Motion Analysis

Image and video processing

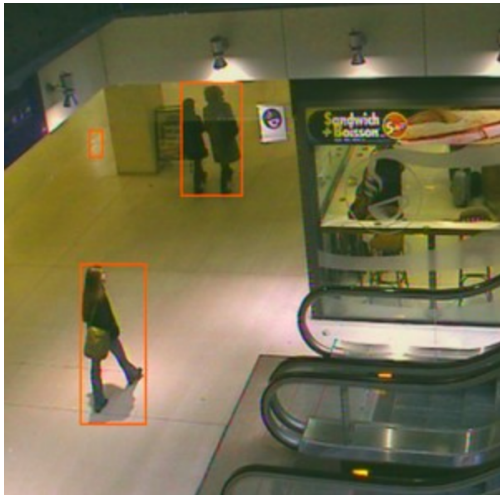
Samia Bouchafa-Bruneau

M2 E3A - UEVE/Université Paris-Saclay

- **Fixed camera**
  - Video monitoring
  - Compression of image sequences
  - Medical Imaging
- **Moving cameras**
  - ADAS
  - Mobile Robots
  - Video Registration

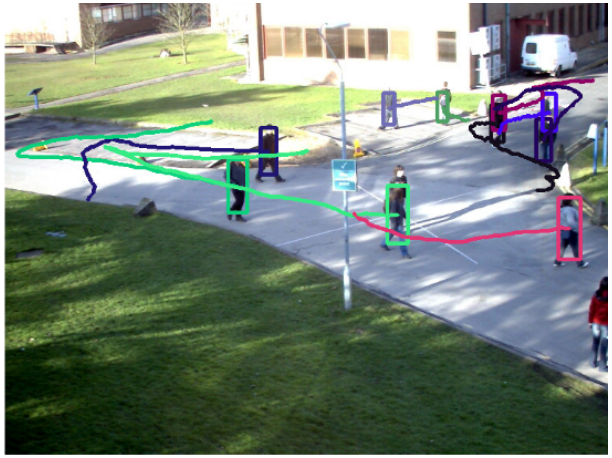
Generalities

Applications

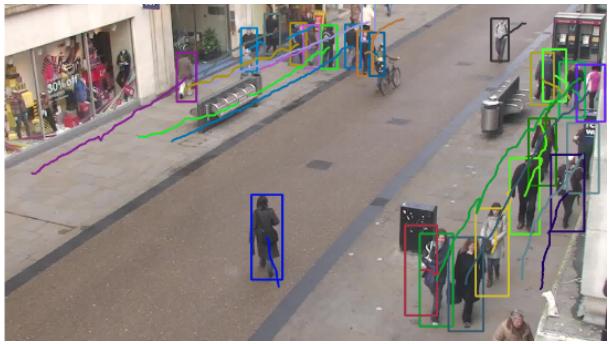


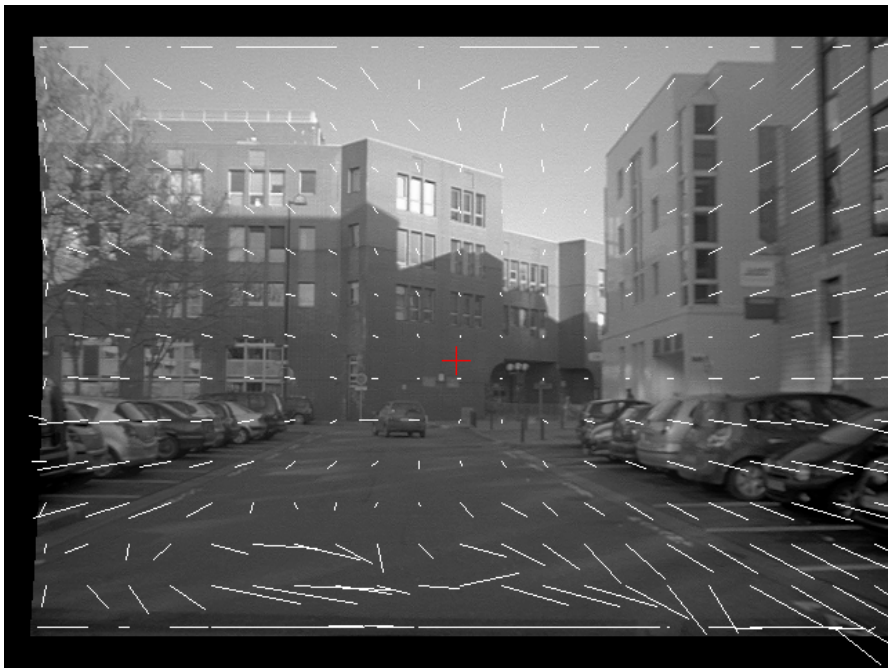
Video monitoring (ADCIS)

Free navigable space detection (IBISC)

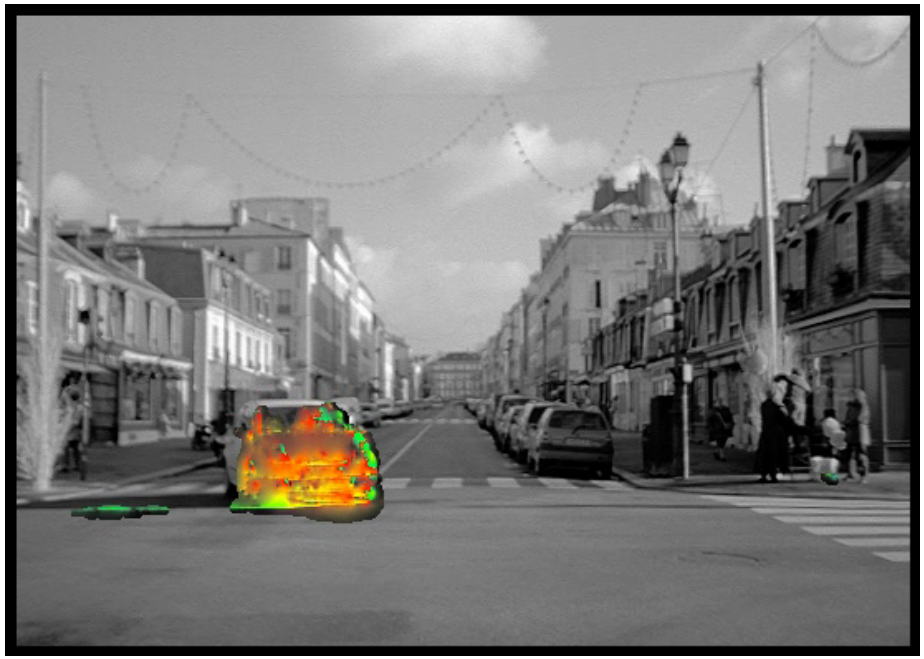


Pedestrian tracking (CEA)





Egomotion Estimation
(IBISC/LIVIC)



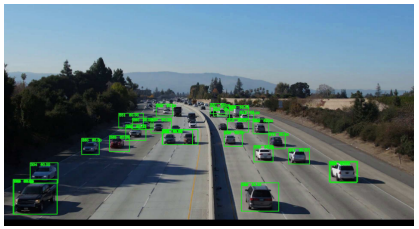
Obstacle Detection
IBISC/LIVIC)

Visual Odometry (LIVIC)

- **Two main issues :**
  - Matching
    * Feature extraction + Finding correspondences ?
  - Reconstruction
    * Starting from a given number of correspondents (couples), what can we say about the 3D motion of the camera and the structure of the scene (3D reconstruction) ?
- **Difference between motion and stereovision :**
  - Correspondence
    * Disparities are smaller when considering two successive images
  - Reconstruction
    * The relative 3D motion between the camera and the scene is not necessarily rigid.

Motion analysis main steps :

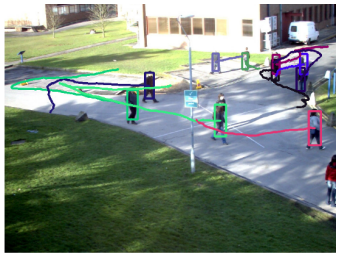
- **Detection** : highlighting of the (primitive) zones affected by the movement.
- **Estimation** : 2D motion estimation (amplitude, orientation), dense or sparse.
- **Segmentation** : label according to criterion of similarity of movement and of connectedness.
- **Tracking** : reconstruct the 2D trajectory of moving objects.



