VIC-Introduction to Visual Computing
Course 7 : Motion

Maria Vakalopoulou and Céline Hudelot

3 février 2025

Plan

A World in Motion

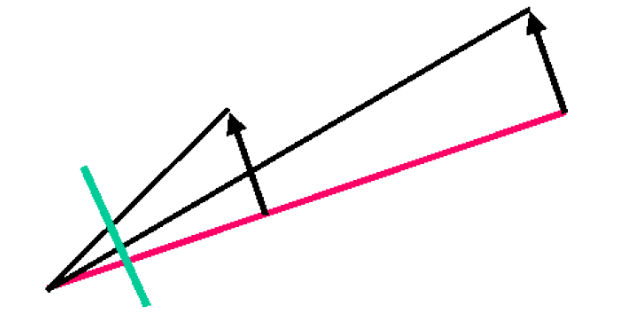
Perception, understanding, and prediction of motion are daily actions for a human.

See https://drive.google.com/file/d/1Ew7KUBKud0uL1knt4i9kAki_Bn6nD3q_/view?usp=sharing
slide credit : Ce Liu

Motion Perception

Perception of Speed

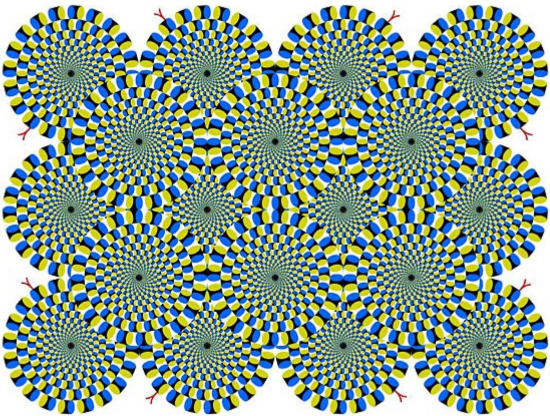
When objects move at the same speed, those farther away seem to move more slowly.



https://isle.hanover.edu/Ch07DepthSize/Ch07MotionParallaxExpl.html

Motion Perception

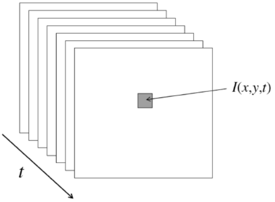
Human vision can play tricks !



http://www.ritsumei.ac.jp/~akitaoka/index-e.html

From Images to Video

- A video is a sequence of images captured over time.
- Image data : a function of space (x, y) and time (t).



Objective : retrieve motion information in the scene from variations in intensity $I(x, y, t)$ over time.

Importance of Motion

Why Extract Motion ?

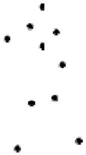
- It is sometimes easier to extract information from an image sequence than from static images.
  - Example : camouflaged objects.
  - The relative size and position of objects can be determined more easily when objects are in motion.

See : https://drive.google.com/file/d/1D5XIZ_Dis1m3953r95lJ_kWU_Fv2FgIo/view?usp=sharing

Importance of Motion

Why Extract Motion ?

- Motion is sometimes the only feature available for recognition.
- Very sparse motion information can evoke a strong percept.



see https://drive.google.com/file/d/1Q8O3oU6vbZRx8pcY3k3eK3gZnlFfsxRq/view?usp=sharing

Importance of Motion in Vision

Numerous Applications

- 3D Reconstruction from Motion : Structure from Motion
- Object Segmentation based on Motion Features
- Learning Dynamic Models : Object Tracking
- Event and Activity Recognition
- Video Quality Enhancement : Motion Stabilization
- ...

Causes of Motion

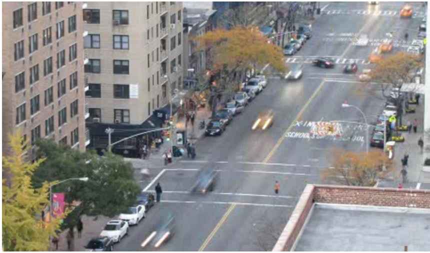
Three Factors

- Light
- Objects
- Camera

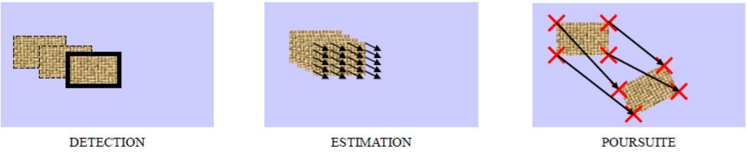
Their Variation Can Generate Motion

- Static Camera, Moving Objects (e.g., surveillance).
- Moving Camera, Static Scene (e.g., 3D capture).
- Moving Camera, Moving Scene (e.g., sports, cinema).
- Static Camera, Moving Objects, Illumination Changes (e.g., outdoor surveillance).

Motion : Some Examples



Main Problems



- Detection : Identify in each image the pixels belonging to moving objects.
- Estimation : Compute the apparent motion of each pixel.
- Tracking : Match certain spatial structures for each pair of images.

Other Problems

- Recognition : Recognizing the scenario corresponding to the movement, event, and activity recognition.
- Structure from Motion Estimation.
- View Interpolation : Synthesis of new images in the sequence.

Outline

Motion Detection : Introduction

Objective

Detect moving objects using a camera.

Objective

Detect only image changes that correspond to motion.

Assumptions

- Fixed camera.
- Stable illumination conditions.
- No prior knowledge about the nature and dynamics of the object.

Image Difference

Difference Between Images

For each pixel s, threshold the difference between images :

$$|I_2(s) - I_1(s)| > \textit{threshold}$$

Average Difference Between Images

For each pixel s, consider a window $W(s)$ of size $n \times n$ around s :

$$\frac{\sum_{r \in W(s)} |I_2(r) - I_1(r)|}{n \times n} > \text{threshold}$$

- A simple idea that can prove effective.

Image Difference

Types of Changes

- Appearance of the background previously hidden by the object (at $t-1$).
- Occlusion of the background by the object at its current position (t).
- The sign of the difference can help differentiate these two areas.

Image Difference

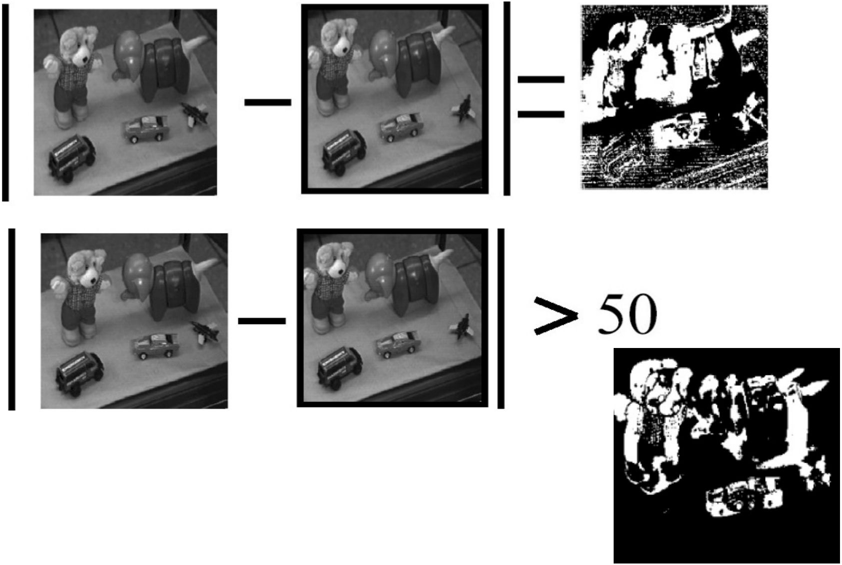
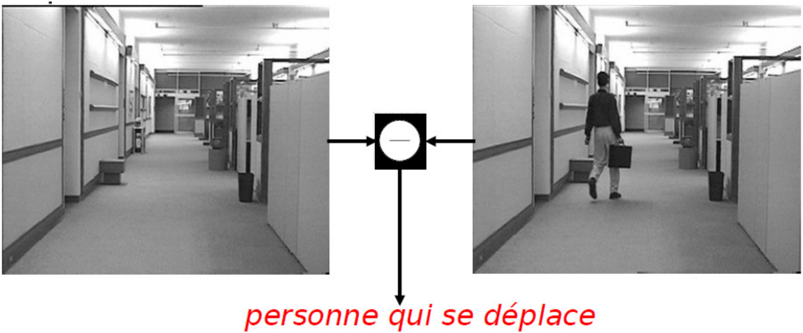


