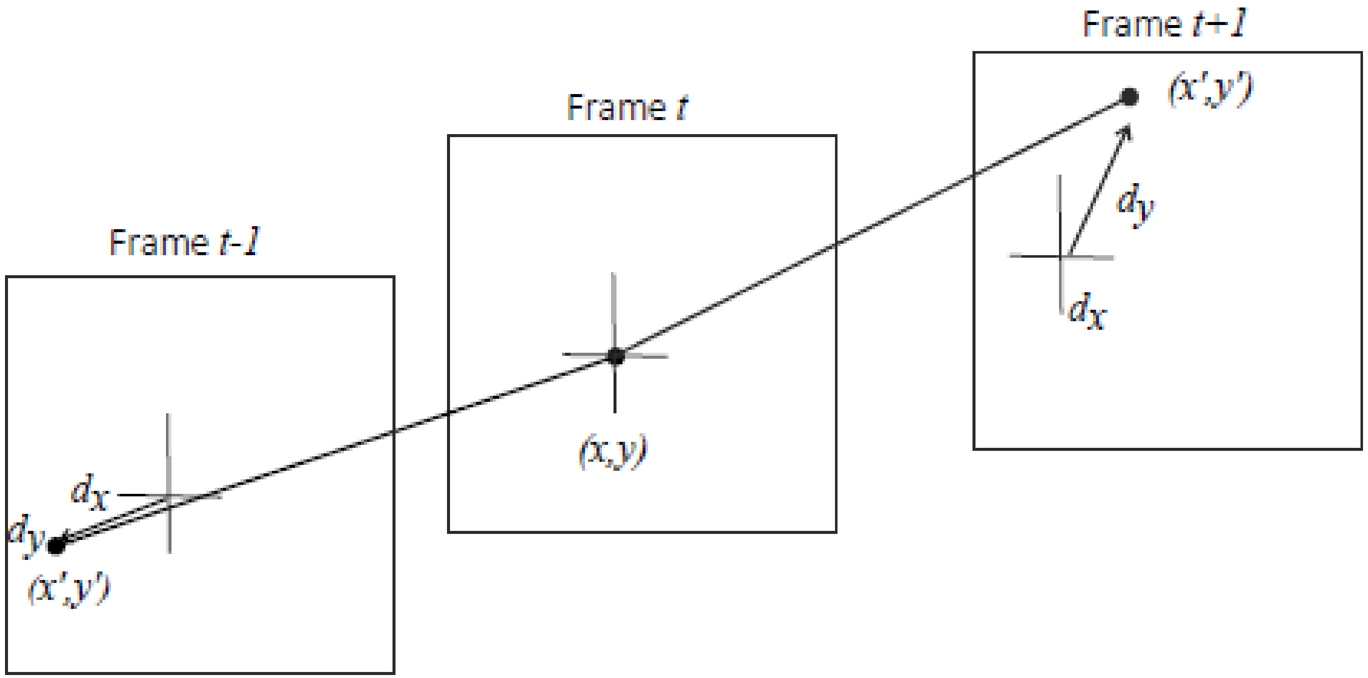
Subtractive radiography.



# Two-dimensional motion models

- 2D affine mapping transformation for image mosaicking.