Detection

Estimation

Segmentation



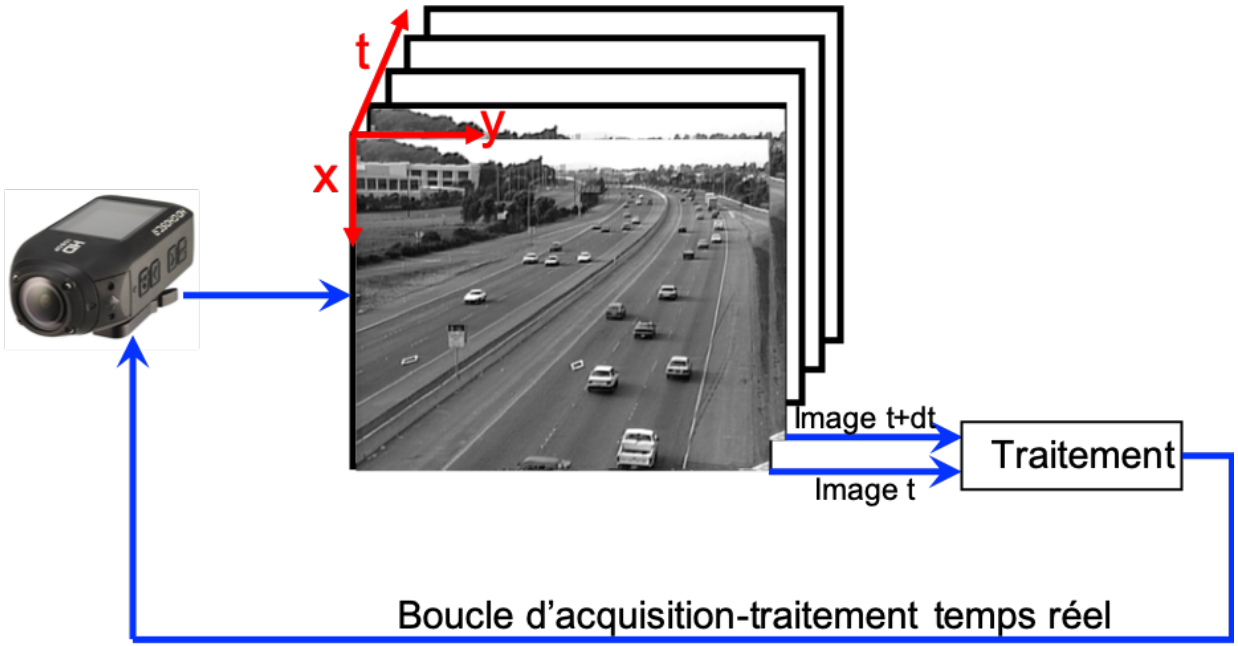
Tracking

Motion detection

- **Input** $\rightarrow$ image sequence
- **Main difficulties :**
  - How to build relevant/reliable information from pixels ?
  - Acquisition-processing loop $\rightarrow$ problems with expansive algorithms
  - Fixed Camera : how to distinguish between changes that are due to motion and changes that are caused by noise or illumination and contrast changes.
  - Moving Camera : how to distinguish between changes that are due to motion and changes that are caused by moving objects.
- **First Hypotheses :**
  - Fixed Camera
  - Brightness Invariance during displacement
  - Temporal Changes are due to motion

Motion detection

- **Difficulty (vicious circle)** :
  - The capture of the next image is only possible at the end of the processing
  - If the processing is long, the capture is delayed item If the capture is delayed, the shift between images is greater making the comparison difficult and the processing longer !



Motion detection

- **How existing methods work ?**
  - Two successive images : Image 1 = Image at t and Image 2 = Image at $t + dt$
  - Current image and a reference image : Image 1 = Image at t and Image 2 = Background Image
- **Primitives to extract : same as stereovision**
  - point
  - point + spatial context (neighbors)
  - region
  - gradient, edges, contour
- **Features : same as stereovision**
  - point : brightness
  - point + spatial context : mean, median,...
  - regions : surfaces, other moments of inertia,...
  - contours : length, orientation,...

Motion detection

- **Drawbacks :**
  - Sensitivity to global brightness changes
  - How to choose suitable thresholds ?
  - Point : depend directly on brightness, sensitive to changes
  - Point + spatial context :arbitrary neighborhood size ?
  - Region : all features depend (directly or indirectly) on brightness
  - Contours : what is a contour ? Several definitions...

Motion detection

- **Point :**
  - Frame difference
  - Convolve with $\begin{bmatrix} 1 & -1 \end{bmatrix}$ which is a basic temporal derivative filter

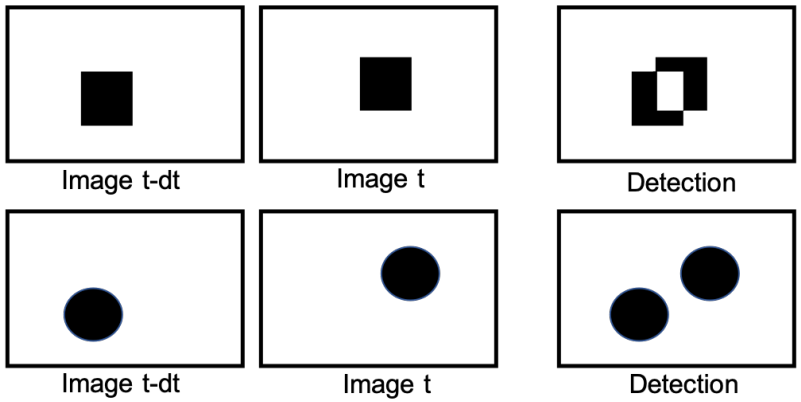
$$D(x,y) = \begin{cases} 1 \text{ if } |E(x,y,t) - E(x,y,t+dt)| > seuil \\ 0 \text{ else} \end{cases}$$

- **Point + spatial context**
  - Average filtering then frame difference

$$D(x,y) = \begin{cases} 1 \text{ if } \frac{1}{N} \sum_{(x,y) \in V(N)} |E(x,y,t) - E(x,y,t+dt)| > seuil \\ 0 \text{ else} \end{cases}$$

Motion detection

- **Nature of used images :**
  - Two successive images : but reconstruction problems (complete shape of moving objects)
  * **Weak Amplitudes (recovering)** : only the temporal edge is detected.
  * **High Amplitudes (no recovering)** : shape is redoubled.