Image Difference



A very fast method for simple motion detection.

Image Difference : Applications

- Person detection.
- Human-computer interaction : video games - webcam.
- Counting cars on a road.

Requirements

- Use of a uniformly colored background.
- A static but arbitrary background.

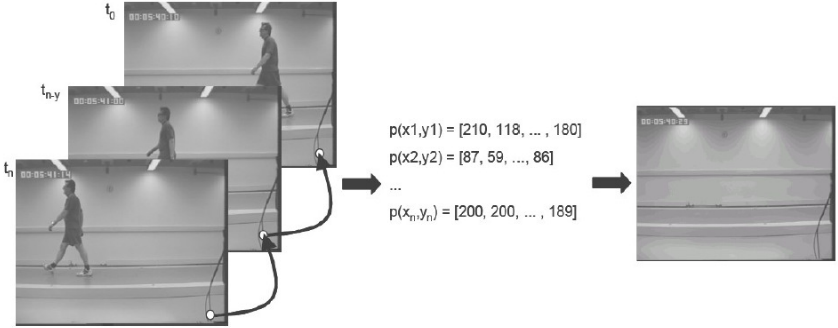
Background Subtraction

Principle

- Assume that we have an image of the background (BG).
- Subtract the current image from this background image to isolate the moving object.
  - The background image does not vary over time.
- The background image can be updated :
  - To account for lighting conditions.
  - Temporal variations in the scene (moving objects, etc.).

Background Subtraction

Background Construction



Séquence d'images — *Tableau des valeurs pour chaque pixel* — *Valeur médiane de chaque pixel*

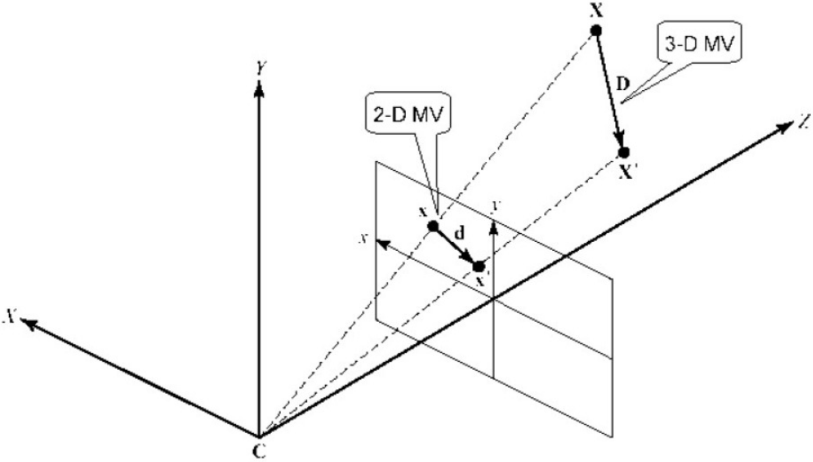
Background Subtraction

Image — *Image du fond* — *Détection*

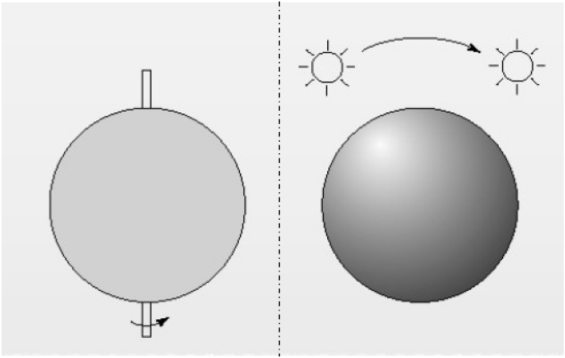
Post-processing is required after detection to obtain the object of interest.

Outline

From 3D Motion to 2D Motion



Real Motion - Apparent Motion



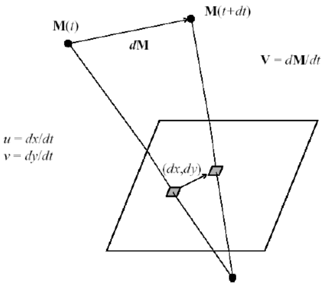
- Case 1 : the sphere rotates on itself.
  - Non-zero real motion.
  - Zero apparent motion.
- Case 2 : the light source moves.
  - Zero real motion.
  - Non-zero apparent motion.

Different Types of Motion

- Rigid scene : only the camera moves (*ego motion*).
  - Example : onboard camera.
- Multiple rigid objects with different motions.
  - Example : traffic monitoring at an intersection.
- One or more non-rigid objects.
  - Example : living cells, human motion, deformations.

Motion Field

When an object moves in front of a visual sensor, there is a change in the image.



- A point M in the scene at time t moves to $M(t+dt)$ after a small interval dt.
- The corresponding velocity vector is $V = dM/dt$.
- $M(t)$ and $M(t+dt)$ project to points $m(t)$ and $m(t+dt)$ in the image sequence.
- Let $x(t)$ and $y(t)$ be the coordinates of m, the apparent velocity in the image has the following components :
  - $u = dx/dt$ and $v = dy/dt$
- The velocity field is the set of values $u(x,y)$ and $v(x,y)$ in the image.
- Recovering motion = estimating u and v for each pixel in the image.

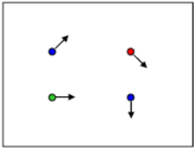
Optical Flow

Definition

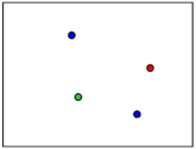
Optical flow is the apparent motion of brightness patterns in the image. It describes the direction and speed of motion of image features.