# Estimation of two-dimensional correspondence vectors



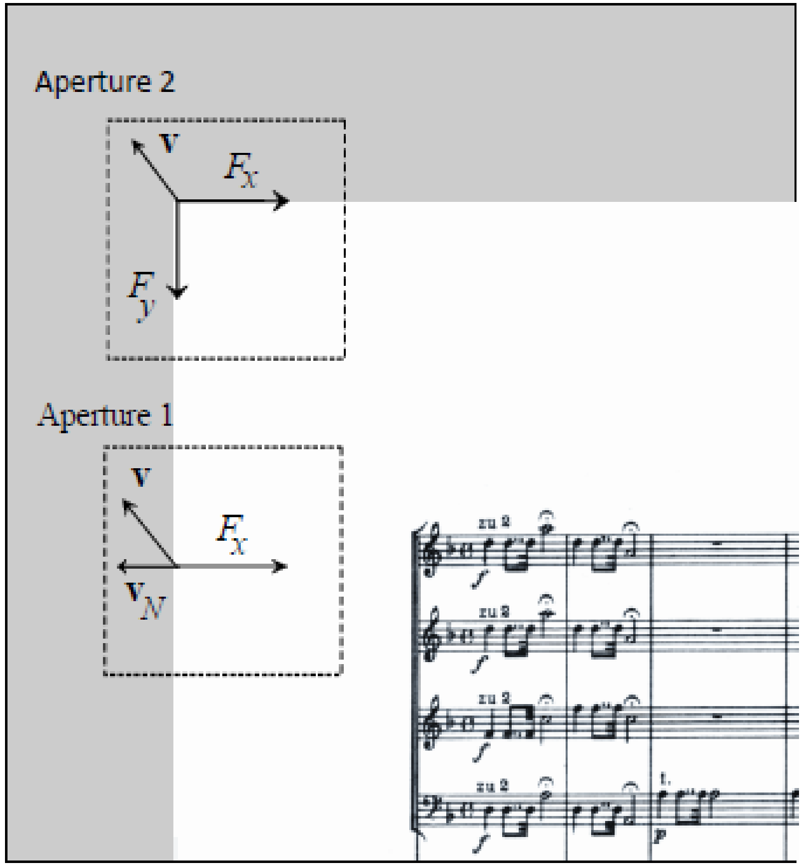

Forward and backward 2D motion estimation.

# Estimation of two-dimensional correspondence vectors

Object occlusion (right) and de-occlusion (left).

# Estimation of two-dimensional correspondence vectors

- Aperture problem: only local spatial information (within the camera aperture) is used for motion estimation.



# Quality metrics for motion estimation

- ***Peak Signal to Noise Ratio (PSNR)***: Metric for testing the quality of motion estimators' results, measured in $dB$:

$$PSNR = 10\log_{10}\frac{N\times M}{\sum[f(x,y,t)-f(x+dx(x,y),\ y+dy(x,y),\ \ t-1)]^2}.$$

- $N \times M$: video frame size in pixels.
- Video luminance scaled in the range $[0,1]$.
- $dx, dy$: the displacement components resulting from motion estimation at pixel $\mathbf{p} = [x, y]^T$.

# Quality metrics for motion estimation

- Denominator: the ***Displaced Frame Difference (DFD)*** between the target frame $t$ and the reference frame $t-1$.
- ***Motion field entropy***:

$$H = -\sum_{dx} p(dx) \log_2 p(dx) - \sum_{dy} p(dy) \log_2 p(dy).$$

- $p(dx), p(dy)$ : the probability density function (relative frequency) of the horizontal and vertical components of the displacement vector $\mathbf{d}(x,y) = [dx(x,y),\ dy(x,y)]^T$.