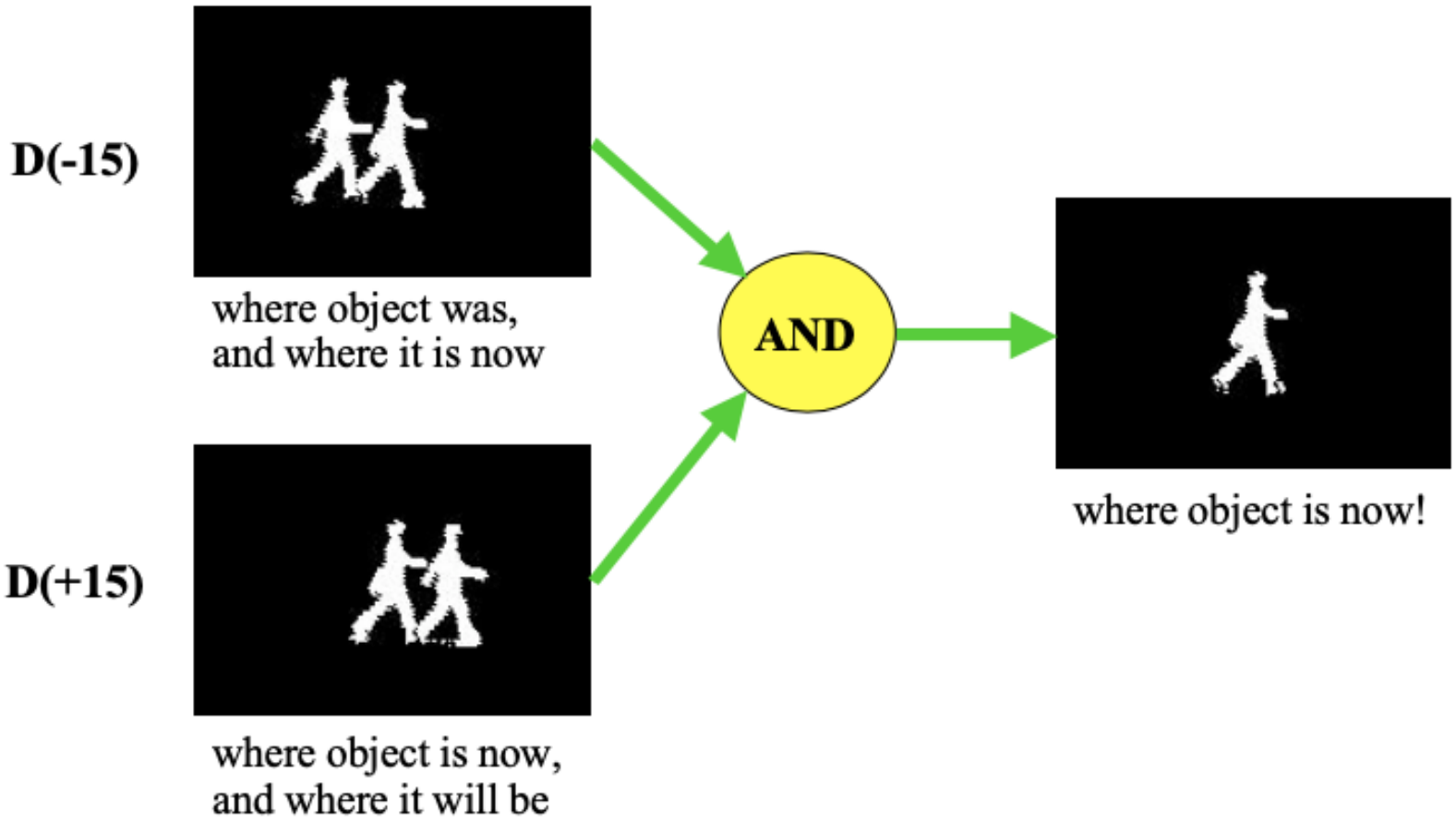
Motion detection

- **Problem of shape recovering**



Motion detection

- **Solution 1** : increase the temporal window



Motion detection

- **Solution 2 :** use a background reference image
  - Create a background reference image that corresponds to the static scene (without moving objects) by temporal averaging

$$R_t(\mathbf{p}) = \frac{1}{N} \sum_{i=1}^{N} E(\mathbf{p}, t - idt)$$

  - Decision : $D(\mathbf{p}) = 1$ if $|R_t(\mathbf{p}) - E(\mathbf{p}, \mathbf{t})| > threshold$
  - Require to store N images in memory.
  - Require using a 1er order low pass filter of input $E(\mathbf{p}, t)$ and output $R_t(\mathbf{p})$:

$$R_t(\mathbf{p}) = f \times R_{t-1}(\mathbf{p}) + (1 - f) \times E(\mathbf{p}, t)_{0 \leq f \leq 1}$$

Motion detection

- **Case of using direction of gradients as basic primitives**
  - One reference image per gradient direction d :

$$R_t(\mathbf{p},d) = f \times R_{t-1}(\mathbf{p},d) + (1-f) \times E(\mathbf{p},t)$$

  - $E(\mathbf{p},t) = 1$ if the direction appears at point $\mathbf{p}$
  - Decision :
    if $R_t(\mathbf{p},d) > 1 - S \rightarrow$ the direction $E(\mathbf{p},t)$ could be added to the associated reference frame.

Motion detection

- **Time Integration of a direction in the reference frame**
  - We suppose that the initial conditions are null $\Longrightarrow$ $R_0(\mathbf{p},d) = 0$ $\forall d, \forall \mathbf{p}$
  - If $E(\mathbf{p},t) = 1$ for $t = 1,..K$ (K images) then we obtain :

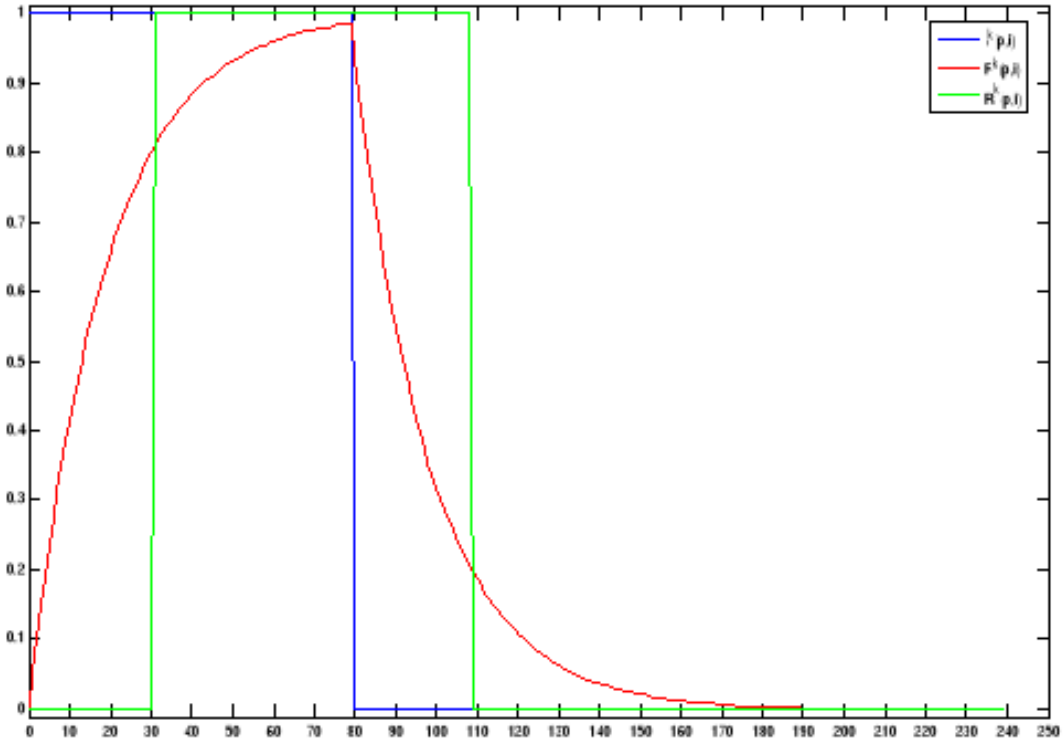
$$R_K(\mathbf{p},d) = (1-f)\sum_{i=0}^{K-1} f^i = (1-f^K)$$

  - The direction is incorporated in the reference after K images if $(1-f^K) = 1-S$:

$$f = \exp\left(\frac{\log(S)}{t_{int} \times f_{acq}}\right)$$

  - $t_{int} \times f_{acq} = K$ where t_{int} is the integration time in the reference and f_{acq} the images acquisition frequency.