- Note : apparent motion can be caused by illumination changes and may not correspond to real motion.
- Objective : Associate to each pixel $s = (x, y)$ a motion vector $V(s) = (u, v)$ representing its instantaneous apparent velocity.
- **Study of variations in the intensity function $I(x, y, t)$.**

Optical Flow



$H(x, y)$



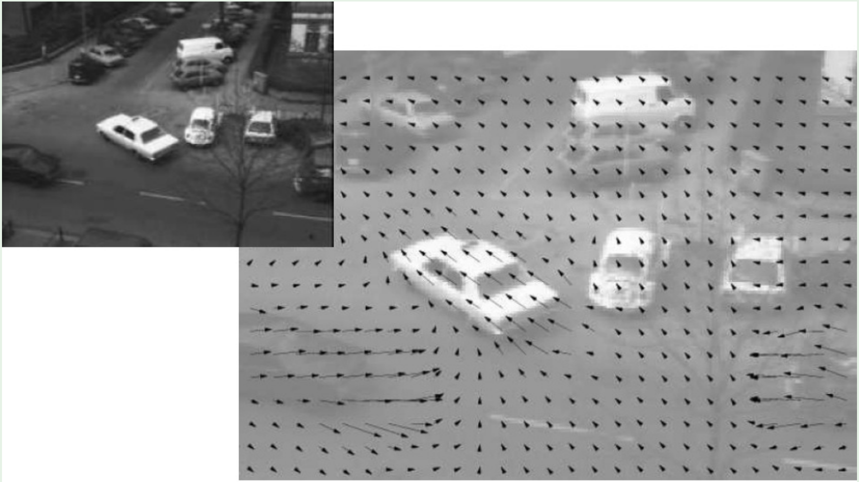
$I(x, y)$

How to estimate the motion of a pixel from image H to image I ?

- Pixel correspondence problem : which pixel in I corresponds to pixel $H(x, y)$? (yes, it's a correspondence problem again)
- Result in the form of a relative displacement :
  - Displacement : $I(x, y) = H(x + dx, y + dy)$
  - Velocity : $v(x, y) = (dx, dy)^T$

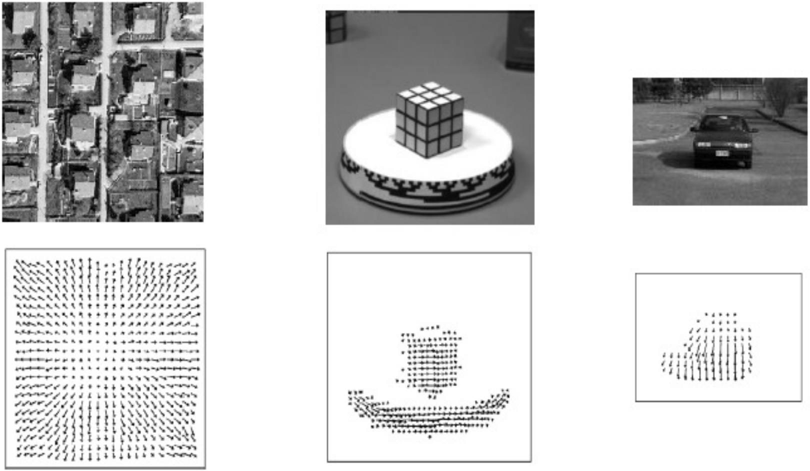
Optical Flow

Example of Optical Flow



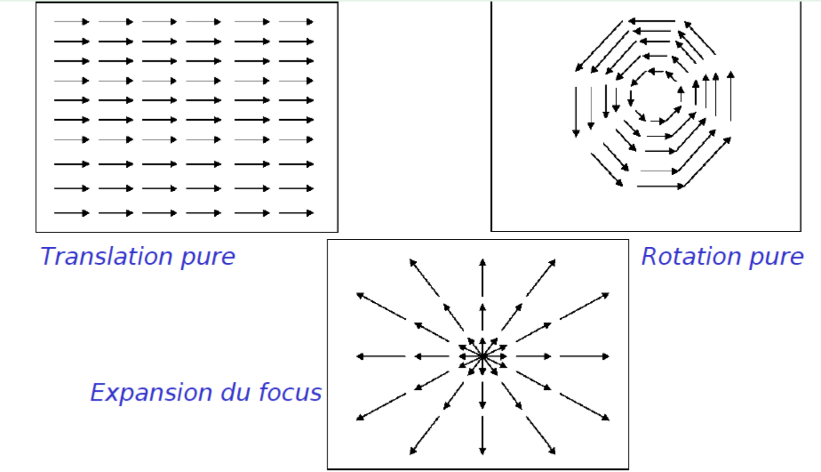
Optical Flow

Example of Optical Flow



Optical Flow

Example of Motion Fields



Motion Estimation

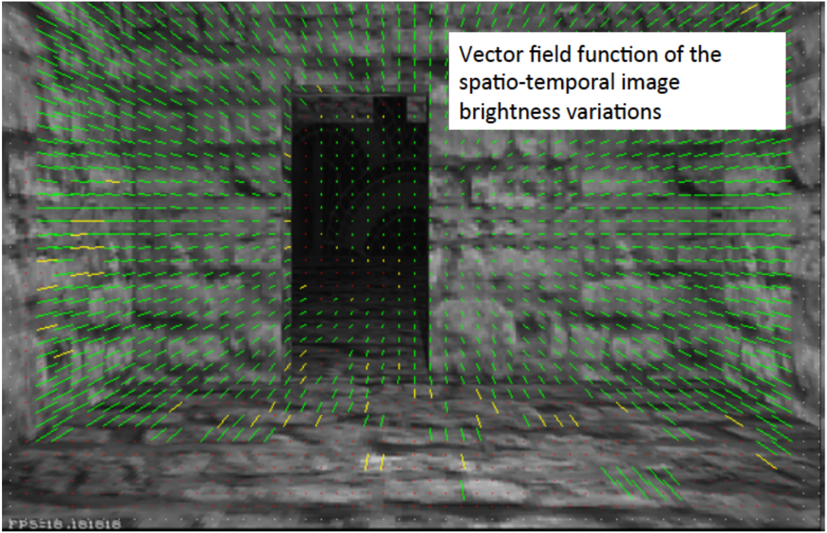
Two Approaches

- Optical Flow : Find the motion of each pixel from the spatio-temporal variations of the intensity function.
- Feature Tracking : Extract visual features (corners, textured areas, interest points, etc.) and track them over multiple frames.

Two problems, one method : Lucas-Kanade.

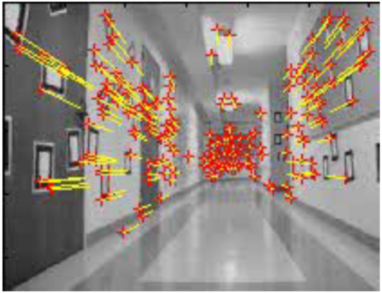
B. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pp 674-679. 1981

Optical Flow

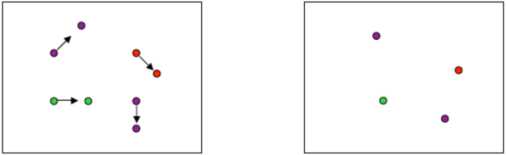


Vector field function of the spatio-temporal image brightness variations

Feature Tracking



Motion Estimation : Assumptions



$I(x,y,t-1)$ $I(x,y,t)$

Key Assumptions

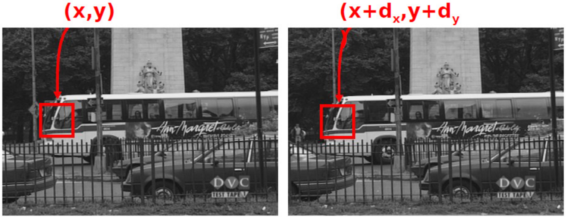
- Brightness (or color) constancy : the projection of the same point has the same appearance in different images.
- Spatial coherence : points move similarly to their neighbors.
- Small motion : points do not move too far.

Motion Estimation : Assumptions

Assumption 1 : Brightness Constancy

The observed intensity of an object does not change as it moves.