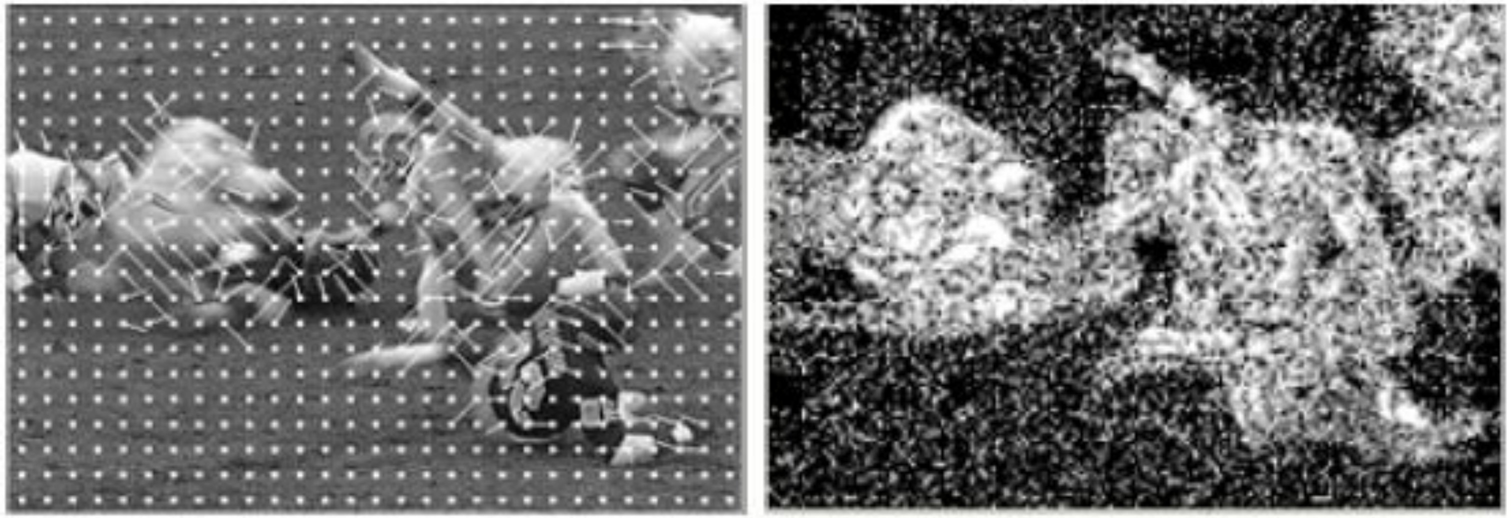
# Block matching

- Block displacement $\mathbf{d}$ can be estimated by minimizing the displaced section difference for selecting the optimal displacement $\mathbf{d} = [dx, dy]^T$:

$$\min_{dx,dy} E(\mathbf{d}) = \sum_{n_1} \sum_{n_2} \| f(n_1, n_2, t) - f(n_1 + dx, n_2 + dy, t - 1) \|.$$

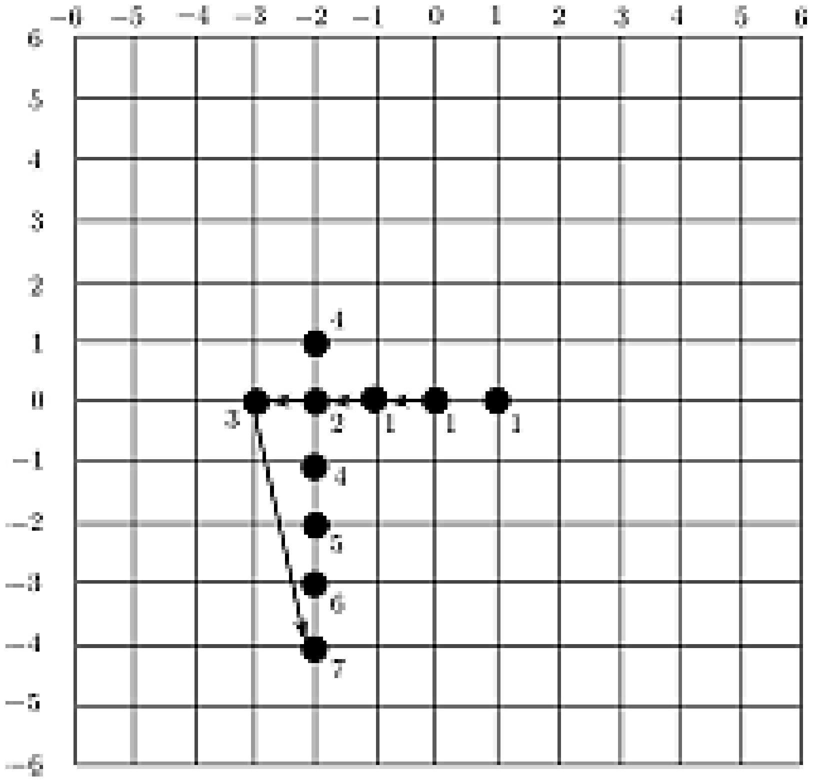
- $n_1, n_2$ are pixel coordinates.
- $L_1, L_2, L_p$ norms can be used for displaced frame difference estimation.

# Block matching



Sparse and dense motion fields.

# One dimensional search



- A two-step method for searching for the minimum of $E(\mathbf{d})$ along the horizontal and vertical directions:
  - 1st step. Search along the horizontal direction.
  - 2nd step. Based on the results of step 1, the minimum is searched for along the vertical direction.