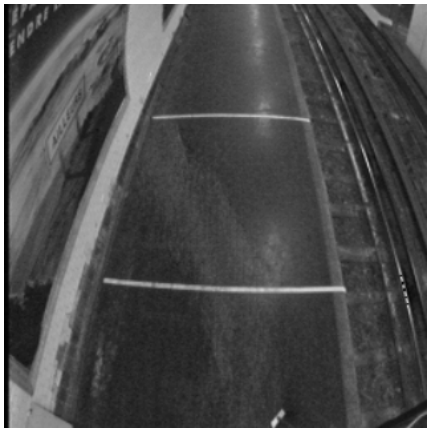
Motion detection



A direction in $(\mathbf{p}, d)$ appears during 80 images for an integration period of $K = 30$ images and considering a threshold $S = 0.2$

Motion detection

Example of resulting background image by temporal averaging



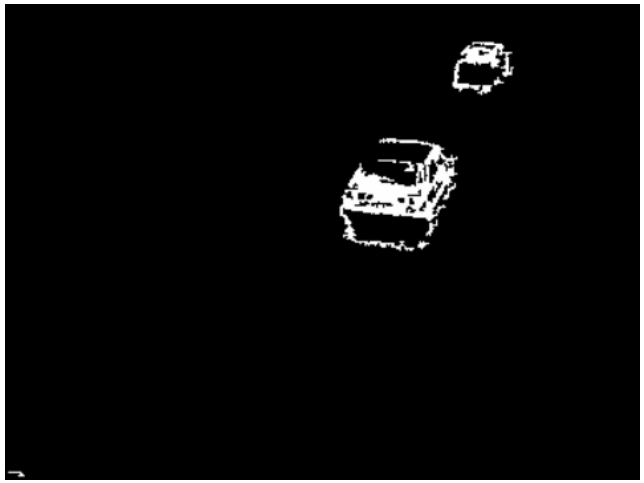
Motion detection

Example of resulting background image using contours

Human Detection on automatic mobile bridges

Motion detection

Example of resulting background image using contours



Presence Detection on mobile bridges

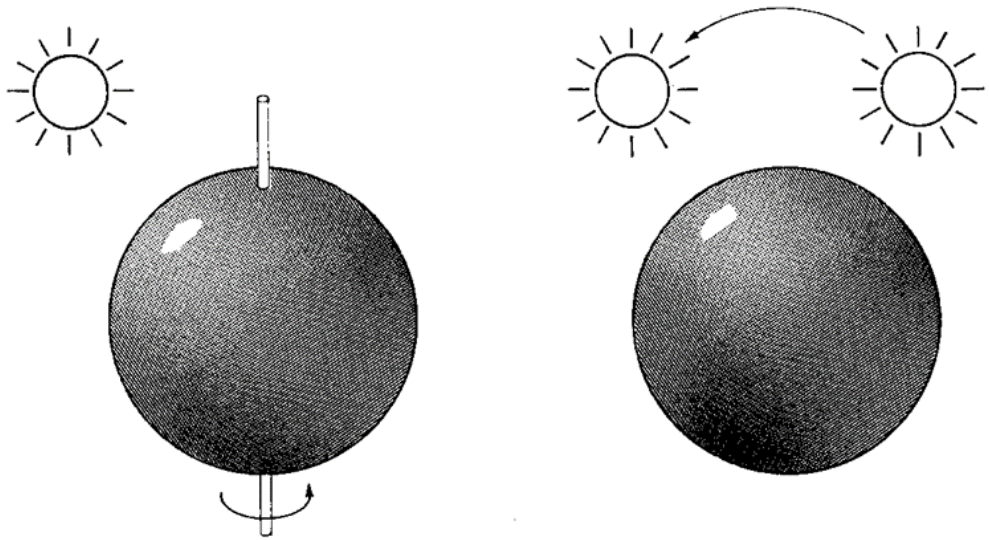
Car Detection on automatic mobile bridges

Motion estimation

- **Hypotheses**
  - Small Displacements
    - Differential Approaches $\rightarrow$ dense motion field
  - Large Displacements
    - Matching Strategies $\rightarrow$ sparse motion field

Motion estimation

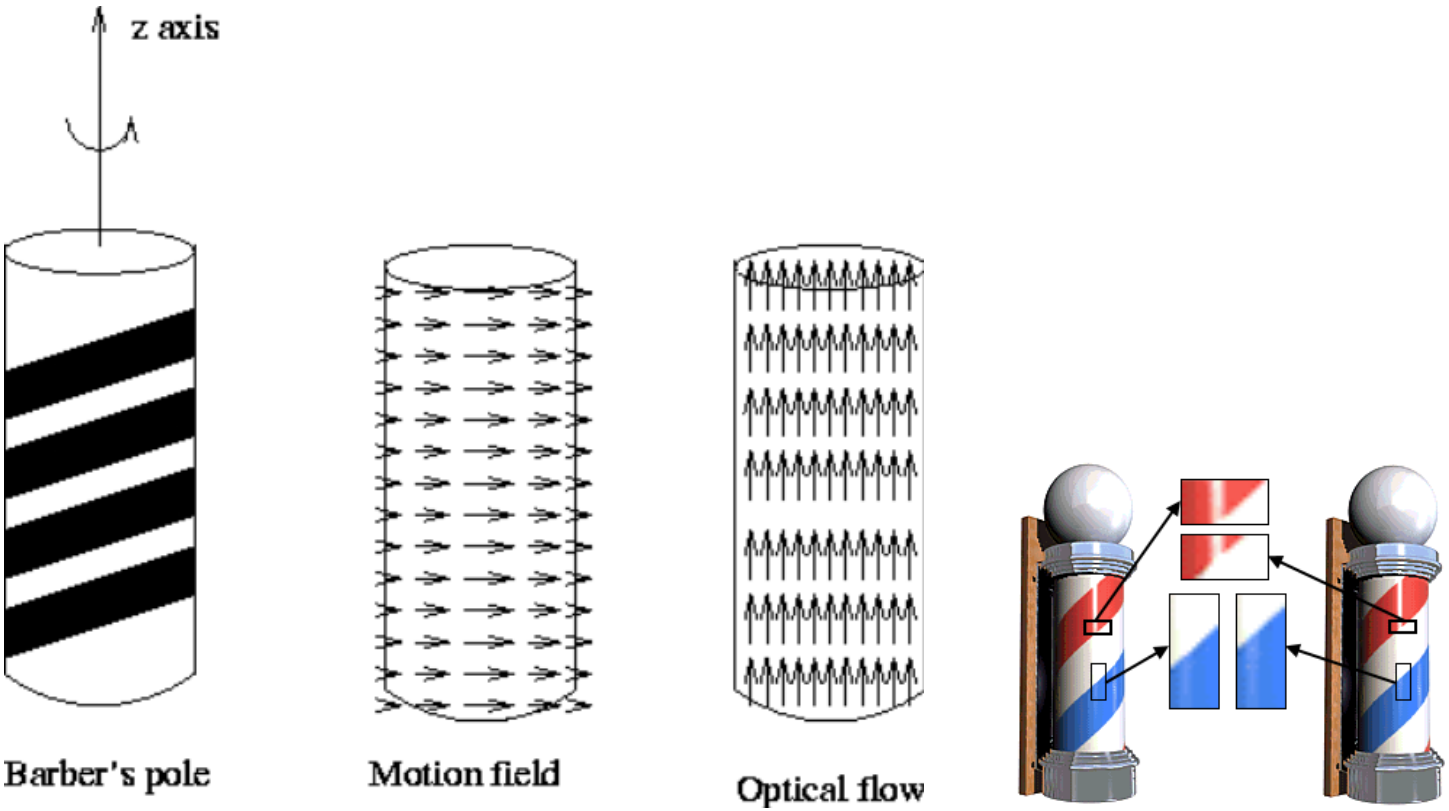
- **Optical flow**
  - Variations of Brightness due to motion in a small time interval.
- **Difference between real motion and optical flow**



1. A sphere rotates (constant illumination). Flow = 0 ; Real motion$\neq$0
2. A sphere is fixed but the illumination changes (changes in light source). Flow$\neq$0 ; real motion = 0.