- If intensity changes : there is motion.
- Not always true in practice.
- Small motions : ideally less than 1 pixel per frame.



Sous l'hypothèse d'intensité constante :

$$I(x,y,t) = I(x+dx,y+dy,t+1)$$

Motion Estimation : Assumptions

Assumption 2 : Spatial Coherence

Points move similarly to their neighbors.

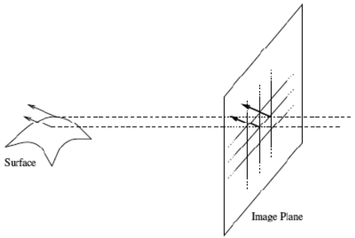


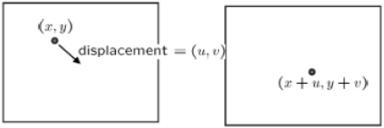
Figure 1.7: Spatial coherence assumption. Neighboring points in the image are assumed to belong to the same surface in the scene.

Optical Flow Estimation

Different Approaches

- Differential approach :
  - Use the spatio-temporal derivatives of the image as a source of information about the motion.
- Correlation-based approach :
  - Search for a corresponding point through correlation between images.
- Frequency-based approach :
  - Use changes in frequency and phase to determine the motion.

Optical Flow Estimation



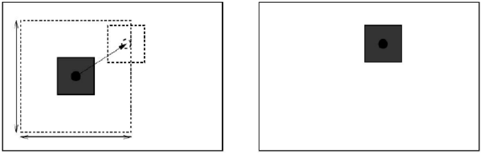
$+$ contextual information

Different Approaches

- Correlation-based approach : $\forall s, I_2(s + \mathbf{v}(s)) = I_1(s)$
- Differential approach : $\frac{dI(x,y,t)}{dt} = 0$

Optical Flow Measurement : Correlation-based Approach

- To measure optical flow, it is necessary to find the corresponding points between two frames.
- Idea : exploit similarities between patches of the image surrounding individual points.
  - $W(s)$ is the window centered at s.
  - Goal : find $\mathbf{v}(s)$ that maximizes the similarity between I_1 in $W(s)$ and I_2 in $W(s+\mathbf{v}(s))$



Optical Flow Measurement : Correlation-based Approach

Similarity Measurements

- SAD : Sum of absolute differences

$$\sum_{r \in W(s)} |I_2(r + \mathbf{v}) - I_1(r)|$$

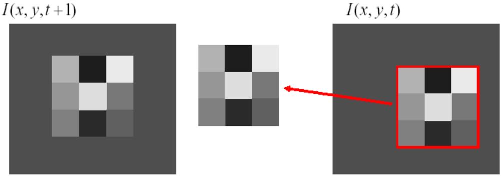
- SSD : Sum of squared differences

$$\sum_{r \in W(s)} (I_2(r + \mathbf{v}) - I_1(r))^2$$

- Cross-correlation

$$\sum_{r \in W(s)} (I_2(r + \mathbf{v}).I_1(r))$$

Optical Flow Measurement : Correlation-based Approach



Minimizing the intensity difference

$$E_{SSD}(u, v) = \sum_{x,y \in W} (I(x+u, y+v, t+1) - I(x, y, t))^2$$

Optical Flow Estimation : Differential Approach

$$\frac{dI(x,y,t)}{dt} = 0$$

$$\frac{dI(x,y,t)}{dt} = \frac{\partial I}{\partial x}\frac{dx}{dt} + \frac{\partial I}{\partial y}\frac{dy}{dt} + \frac{\partial I}{\partial t} = 0$$

- $\frac{\partial I}{\partial x}$ and $\frac{\partial I}{\partial y}$: Spatial gradients of the image, i.e., how the image changes in x and y for a fixed time.
- $\frac{\partial I}{\partial t}$: Temporal derivative of the image, i.e., how the image changes over time for a fixed position.
- $\frac{dx}{dt} = u$, $\frac{dy}{dt} = v$, temporal derivatives, i.e., velocity components representing the rate of change in x and y.

Optical Flow Estimation : Differential Approach

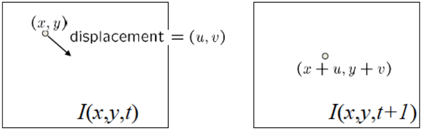
$$\frac{dI(x,y,t)}{dt} = 0$$
$$\frac{dI(x,y,t)}{dt} = \frac{\partial I}{\partial x}\frac{dx}{dt} + \frac{\partial I}{\partial y}\frac{dy}{dt} + \frac{\partial I}{\partial t} = 0$$

Rewriting as :

$\frac{\partial I}{\partial t} + \nabla I^T.\mathbf{v} = 0$ with $\mathbf{v} = (u, v)$ the unknown velocity.

Optical Flow Estimation : Differential Approach

Another way to obtain this equation.



Constant intensity ($\Delta t = 1$) :

$$I(x, y, t) = I(x + u(x, y), y + v(x, y), t + 1)$$

Small movement : Taylor expansion of $I(x, y, t)$:

$$I(x + u, y + v, t + 1) \approx I(x, y, t) + I_x u(x, y) + I_y v(x, y) + I_t$$

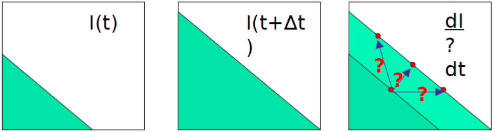
$$I(x + u, y + v, t + 1) - I(x, t, t) = I_x.u + I_y.v + I_t$$

Thus : $I_x.u + I_y.v + I_t \approx 0 \Rightarrow \nabla I.[u\ v]^T + I_t = 0$

Optical Flow Estimation : Differential Approach

$$\frac{\partial I}{\partial t} + \nabla I^T.\mathbf{v} = 0 \text{ with } \mathbf{v} = (u, v) \text{ the unknown velocity.}$$

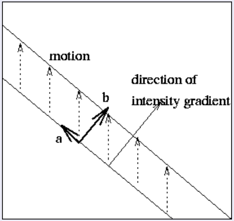
- One equation, two unknowns, so multiple solutions are possible.
- Multiple motions can correspond to the situation below (aperture problem).



Idea : reduce to the simplest motion (minimal solution) by adding constraints to the problem.

Optical Flow Estimation : Differential Approach

Aperture Problem

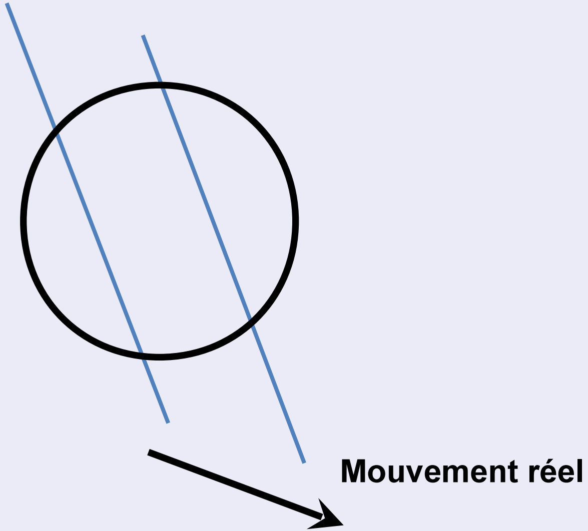


Consider a point in the image and compute the gradient in a small window around that point (the aperture)

- In this window, intensity varies in the direction of the gradient but not in the perpendicular direction.
- From the contour's perspective, intensity varies across the contour but not along the contour.
- A motion parallel to the contour cannot be recovered.
- Only *b*, the normal component to the edge, can be estimated.