# Phase correlation

- Relative image blocks displacement is calculated using a normalized cross-correlation function calculated on the 2D spatial or Fourier domain.
- ***Cross-correlation*** between two video frames of size $N_1 \times N_2$ at times $t$ and $t-1$:

$$r_{t,t-1}(n_1, n_2) = \sum_{k_1=0}^{N_1-1} \sum_{k_2=0}^{N_2-1} f(k_1, k_2, t) f(n_1 + k_1, n_2 + k_2, t-1) = f(n_1, n_2, t) ** f(-n_1, -n_2, t-1).$$

** denotes a 2D convolution.

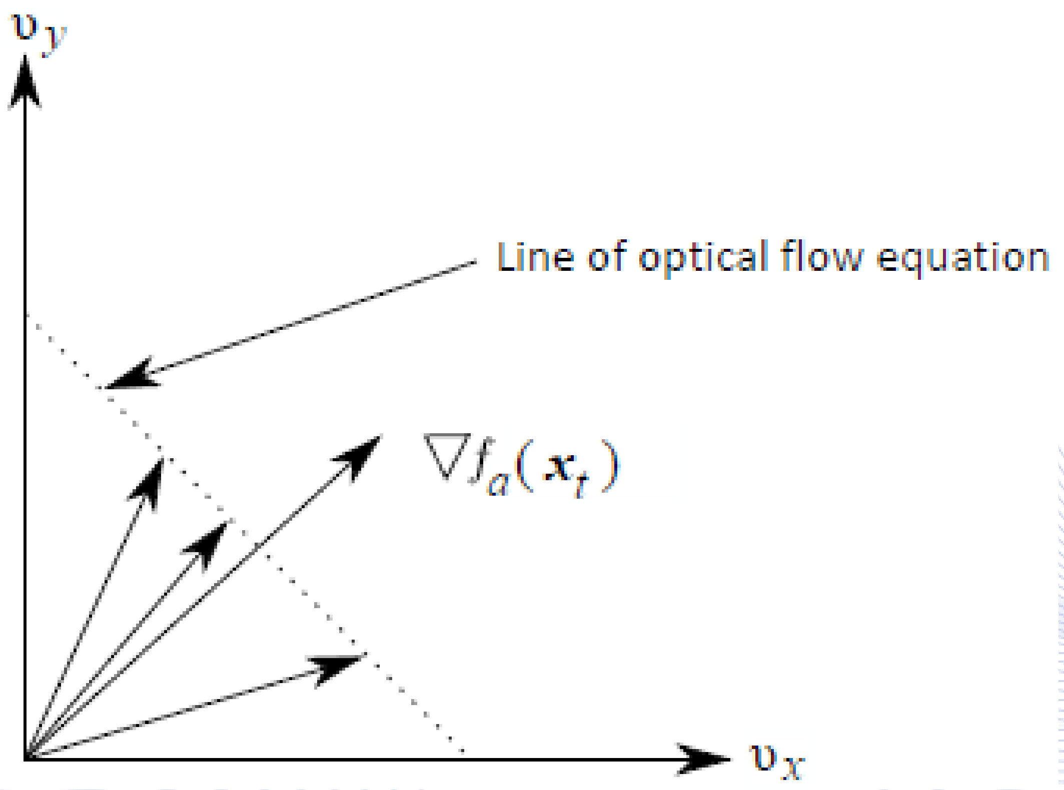
# Optical flow equation methods

- The continuous spatiotemporal video luminance $f_a(x,y,t)$, not $f_a(x,y,t)$ does not change along the object motion trajectory.
- For $\mathbf{x}_t = [x,y,t]^T$ on motion trajectory, the total derivative $\frac{df_a(\mathbf{x}_t)}{dt} = 0$ leads to ***optical flow equation (OFE)***:

$$\frac{\partial f_a(\mathbf{x}_t)}{\partial x} v_x(\mathbf{x},t) + \frac{\partial f_a(\mathbf{x}_t)}{\partial y} v_y(\mathbf{x},t) + \frac{\partial f_a(\mathbf{x}_t)}{\partial t} = 0.$$

- $\mathbf{x} = [x,y]$, $\mathbf{x}_t = [x,y,t]^T$, $v_x(\mathbf{x},t) = dx/dt$, $v_y(\mathbf{x},t) = dy/dt$.

# Optical flow equation methods



# OFE smoothing methods

- They are based on the assumption that object motion is smooth, so that correspondence motion fields change smoothly in space.
  - Small spatial gradients.

- Horn-Schunck method: searches for a motion field that both satisfies the OFE and has small spatial optical flow vector changes.