Estimation de mouvement

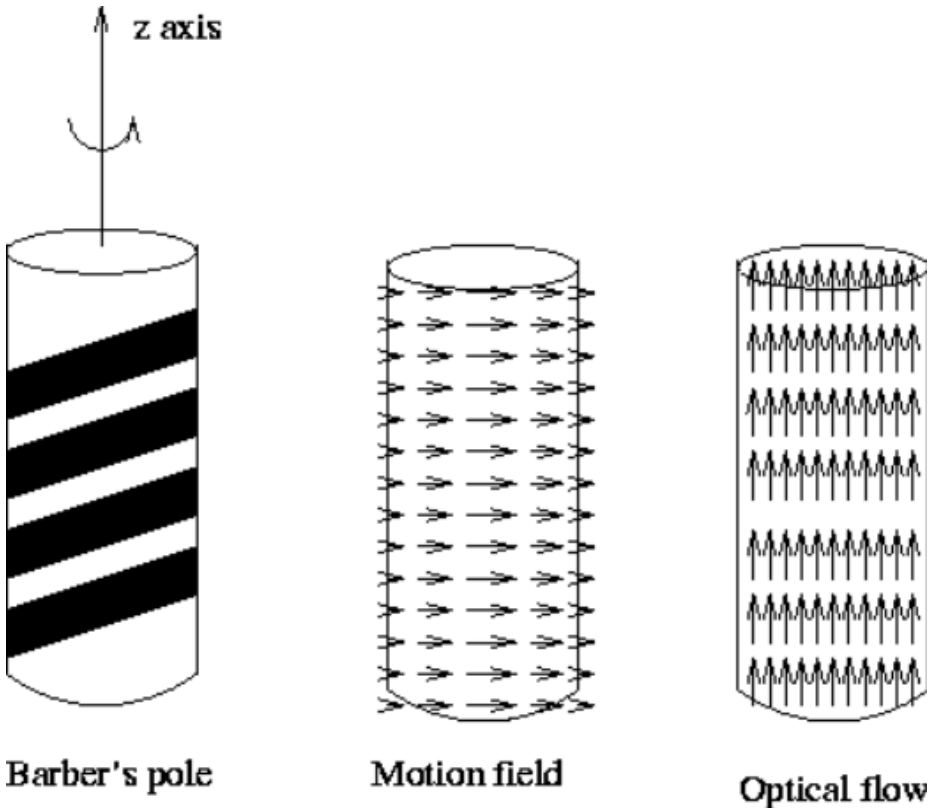
- **aperture problem**



See : https://www.opticalillusion.net/optical-illusions/the-barber-pole-illusion/

Motion estimation

- ***Aperture problem***



Source : http ://www.sandlotscience.com/Ambiguous/barberpole.htm

- **Brightness invariance hypothesis for each moving point** :

$$\frac{dE}{dt} = 0$$
$$\frac{dE(x(t),y(t),t)}{dt} = \frac{\partial E}{\partial x}\frac{\partial x}{\partial t} + \frac{\partial E}{\partial y}\frac{\partial y}{\partial t} + \frac{\partial E}{\partial t}$$

- Let us consider a point (x, y) that moves with a displacement of (dx, dy). Its brightness does not change $\rightarrow$ $E(x, y, t) = E(x + dx, y + dy, t + dt)$
- First order decomposition using Taylor series, when omitting greater than 1 order terms, gives :

$$E(x + dx, y + dy, t + dt) = E(x, y, t) + dx\frac{\partial E}{\partial x} + dy\frac{\partial E}{\partial y} + dt\frac{\partial E}{\partial t}$$

- Let consider :

$$\begin{cases} u = \frac{dx}{dt} \\ v = \frac{dy}{dt} \end{cases}$$

- For each point, we can write one equation of two unknowns u and v (velocity vector $\mathbf{w}$ components'). We call this equation : motion constraint equation :

$$\frac{\partial E}{\partial x}u + \frac{\partial E}{\partial y}v + \frac{\partial E}{\partial t} = 0$$
$$\Leftrightarrow \nabla \mathbf{E} \cdot \mathbf{w} = -\frac{\partial E}{\partial t}$$

$$\text{with } \nabla \mathbf{E} = \left(\frac{\partial E}{\partial x}, \frac{\partial E}{\partial y}\right)^T \text{ and } \mathbf{w} = (u, v)^T$$

- **Velocity vectors smoothing**
  - take advantage of the spatial and temporal redundancies.
  - velocity vectors of moving points do not vary too much $\rightarrow$ smoothing constraint $\rightarrow$ gradual change of the gradient magnitude of velocity vectors :

$$\left(\frac{\partial u}{\partial x}\right)^2 + \left(\frac{\partial u}{\partial y}\right)^2 \text{ et } \left(\frac{\partial v}{\partial x}\right)^2 + \left(\frac{\partial v}{\partial y}\right)^2$$

- Error Minimization :

$$\begin{aligned} \varepsilon_b &= \frac{\partial E}{\partial x}u + \frac{\partial E}{\partial y}v + \frac{\partial E}{\partial t} \\ \varepsilon_c^2 &= \left(\frac{\partial u}{\partial x}\right)^2 + \left(\frac{\partial u}{\partial y}\right)^2 + \left(\frac{\partial v}{\partial x}\right)^2 + \left(\frac{\partial v}{\partial y}\right)^2 \end{aligned}$$

- To simplify, we can write :

$$\begin{aligned} \varepsilon_b &= E_x u + E_y v + E_t \\ \varepsilon_c^2 &= u_x^2 + u_y^2 + v_x^2 + v_y^2 \end{aligned}$$

- The total error to minimize is then : $\varepsilon = \iint \left(\alpha^2 \varepsilon_b^2 + \varepsilon_c^2\right) dxdy$

- **Euler condition :**
  - Variations calculus allow us to find the *extrema* of a functional $I(f)$. We consider that $I(f)$ could be expressed as :

$$I(f) = \int_{x_1}^{x_2} F(x, f, f')dx$$

- Euler-Lagrange equations are necessary condition for f to be an textitextremum. Let : $F_f = \frac{\partial}{\partial f}F$ et $F_{f'} = \frac{\partial}{\partial f'}F$ Then :

$$F_f - \frac{d}{dx}F_{f'} = 0$$

- In our case, the functional to minimize is :

$$\iint F(u, v, u_x, u_y, v_x, v_y)$$

- The generalization of Euler equations in 2D gives :

$$\begin{aligned} F_u - \frac{\partial}{\partial x} F_{u_x} - \frac{\partial}{\partial y} F_{u_y} &= 0 \\ F_v - \frac{\partial}{\partial x} F_{v_x} - \frac{\partial}{\partial y} F_{v_y} &= 0 \end{aligned}$$

- In our case :

$$F = \alpha^2(E_x u + E_y v + E_t)^2 + u_x^2 + u_y^2 + v_x^2 + v_y^2.$$

- Euler equations give :

$$\begin{cases} \alpha^2(E_x u + E_y v + E_t)E_x = \nabla^2 u \\ \alpha^2(E_x u + E_y v + E_t)E_y = \nabla^2 v \end{cases}$$

Where : $\nabla^2 u = u_x^2 + u_y^2$ and $\nabla^2 v = v_x^2 + v_y^2$
$\nabla^2 u$ and $\nabla^2 v$ are the Laplacians of u and v.

- A Laplacian approximation that is frequently used in image processing :

$$\begin{aligned} \nabla^2 u &= u - \overline{u} \\ \nabla^2 v &= v - \overline{v} \end{aligned}$$

Where $\overline{u}$ and $\overline{v}$ are the mean values of u and v resp. in a spatial neighborhood. The previous equations then become :

$$\begin{cases} (1+\alpha^2 E_x^2)u + \alpha^2 E_x E_y v = \overline{u} - \alpha^2 E_x E_t \\ (1+\alpha^2 E_y^2)v + \alpha^2 E_x E_y u = \overline{v} - \alpha^2 E_y E_t \end{cases}$$

- **Iterative Solution :**

$$\begin{cases} u^{n+1} = \overline{u^n} - E_x \left[\frac{E_x \overline{u^n} + E_y \overline{v^n} + E_t}{\alpha^2 + E_x^2 + E_y^2} \right] \\ v^{n+1} = \overline{v^n} - E_y \left[\frac{E_x \overline{u^n} + E_y \overline{v^n} + E_t}{\alpha^2 + E_x^2 + E_y^2} \right] \end{cases}$$