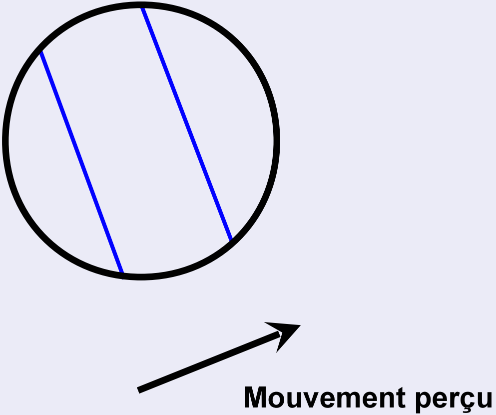
Optical Flow Estimation : Differential Approach

Aperture Problem



Optical Flow Estimation : Differential Approach

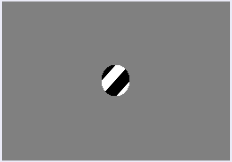
Aperture Problem



http://elvers.us/perception/aperture/

Optical Flow Estimation : Differential Approach

The barber pole illusion


http://en.wikipedia.org/wiki/Barberpole_illusion

Optical Flow Estimation : Differential Approach

Adding Constraints

- Hypothesis 1 : The motion field of a small part of the image is constant : Lucas & Kanade ; constant field in a neighborhood
  - http://en.wikipedia.org/wiki/Lucas-Kanade_Optical_Flow_Method
  - B. D. Lucas and T. Kanade (1981), An iterative image registration technique with an application to stereo vision. Proceedings of Imaging Understanding Workshop, pages 121–130 http://cseweb.ucsd.edu/classes/sp02/cse252/lucaskanade81.pdf
- Hypothesis 2 : The motion field must be smooth : Horn & Schunk
  - http://en.wikipedia.org/wiki/Horn-Schunck_method
  - B.K.P. Horn and B.G. Schunck, Determining optical flow. Artificial Intelligence, vol 17, pp 185-203, 1981. Manuscript. http://dspace.mit.edu/handle/1721.1/6337

Optical Flow Estimation : Differential Approach

Lucas & Kanade Approach (Spatial Coherence Constraint)

The motion field of a small part of the image is constant.

- A neighborhood of $N \times N$ pixels is considered (typically 5), and it is assumed that the pixels have the same (u, v)
- With a 5×5 window, we get 25 equations per pixel.

$$0 = \frac{\partial I}{\partial t}(p_i) + \nabla I(p_i).\,[u\ v]$$

$$\begin{bmatrix} I_x(p_1) & I_y(p_1) \\ I_x(p_2) & I_y(p_2) \\ . & . \\ . & . \\ . & . \\ I_x(p_{25}) & I_y(p_{25}) \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = - \begin{bmatrix} I_t(p_1) \\ I_t(p_2) \\ . \\ . \\ . \\ I_t(p_{25}) \end{bmatrix}$$

Optical Flow Estimation : Differential Approach

Lucas & Kanade Approach

Solution by minimization (least squares)

$$\begin{bmatrix} I_x(p_1) & I_y(p_1) \\ I_x(p_2) & I_y(p_2) \\ . & . \\ . & . \\ . & . \\ I_x(p_{25}) & I_y(p_{25}) \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = - \begin{bmatrix} I_t(p_1) \\ I_t(p_2) \\ . \\ . \\ . \\ I_t(p_{25}) \end{bmatrix} \Leftrightarrow A\mathbf{v} = b$$

- Solution to $A^T A\mathbf{v} = A^T b$
- $\mathbf{v} = (A^T A)^{-1} A^T b.$

Optical Flow Estimation : Differential Approach

Lucas & Kanade Approach

$$A^T A\mathbf{v} = A^T b$$

$$\begin{bmatrix} \sum I_x I_x & \sum I_x I_y \\ \sum I_x I_y & \sum I_y I_y \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = - \begin{bmatrix} \sum I_x I_t \\ \sum I_y I_t \end{bmatrix}$$

- Sums over all pixels in the $N \times N$ window
- Problem solvable when (which points are the right ones to track ?) :
  - $A^T A$ is invertible.
  - $A^T A$ should not be too small due to noise :
    - The eigenvalues λ_1 and λ_2 should not be too small.
  - $A^T A$ should be well-conditioned (λ_1/λ_2 should not be too large)
  - $A^T A$: Does this remind you of anything ?

Optical Flow Estimation : Differential Approach

Lucas & Kanade Approach

A^TA is equivalent to the matrix M of the Harris detector.

$$A^TA = \begin{bmatrix} \sum I_x I_x & \sum I_x I_y \\ \sum I_x I_y & \sum I_y I_y \end{bmatrix} = \sum \begin{bmatrix} I_x \\ I_y \end{bmatrix} [I_x \; I_y] = \sum \nabla I (\nabla I)^T$$

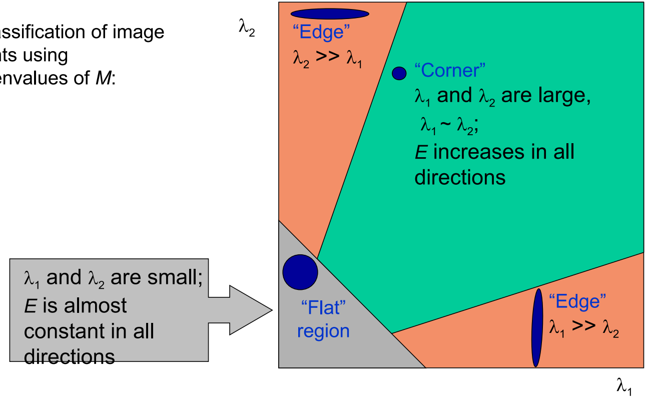
- The eigenvalues and eigenvectors of A^TA are related to the direction and magnitude of the gradient.
  - The eigenvector associated with the largest eigenvalue : direction of a sharp intensity change.
  - The other eigenvector is orthogonal.

Indication on the good features to track.
We can solve A^TA when there are no issues with the aperture problem.

Optical Flow Estimation : Differential Approach

Eigenvalue Interpretation.

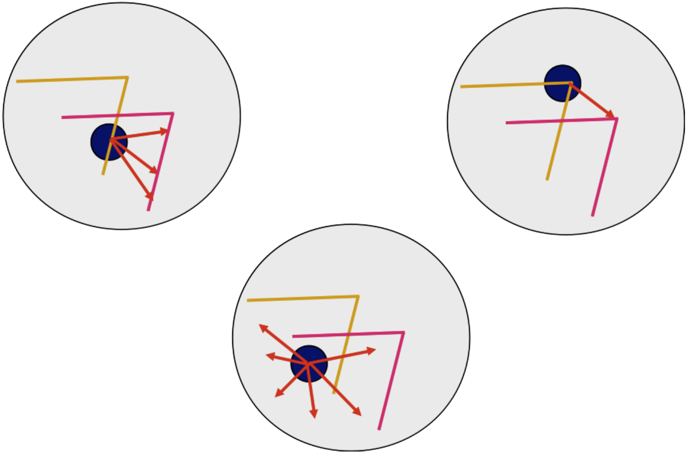
Classification of image points using eigenvalues of *M*:



Source : Daria Frolova, Invariant feature detectors and descriptors.
`http://www.wisdom.weizmann.ac.il/~daryaf/InvariantFeatures.ppt`

Optical Flow Estimation : Differential Approach

Feature Analysis.