# OFE smoothing methods

- Satisfaction of OFE requires minimization of the squared error of:

$$E_1\big(\mathbf{v}(\mathbf{x},t)\big) = \nabla f_\alpha(\mathbf{x}_t)\cdot \mathbf{v}^T(\mathbf{x},t) + \frac{\partial f_a(\mathbf{x}_t)}{\partial t}.$$

- Spatial changes in the velocity vector field can be quantified by:

$$E_2^2\big(\mathbf{v}(\mathbf{x},t)\big) = \|\nabla v_x(\mathbf{x},t)\|^2 + \left\|\nabla v_y(\mathbf{x},t)\right\|^2 =$$
$$= \left(\frac{\partial v_x}{\partial x}\right)^2 + \left(\frac{\partial v_x}{\partial y}\right)^2 + \left(\frac{\partial v_y}{\partial x}\right)^2 + \left(\frac{\partial v_y}{\partial y}\right)^2.$$

# OFE smoothing methods

- OFE smoothing minimizes $E_1^2(\mathbf{v}), E_2^2(\mathbf{v})$ wrt the velocity vector components $(v_x, v_y)$ at each point $\mathbf{x} = [x, y]^T$:

$$\min_{\mathbf{v}(\mathbf{x},t)} \int_{\mathcal{A}} \left( E_1^2(\mathbf{v}) + \lambda E_2^2(\mathbf{v}) \right) dx.$$

$\lambda$: chosen heuristically parameter controling motion field smoothing.

# Neural Optical Flow estimation

- Optical flow estimation by using Convolutional Neural Networks.
- High accuracy, dense flow field, fast implementations.
- Supervised methods:
  - Highest accuracy;
  - Ground truth for real world video sequences is required.
- Unsupervised methods:
  - Lower, but comparable accuracy;
  - No need for optical flow ground truth.

# Neural Optical Flow estimation

***Flownet***: Supervised NN optical flow estimation.

- Foundation stone for almost all later supervised networks.
- FlowNetS (**S**imple):
  - A single network branch.
  - Refinement module upscales conv6's output using outputs from various intermediate stages.
  - Two consecutive input frames, concatenated in the channel dimension.

# Neural Optical Flow estimation

FlowNetC (**C**orrelation):

- two separate branches extracting features for each input image;
- they are later merged into one branch by correlating the extracted feature maps:

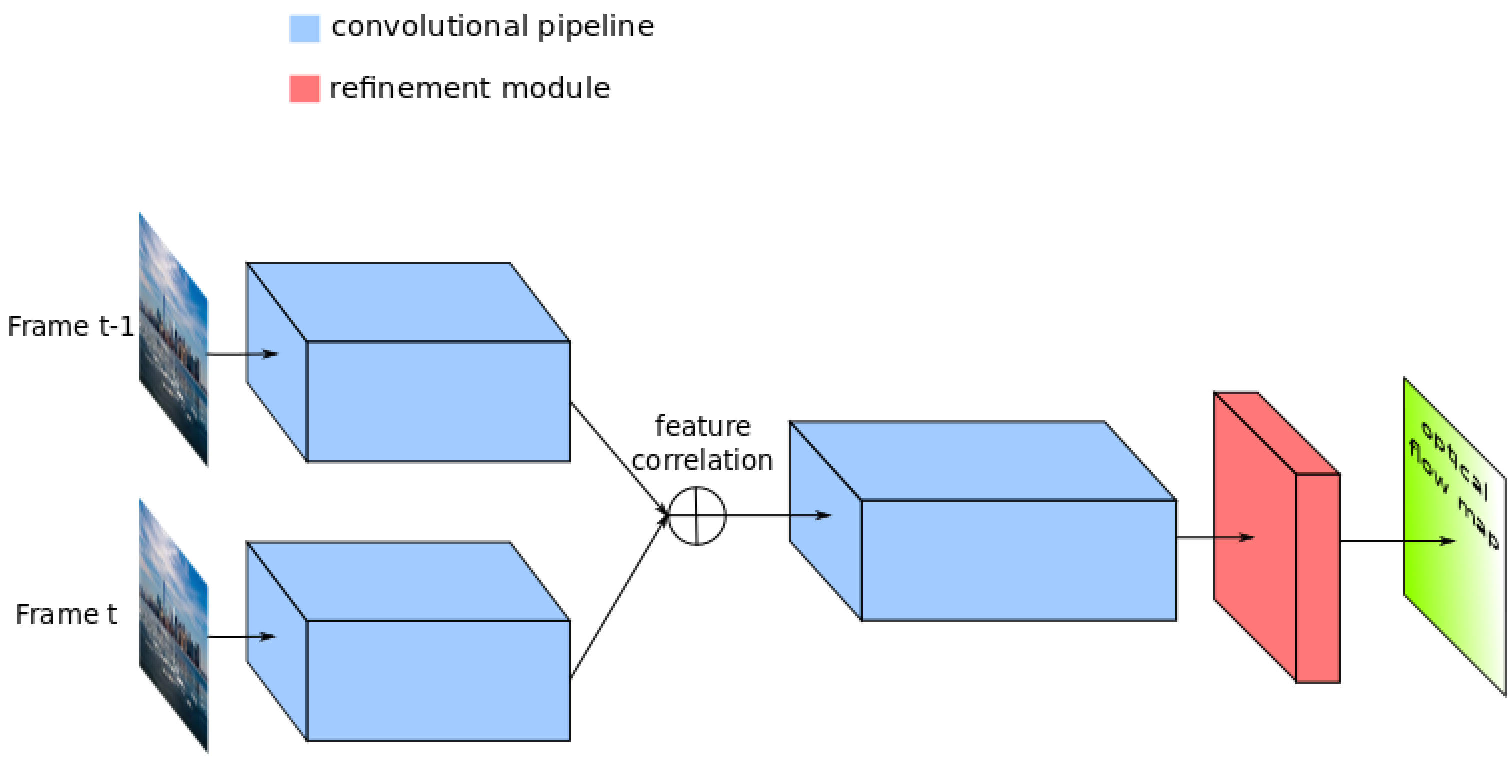
$$r_{f_1 f_2}(n_1, n_2) = f_1(n_1, n_2) ** f_2(-n_1, -n_2).$$