- **Spatio-temporal derivative estimation**
  - Horn and Schunk propose to use the following approximation of Spatio-temporal derivative of image intensity :

$$
\begin{aligned}
4E_y \approx & E(x,y+1,t) - E(x,y,t) + E(x+1,y+1,t) - E(x+1,y,t) + E(x,y+ \\
& 1,t+1) - E(x,y,t+1) + E(x+1,y+1,t+1) - E(x+1,y,t+1) \\
4E_x \approx & E(x+1,y,t) - E(x,y,t) + E(x+1,y+1,t) - E(x,y+1,t) + \\
& E(x+1,y,t+1) - E(x,y,t+1) + E(x+1,y+1,t+1) - E(x,y+1,t+1) \\
4E_t \approx & E(x,y,t+1) - E(x,y,t) + E(x+1,y,t+1) - E(x+1,y,t) + \\
& E(x,y+1,t+1) - E(x,y+1,t) + E(x+1,y+1,t+1) - E(x+1,y+1,t)
\end{aligned}
$$

- **Velocity Laplacian Estimation :**

$$\begin{aligned}
\overline{u}(x,y,t) =& \tfrac{1}{6}\left[u(x-1,y,t)+u(x,y+1,t)+u(x+1,y,t)+u(x,y-1,t)\right] \\
&+\tfrac{1}{12}\left[u(x-1,y-1,t)+u(x-1,y+1,t)\right] \\
&+\tfrac{1}{12}\left[u(x+1,y+1,t)+u(x+1,y-1,t)\right] \\
\overline{v}(x,y,t) =& \tfrac{1}{6}\left[v(x-1,y,t)+v(x,y+1,t)+v(x+1,y,t)+v(x,y-1,t)\right] \\
&+\tfrac{1}{12}\left[v(x-1,y-1,t)+v(x-1,y+1,t)\right] \\
&+\tfrac{1}{12}\left[v(x+1,y+1,t)+v(x+1,y-1,t)\right]
\end{aligned}$$

- **Example of results**

Representation and validation of results

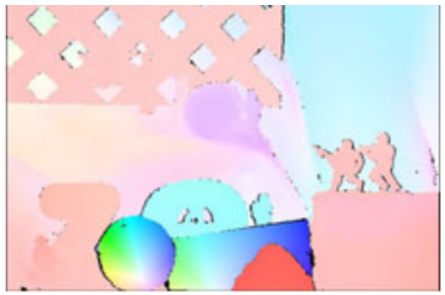
- Comparison with ground truth : ex. Middelebury.
- Colorful depiction of Middlebury. Hue : orientation, Saturation : norm.
- $\epsilon_{AE} = \arccos\left(\frac{\mathbf{w}^T\tilde{\mathbf{w}}}{\|\mathbf{w}\|\tilde{\mathbf{w}}\|}\right)$ with $\mathbf{w}$: GT flow and $\tilde{\mathbf{w}}$: estimated flow.
- $\epsilon_{EPE} = \|\mathbf{w} - \tilde{\mathbf{w}}\|$
- These are then the average of these errors which are calculated on the whole of the image.

Motion estimation

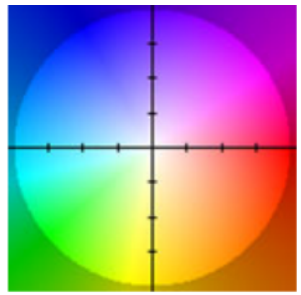
Horn and Schunk Algorithm

Army frame 0

Army frame 1



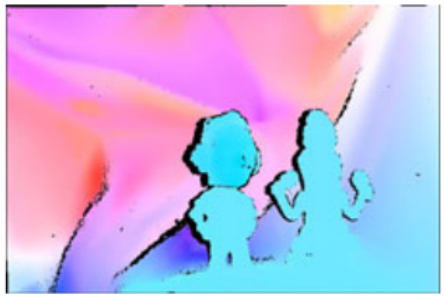
Army GT flow



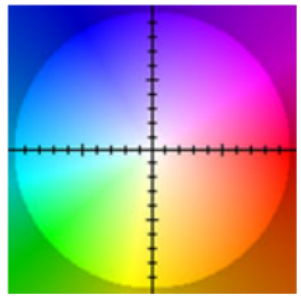
flow color coding

Mequon frame 0

Mequon frame 1



Mequon GT flow

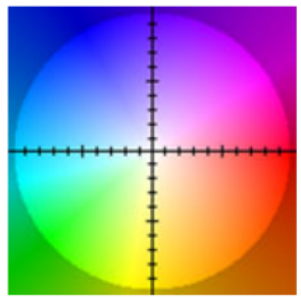


flow color coding

Schefflera frame 0

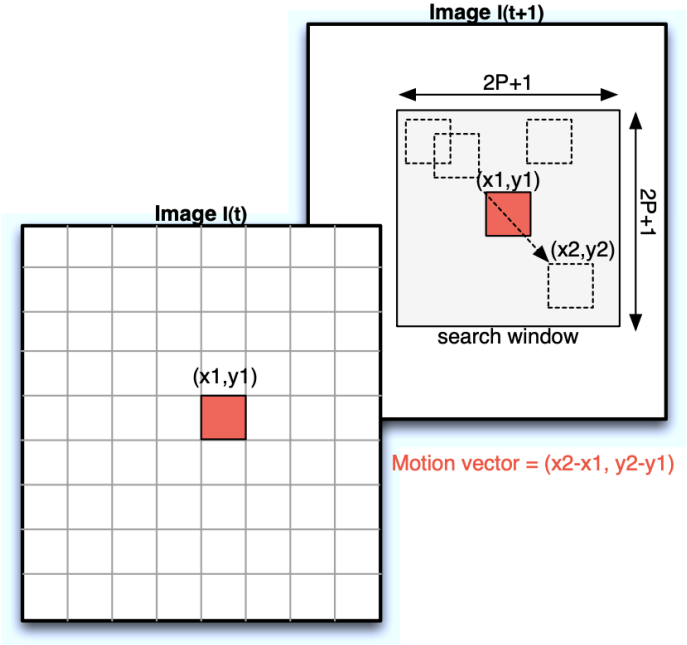
Schefflera frame 1

Schefflera GT flow



flow color coding

- **2 possibilities :**
  - Direct match : find similar block in the following image. The size of the search area is chosen according to of the maximum amplitude of the displacements. The block size is a compromise difficult to find ...
    * Similarity measures : Euclidean distances,lations, normalized correlations, etc.



  - Analytical resolution : minimize for example

- **Hypotheses :**
  - In a neighborhood of size 5×5, points are considered to have the same velocity vector $\Longrightarrow$ 25 equations.

$$\begin{bmatrix} E_x(\mathbf{p}_1) & E_y(\mathbf{p}_1) \\ E_x(\mathbf{p}_2) & E_y(\mathbf{p}_2) \\ . & . \\ E_x(\mathbf{p}_{25}) & E_y(\mathbf{p}_{25}) \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = - \begin{bmatrix} E_t(\mathbf{p}_1) \\ E_t(\mathbf{p}_2) \\ . \\ E_t(\mathbf{p}_{25}) \end{bmatrix}$$