Optical Flow Estimation : Differential Approach

Contour

http://persci.mit.edu/demos/jwang/garden-layer/movies/mov/flower-orig2.mov

$$\sum \nabla I (\nabla I)^T$$

- Gradients with very large or very small values
- λ_1 large and λ_2 small.

Optical Flow Estimation : Differential Approach

Low-Textured Region

$$\sum \nabla I (\nabla I)^T$$

- Gradients with small magnitude
- λ_1 small and λ_2 small.

Optical Flow Estimation : Differential Approach

Highly Textured Region

$$\sum \nabla I (\nabla I)^T$$

- Gradients with different large magnitudes
- λ_1 large and λ_2 large.

Motion Estimation : Reminder of Assumptions

Hypothesis 1 : Constant Intensity

The observed intensity of an object does not change when it moves.

Hypothesis 2 : Spatial Coherence

Points move in the same way as their neighbors.

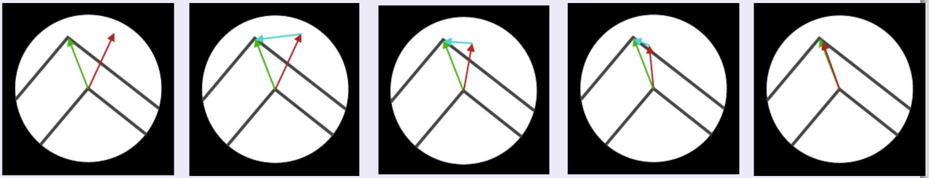
Hypothesis 3 : Small Movements

Ideally less than 1 pixel per frame.

Optical Flow Estimation : Differential Approach

Lucas & Kanade Approach : Limitations

The basic Lucas and Kanade approach is valid only if the displacement is small (first-order approximations) : an iterative version of the algorithm allows refining the motion estimation.



Optical Flow Estimation : Differential Approach

Iterative Lucas Kanade Algorithm

1. Initialize $(x', y') = (x, y)$.
2. Compute (u, v) using the basic method :

$$\begin{bmatrix} \sum I_x I_x & \sum I_x I_y \\ \sum I_x I_y & \sum I_y I_y \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = - \begin{bmatrix} \sum I_x I_t \\ \sum I_y I_t \end{bmatrix}$$

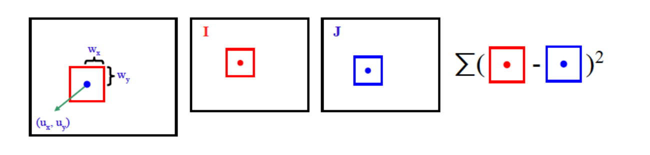
   where $I_t = I(x', y', t+1) - I(x, y, t)$.
3. Move the window by (u, v) :

$$x' \leftarrow x' + u; \; y' \leftarrow y' + v$$

   .
4. Recalculate I_t.
5. Repeat steps 2 to 4 until very small changes are obtained.

Iterative Lucas and Kanade Algorithm

General Formulation



We are looking for d $= (d_x, d_y)$ to minimize the residual error :

$$\epsilon(\mathrm{d}) = \epsilon(d_x, d_y) = \sum_{x=u_x-w_x}^{u_x+w_x} \sum_{y=u_y-w_y}^{u_y+w_y} (I(x,y) - J(x+d_x, y+d_y))^2$$

w_x and w_y define the window size $((2w_x+1)\times(2w_y+1))$.
Let $\bar{\nu} = [\nu_x \; \nu_y]^T =$ d be the sought displacement. The optimum occurs when :

$$\frac{\partial\epsilon(\bar{\nu})}{\partial\bar{\nu}}|_{\bar{\nu}=\bar{\nu}_{opt}} = [0 \; 0]$$

Iterative Lucas and Kanade Algorithm

General Formulation

$$\frac{\partial \epsilon(\bar{\nu})}{\partial \bar{\nu}} = -2 \sum \sum (I(x,y) - J(x+\nu_x, y+\nu_y)). \begin{bmatrix} \frac{\partial J}{\partial x} & \frac{\partial J}{\partial y} \end{bmatrix}$$

Substitute $J(x+\nu_x, y+\nu_y)$ with its first-order Taylor expansion at $\bar{\nu} = [0\ 0]^T$.

$$\frac{\partial \epsilon(\bar{\nu})}{\partial \bar{\nu}} \approx -2 \sum \sum (I(x,y) - J(x,y) - \begin{bmatrix} \frac{\partial J}{\partial x} & \frac{\partial J}{\partial y} \end{bmatrix} \bar{\nu}). \begin{bmatrix} \frac{\partial J}{\partial x} & \frac{\partial J}{\partial y} \end{bmatrix}$$

$\begin{bmatrix} \frac{\partial J}{\partial x} & \frac{\partial J}{\partial y} \end{bmatrix} =$ gradient of the image $\Rightarrow \nabla I^T$

$I(x,y) - J(x,y) =$ frame difference which can be interpreted as the temporal derivative of the image at $[x\ y]^T$ (within the considered window) : δI.

$$\frac{1}{2} \frac{\partial \epsilon(\bar{\nu})}{\partial \bar{\nu}} \approx \sum \sum (\nabla I^T \bar{\nu} - \delta I) \nabla I^T$$

$$\frac{1}{2} \left[\frac{\partial \epsilon(\bar{\nu})}{\partial \bar{\nu}} \right]^T \approx \sum \sum \left(\begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} \bar{\nu} - \begin{bmatrix} \delta I\ I_x \\ \delta I\ I_y \end{bmatrix} \right)$$

Iterative Lucas and Kanade Algorithm

General Formulation

With $G = \sum\sum \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}$ and $b = \sum\sum \begin{bmatrix} \delta I \; I_x \\ \delta I \; I_y \end{bmatrix}$; we get :

$$\bar{\nu}_{opt} = G^{-1} b$$

This approach is valid only if the displacement is small, and to obtain a more accurate solution, we proceed iteratively.

Iterative Lucas and Kanade Algorithm

General Formulation : Iterative Version

- k : iteration index.
- $\bar{\nu}^{k-1} = [\nu_x^{k-1}\ \nu_y^{k-1}]^T$: hypothetical displacement derived from the first iterations $1, 2, ..., k-1$.
- J_k : new image translated according to $\bar{\nu}^{k-1}$:

$$\forall (w, y) \in W, J_k(x, y) = J_{k-1}(x + \nu_x^{k-1}, y + \nu_y^{k-1})$$

- The objective is to compute $\bar{\eta}^k = [\eta_x^k\ \eta_y^k]$ that minimizes :

$$\epsilon(\bar{\eta}^k) = \epsilon(\eta_x^k, \eta_y^k) = \sum\sum (I(x, y) - J_k(x + \eta_x^k, y + \eta_y^k))^2$$

Iterative Lucas and Kanade Algorithm

General Formulation : Iterative Version

- At iteration k :

$$\bar{\eta}^k = G^{-1} b_k$$

$$b_k = \sum \sum \begin{bmatrix} \delta I_k \, I_x \\ \delta I_k \, I_y \end{bmatrix}, \quad \delta I_k(x,y) = I(x,y) - J_k(x,y)$$

.