- $f_1, f_2$: $(2k+1) \times (2k+1)$ 2D feature maps.



[8]

# Neural Optical Flow estimation




[DOS2015]

# Object detection and Tracking

- Motion estimation estimates motion vectors on entire video frames.
- Object tracking relies on:
  - Object detection on a video frame.
  - Tracking of this object (essentially estimating its motion) over subsequent video frames.

# Object Detection and Tracking


- Problem statement:
  - To detect an object (e.g. human face) that appear in each video frame and localize its ***Region-Of-Interest (ROI)***.
  - To track the detected object over the video frames.

# Object detection and Tracking

- Tracking associates each detected object ROI in the current video frame with one in the next video frame.

- Therefore, we can describe the ***object ROI trajectory*** in a video segment in $(x, y, t)$ coordinates.

# Object Detection and Tracking

- **Tracking failure** may occur, i.e.,
  - after occlusions;
  - when the tracker drifts to the background or to another object.
- In such cases, **object re-detection** is employed.
- However, if any of the detected objects coincides with any of the objects already being tracked, the former ones are retained, while the latter ones are discarded from any further processing.

# Object Detection and Tracking

- **Periodic object re-detection** can be applied to account for new faces entering the camera's field-of-view.

- **Forward and backward tracking**, when the entire video is available.

# References

**[DOS2015]** Dosovitskiy, Alexey, et al. "Flownet: Learning optical flow with convolutional networks." *Proceedings of the IEEE international conference on computer vision*. 2015.

**[ILG2017]** Ilg, Eddy, et al. "Flownet 2.0: Evolution of optical flow estimation with deep networks." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017.

**[HUI2018]** Hui, Tak-Wai, Xiaoou Tang, and Chen Change Loy. "Liteflownet: A lightweight convolutional neural network for optical flow estimation." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018.

**[RAN2017]** Ranjan, Anurag, and Michael J. Black. "Optical flow estimation using a spatial pyramid network." *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017.

**[ZHI2018]** Yin, Zhichao, and Jianping Shi. "Geonet: Unsupervised learning of dense depth, optical flow and camera pose." *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018.

**[RAN2019]** Ranjan, Anurag, et al. "Competitive collaboration: Joint unsupervised learning of depth, camera motion, optical flow and motion segmentation." *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019.

**[ZOU2018]** Zou, Yuliang, Zelun Luo, and Jia-Bin Huang. "Df-net: Unsupervised joint learning of depth and flow using cross-task consistency." *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018.

**[ZHOU2019]** Zhu, Alex Zihao, et al. "Unsupervised event-based learning of optical flow, depth, and egomotion." *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019.

# Q & A

Thank you very much for your attention!

Contact: Prof. I. Pitas
pitas@csd.auth.gr