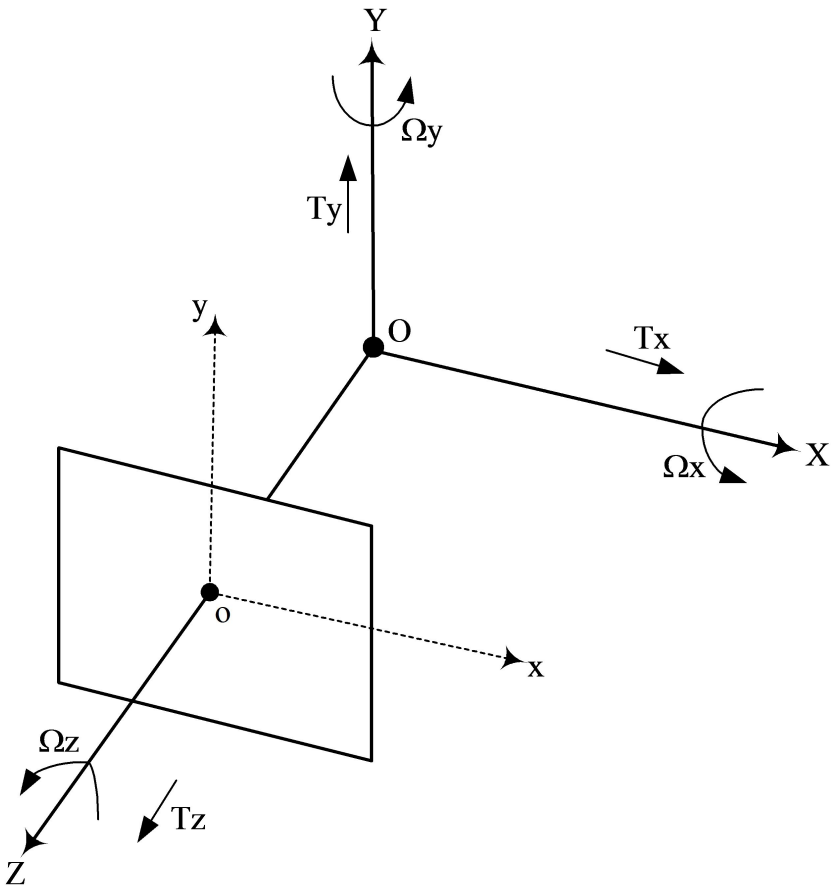
$$\mathbf{A}\mathbf{d} = \mathbf{b}$$

- Minimize $\|\mathbf{A}\mathbf{d} - \mathbf{b}\|^2 = 0$
- Least square solution : $\left(\mathbf{A}^T\mathbf{A}\right)\mathbf{d} = \mathbf{A}^T\mathbf{b}$

$$\begin{bmatrix} \sum E_x E_x & \sum E_x E_y \\ \sum E_x E_y & \sum E_y E_y \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = - \begin{bmatrix} \sum E_x E_t \\ \sum E_y E_t \end{bmatrix}$$

Relation between 3D motion and 2D motion

Basic Model and notations



The origin of the 3D frame is placed on the Optical center of the camera. The 2D image frame is on the middle of the image (optical center projection).

- (O, X, Y, Z) : camera frame with origin O (optical center).
- (o, x, y) : image frame with origin o (principal point). Axis $\overrightarrow{ox}$ and $\overrightarrow{oy}$ are parallel to $\overrightarrow{OX}$ and $\overrightarrow{OY}$ respectively.
- $\widehat{\mathbf{x}}$, $\widehat{\mathbf{y}}$ and $\widehat{\mathbf{z}}$ unitary vectors $\overrightarrow{OX}$, $\overrightarrow{OY}$ and $\overrightarrow{OZ}$ respectively.
- $P = (X, Y, Z)^T$: cartesian coordinates of a 3D point in the scene.
- $\mathbf{R} = (X, Y, Z)^T = \overrightarrow{OP}$
- $p = (x, y)^T$: 2D projection of point P on the image plane.
- $\mathbf{r} = (x, y)^T = \overrightarrow{op}$

- $\mathbf{V}_T = (V_{T_X}, V_{T_X}, V_{T_X})$: translational velocity of the camera.
- $\Omega = (\Omega_X, \Omega_Y, \Omega_Z)^T$ angular velocity of the camera.
- $\mathbf{T} = (T_X, T_Y, T_Z)^T$: translation vector according to $\overrightarrow{OX}$, $\overrightarrow{OY}$ and $\overrightarrow{OZ}$ respectively.
- θ_X, θ_Y, θ_Z : rotation angles around $\overrightarrow{OX}$, $\overrightarrow{OY}$ and $\overrightarrow{OZ}$ respectively.
- f : focal length (distance between the principal point and the optical center.

Relation between 3D motion and 2D motion

Recall : projection of a 3D point in the image plane

- **Perspective Projection :** The projection $p = (x, y)^t$ of a 3D point $P = (X, Y, Z)^t$ is given by :

$$p = \begin{bmatrix} x \\ y \end{bmatrix} = f \begin{bmatrix} \frac{X}{Z} \\ \frac{Y}{Z} \end{bmatrix}$$

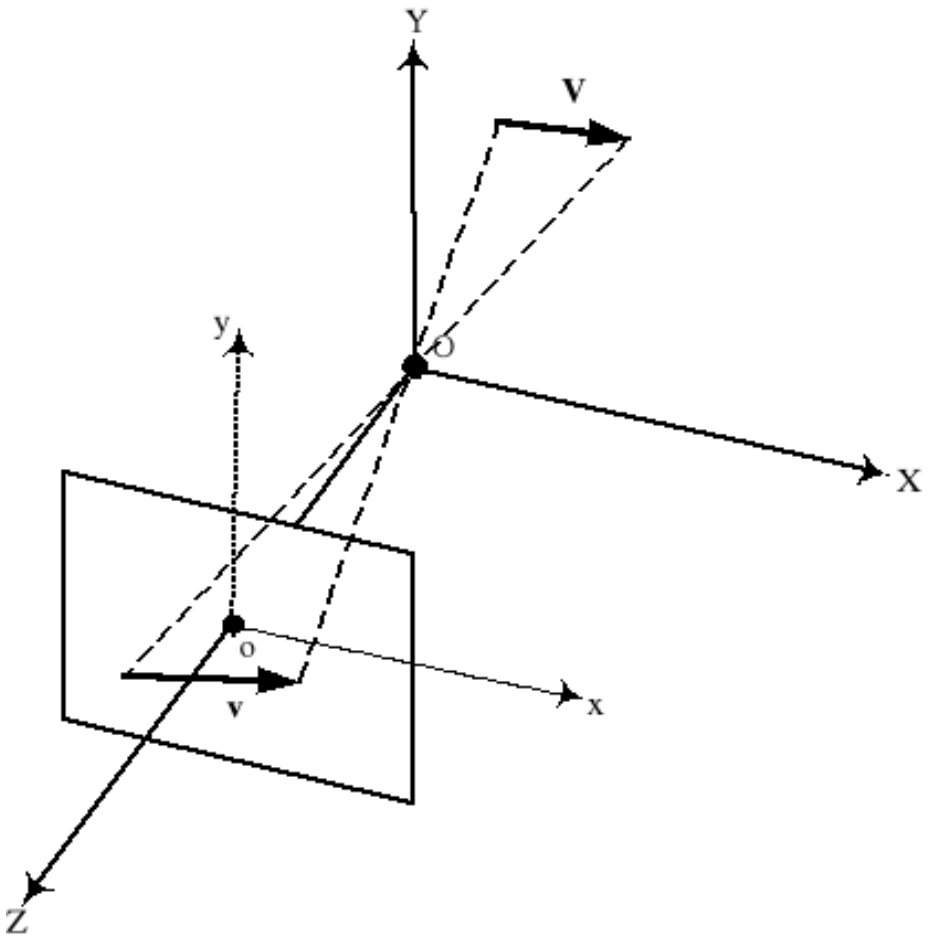
- Writing the expression in vectorial form is more readable :

$$\mathbf{r} = f \frac{\mathbf{R}}{\mathbf{R}.\widehat{\mathbf{z}}}$$

Where : $\mathbf{R}.\widehat{\mathbf{z}} = Z$

Relation between 3D motion and 2D motion

- **Relation between 3D and 2D velocity**



Relation between 3D motion and 2D motion

Let $\mathbf{V} = \frac{d\mathbf{R}}{dt}$ be the velocity of a point P and $\mathbf{v} = \frac{d\mathbf{r}}{dt}$ the velocity of its projection point p. When deriving the vectorial expression of perspective projection, we get :

$$\frac{d\mathbf{r}}{dt} = \frac{d(f\frac{\mathbf{R}}{\mathbf{R}.\widehat{\mathbf{z}}})}{dt} = f\frac{(\mathbf{R})'\mathbf{R}.\widehat{\mathbf{z}} - \mathbf{R}(\mathbf{R}.\widehat{\mathbf{z}})'}{(\mathbf{R}.\widehat{\mathbf{z}})^2} = f\frac{\mathbf{V}(\mathbf{R}.\widehat{\mathbf{z}}) - \mathbf{R}(\mathbf{V}.\widehat{\mathbf{z}})}{(\mathbf{R}.\widehat{\mathbf{z}})^2}$$

$$\mathbf{v} = f\frac{(\mathbf{R}\times\mathbf{V})\times\widehat{\mathbf{z}}}{(\mathbf{R}.\widehat{\mathbf{z}})^2}$$

2D motion field depends on depth $Z = R.\widehat{\mathbf{z}}$.