- I_x and I_y are computed only once at the beginning of the iteration.
- G is also constant.
- δI_k must be computed at each iteration.
- After computing $\bar{\eta}^k$, we deduce a new displacement hypothesis $\bar{\nu}^k$:

$$\bar{\nu}^k = \bar{\nu}^{k-1} + \bar{\eta}^k$$

- We iterate until $\bar{\eta}^k$ is smaller than a set threshold or until a fixed number of iterations.
- Then we have $\bar{\nu} = \mathrm{d} = \sum_{k=1}^{K} \bar{\eta}^k$.

Optical Flow Estimation : Differential Approach

Lucas & Kanade Approach : Limitations

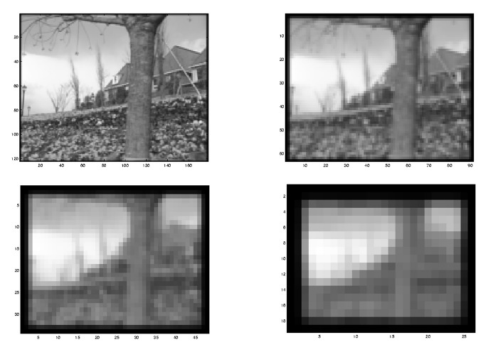
Assumption of small movement

- Is the movement small enough ?
  - Certainly not !
  - How to solve this problem ?

Optical Flow Estimation : Differential Approach

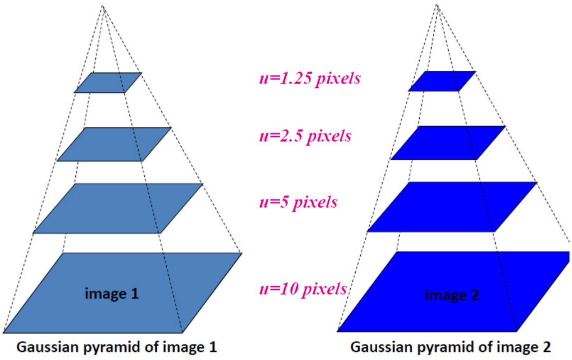
Lucas & Kanade Approach : Limitations

Idea : Reduce the resolution : reducing the image size reduces the observed movement.



Optical Flow Estimation : Differential Approach

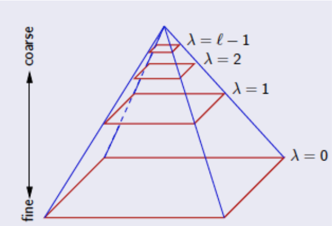
Multi-scale optical flow estimation

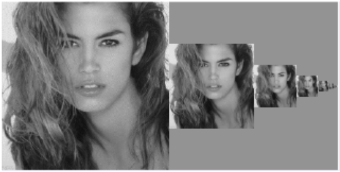


Optical Flow Estimation : Differential Approach

Multi-scale optical flow estimation : multi-resolution pyramid representation :

- Image pyramid : $I^{(\kappa)}$
- Pixel positions : $\mathbf{p}^{(\kappa)} = \frac{1}{2^\kappa}\mathrm{p}$



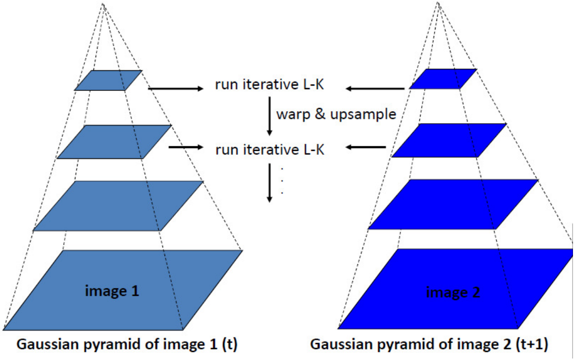


OpenCV

Image Pyramids (cv2.pyrUp() ; cv2.pyrDown())
`http://docs.opencv.org/doc/tutorials/imgproc/pyramids/pyramids.html`
`http://docs.opencv.org/trunk/doc/py_tutorials/py_imgproc/py_pyramids/py_pyramids.html`

Optical Flow Estimation : Differential Approach

Multi-scale optical flow estimation



Optical Flow Estimation : Differential Approach

Multi-scale optical flow estimation : algorithm

Goal: Let $\mathbf{u}$ be a point on image I. Find its corresponding location $\mathbf{v}$ on image J

Build pyramid representations of I and J: $\{I^L\}_{L=0,\dots,L_m}$ and $\{J^L\}_{L=0,\dots,L_m}$

Initialization of pyramidal guess: $\mathbf{g}^{L_m} = [g_x^{L_m} \;\; g_x^{L_m}]^T = [0 \;\; 0]^T$

for $L = L_m$ **down to** 0 **with step of** -1

- *Location of point $\mathbf{u}$ on image I^L:* $\mathbf{u}^L = [p_x \;\; p_y]^T = \mathbf{u}/2^L$
- *Derivative of I^L with respect to x:* $I_x(x,y) = \dfrac{I^L(x+1,y) - I^L(x-1,y)}{2}$
- *Derivative of I^L with respect to y:* $I_y(x,y) = \dfrac{I^L(x,y+1) - I^L(x,y-1)}{2}$
- *Spatial gradient matrix:* $G = \sum_{x=p_x-\omega_x}^{p_x+\omega_x} \sum_{y=p_y-\omega_y}^{p_y+\omega_y} \begin{bmatrix} I_x^2(x,y) & I_x(x,y)\, I_y(x,y) \\ I_x(x,y)\, I_y(x,y) & I_y^2(x,y) \end{bmatrix}$
- *Initialization of iterative L-K:* $\overline{\nu}^0 = [0 \;\; 0]^T$
- **for** $k = 1$ **to** K **with step of** 1 (or until $\|\overline{\eta}^k\| <$ accuracy threshold)
  - *Image difference:* $\delta I_k(x,y) = I^L(x,y) - J^L(x + g_x^L + \nu_x^{k-1}, y + g_y^L + \nu_y^{k-1})$
  - *Image mismatch vector:* $\overline{b}_k = \sum_{x=p_x-\omega_x}^{p_x+\omega_x} \sum_{y=p_y-\omega_y}^{p_y+\omega_y} \begin{bmatrix} \delta I_k(x,y)\, I_x(x,y) \\ \delta I_k(x,y)\, I_y(x,y) \end{bmatrix}$
  - *Optical flow (Lucas-Kanade):* $\overline{\eta}^k = G^{-1}\, \overline{b}_k$
  - *Guess for next iteration:* $\overline{\nu}^k = \overline{\nu}^{k-1} + \overline{\eta}^k$
- **end of for-loop on** k
- *Final optical flow at level L:* $\mathbf{d}^L = \overline{\nu}^K$
- *Guess for next level $L-1$:* $\mathbf{g}^{L-1} = [g_x^{L-1} \;\; g_y^{L-1}]^T = 2\,(\mathbf{g}^L + \mathbf{d}^L)$

end of for-loop on L

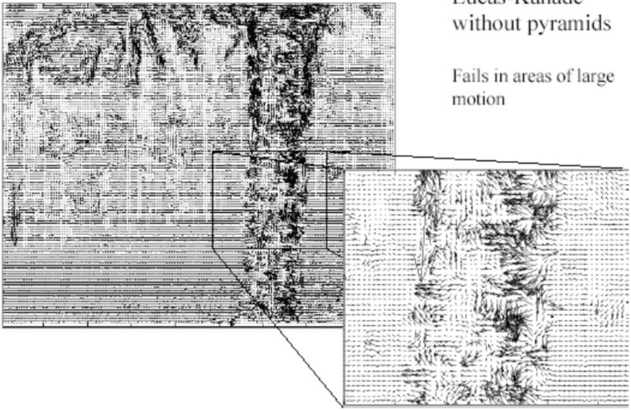
Final optical flow vector: $\mathbf{d} = \mathbf{g}^0 + \mathbf{d}^0$