Relation between 3D motion and 2D motion

When an observer moves with an instantaneous translation velocity $\mathbf{V}_T$ and an instantaneous angular velocityet Ω, relatively to a rigid environment then all scene points seem (for the observer) to move with a velocity given by :

$$\mathbf{V} = \frac{d\mathbf{R}}{dt} = \begin{pmatrix} \dot{X} \\ \dot{Y} \\ \dot{Z} \end{pmatrix} = -\mathbf{V}_T - \Omega \times \mathbf{R}$$

$$\Longleftrightarrow \begin{cases} \dot{X} = -V_{T_X} - \Omega_Y Z + \Omega_Z Y \\ \dot{Y} = -V_{T_Y} - \Omega_Z X + \Omega_X Z \\ \dot{Z} = -V_{T_Z} - \Omega_X Y + \Omega_Y X \end{cases}$$

Each point velocity $p = (x, y)^T$ is defined by $\mathbf{v} = (\dot{x}, \dot{y})^T$.

Relation between 3D motion and 2D motion

- **Case of orthogonal projection**
  - In this case, $X = x$ and $Y = y$. Consequently, the velocity of a point is directly deduced by derivation :

$$\begin{cases} u = \dot{x} = -V_{T_X} - \Omega_Y Z + \Omega_Z y \\ v = \dot{y} = -V_{T_Y} - \Omega_Z x + \Omega_X Z \end{cases}$$

Relation between 3D motion and 2D motion

- **Case of perspective projection**

By deriving the expression of the perspective projection of a point :

$$\dot{x} = \frac{\dot{X}Z - \dot{Z}X}{Z^2} \text{ et } \dot{y} = \frac{\dot{Y}Z - \dot{Z}Y}{Z^2}$$

Replacing by the expressions of velocities, we get :

$$\begin{cases} u = \dot{x} = \left(-f\frac{V_{T_X}}{Z} - \Omega_Y + y\Omega_Z\right) - x\left(-\frac{V_{T_Z}}{Z} - y\frac{\Omega_X}{f} + x\frac{\Omega_Y}{f}\right) \\ v = \dot{y} = \left(-f\frac{V_{T_Y}}{Z} - x\Omega_Z + \Omega_X\right) - y\left(-\frac{V_{T_Z}}{Z} - y\frac{\Omega_X}{f} + x\frac{\Omega_Y}{f}\right) \end{cases}$$

Also, by rearranging terms :

$$\begin{cases} u = \frac{xy}{f}\Omega_X - \left(\frac{x^2}{f} + 1\right)\Omega_Y + y\Omega_Z - \frac{fV_{T_X} + xV_{T_Z}}{Z} \\ v = -\frac{xy}{f}\Omega_Y - \left(\frac{y^2}{f} + 1\right)\Omega_X - x\Omega_Z - \frac{fV_{T_Y} + yV_{T_Z}}{Z} \end{cases}$$

- **Pure translational motion**
  In this case : $\Omega_Z = 0$, $\Omega_Y = 0$, $\Omega_Z = 0$:

$$\begin{cases} u = -\frac{fV_{T_X} + xV_{T_Z}}{Z} \\ v = -\frac{fV_{T_Y} + yV_{T_Z}}{Z} \end{cases}$$

- **Longitudinal Motion**
  In this case, $V_{T_X} = 0$, $V_{T_Y} = 0$:

$$\begin{cases} u = -\frac{xT_Z}{Z} \\ v = -\frac{yT_Z}{Z} \end{cases}$$

A point of infinity depth has a null velocity. In deed, when $Z = \infty$, we have $u = 0$ and $v = 0$. Hence, for each point :

$$\frac{v}{u} = \frac{-fV_{T_Y} + yV_{T_Z}}{-fV_{T_X} + xV_{T_Z}}$$

By setting u and v to zero in the equation of pure translational motion, we get :

$$V_{T_X} = \frac{x_{foe}V_{T_Z}}{f} \quad V_{T_Y} = \frac{y_{foe}V_{T_Z}}{f}.$$

Where (x_{foe}, y_{foe}) are projections in the image of the Focus of Expansion or the focus of Contraction (resp. FOE and FOE) with :

$$x_{foe} = f\frac{V_{T_X}}{V_{T_Z}} \quad y_{foe} = f\frac{V_{T_Y}}{V_{T_Z}}.$$

Finally, if we group the two above expressions, we obtain :

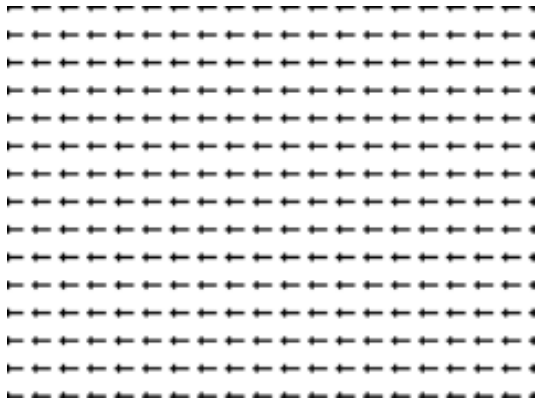
$$\frac{v}{u} = \frac{y - y_{foe}}{x - x_{foe}}$$

This relation proves that all lines from x to $x\prime$ where $u = x\prime - x$ et $v = y\prime - y$, passes through (x_{foe}, y_{foe}).

Relation between 3D motion and 2D motion

Exemple de mouvements simples

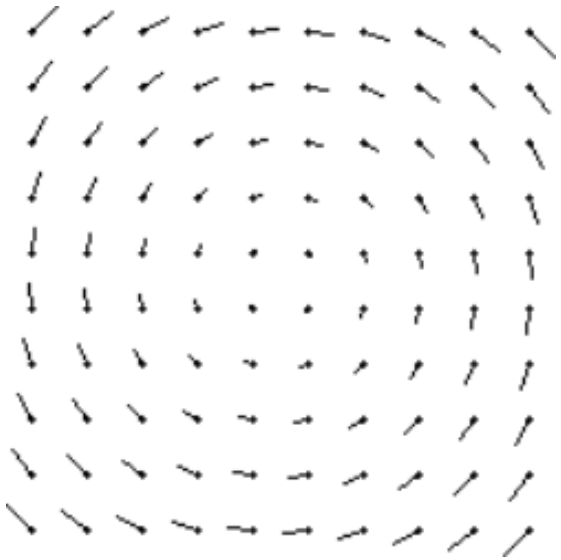
- Translation // to a surface with a constant distance from the camera



Relation between 3D motion and 2D motion

Exemple de mouvements simples

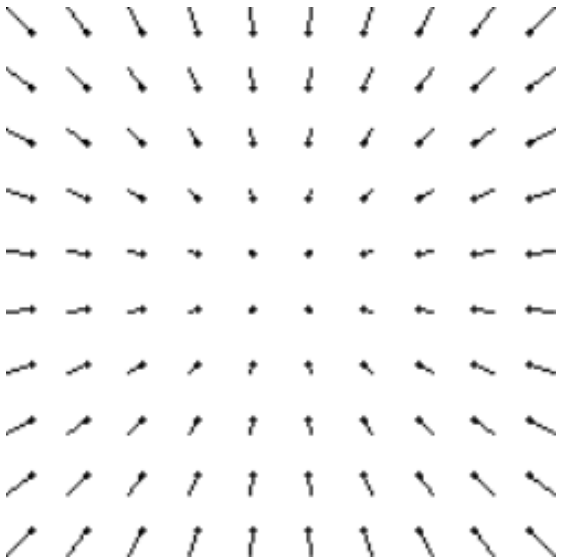
- Rotation toward an axis perpendicular to the projection plane of the camera



Relation between 3D motion and 2D motion

Example of simple movements

- Translation perpendicularly to a surface



Relation between 3D motion and 2D motion

Example of simple movements