Location of point on J: $\mathbf{v} = \mathbf{u} + \mathbf{d}$

Solution: The corresponding point is at location $\mathbf{v}$ on image J

Optical Flow Estimation : Differential Approach

Multi-scale optical flow estimation

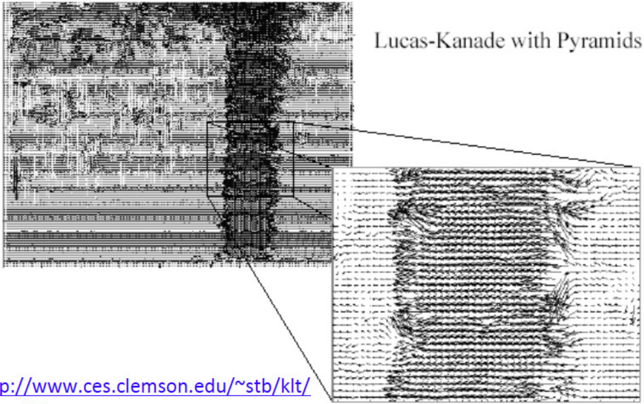


Lucas-Kanade
without pyramids

Fails in areas of large
motion

Optical Flow Estimation : Differential Approach

Multi-scale optical flow estimation



- http://www.ces.clemson.edu/~stb/klt/
- OpenCV

Optical Flow Estimation : Differential Approach

Horn & Schunk Approach

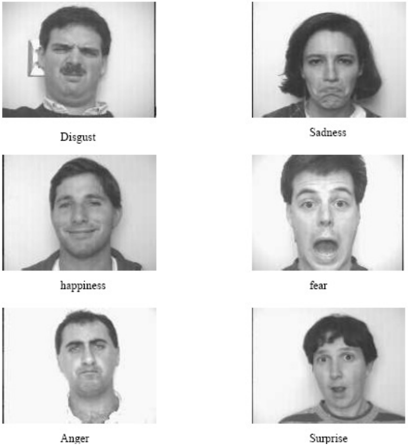
Adding constraints : the motion field must be locally smooth.

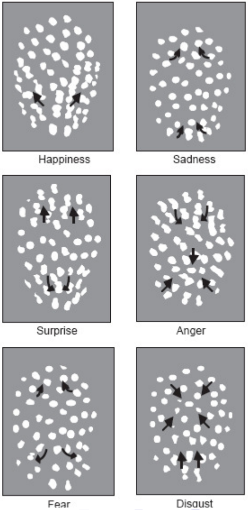
- For example : find v that minimizes :

$$\sum_{\mathbf{x}} \underbrace{\left(\frac{\partial I}{\partial t}(\mathbf{x}) + \nabla I^T(\mathbf{x}).\mathbf{v}(\mathbf{x})\right)^2}_{\text{Brightness constency}} + \lambda \underbrace{\left(||\nabla u(\mathbf{x})||^2 + ||\nabla v(\mathbf{x})||^2\right)}_{\text{Spatial coherence}}$$

Optical Flow Estimation : Differential Approach

Application of optical flow





Optical Flow Estimation : Differential Approach

Application of optical flow

http://www.fxguide.com/featured/art_of_optical_flow/

Optical Flow Estimation : Differential Approach

Application of optical flow

http://www.fxguide.com/featured/art_of_optical_flow/

Outline

What are good features to track ?

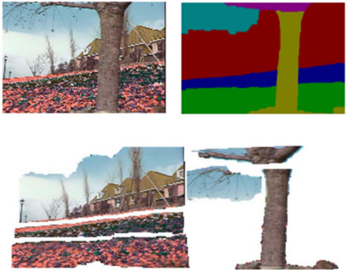
- We have seen all of this before... for example, Harris corners.
- Same algorithm as for optical flow calculation : Lucas Kanade.

Outline

Motion Segmentation

Layered Representation

Divide the image sequence into layers corresponding to different affine motions.



Motion Segmentation

Affine Motion

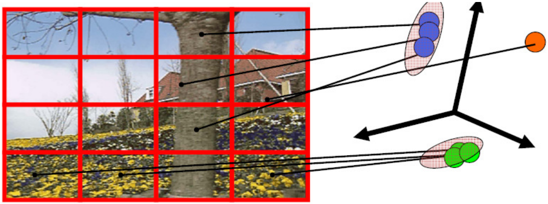
- $u(x, y) = a_1 + a_2 x + a_3 y$
- $v(x, y) = a_4 + a_5 x + a_6 y$
- Substitution into the brightness equation : $0 \approx \frac{\partial I}{\partial t} + \frac{\partial I}{\partial x} u + \frac{\partial I}{\partial y} v$
- $0 \approx \frac{\partial I}{\partial t} + \frac{\partial I}{\partial x}(a_1 + a_2 x + a_3 y) + \frac{\partial I}{\partial y}(a_4 + a_5 x + a_6 y)$
- Least squares minimization :

  $Err(a) = \sum \left[\frac{\partial I}{\partial t} + \frac{\partial I}{\partial x}(a_1 + a_2 x + a_3 y) + \frac{\partial I}{\partial y}(a_4 + a_5 x + a_6 y) \right]^2$

Motion Segmentation

How to estimate the layers ?

1. Obtain a set of hypotheses for the affine motions
   - Divide the image into blocks and estimate affine motion parameters in each block using the least squares method.
   - Eliminate hypotheses with a large residual error.
   - Consider the motion parameter space $(a_1 \dots a_6)$.
   - K-means algorithm on the parameters : merge close groups and retain the largest groups to obtain a smaller set of hypotheses describing the scene motion.

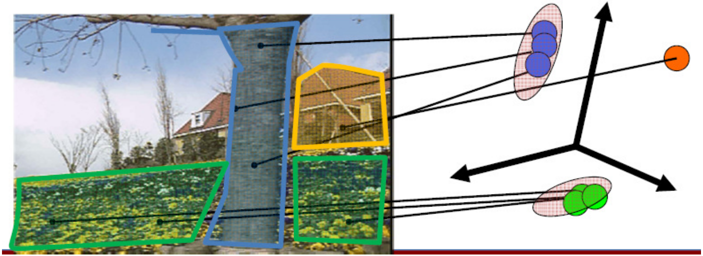


Motion Segmentation

How to estimate the layers ?

1. Obtain a set of hypotheses for the affine motions
   - Divide the image into blocks and estimate affine motion parameters in each block using the least squares method.
   - Eliminate hypotheses with a large residual error.
   - Transform into the motion parameter space.
   - K-means algorithm on the parameters : merge close groups and retain the largest groups to obtain a smaller set of hypotheses describing the scene motion.
2. Iterate until convergence :
   - Assign the best hypothesis to each pixel.
     - Pixels with the highest residual error remain unassigned.

Motion Segmentation



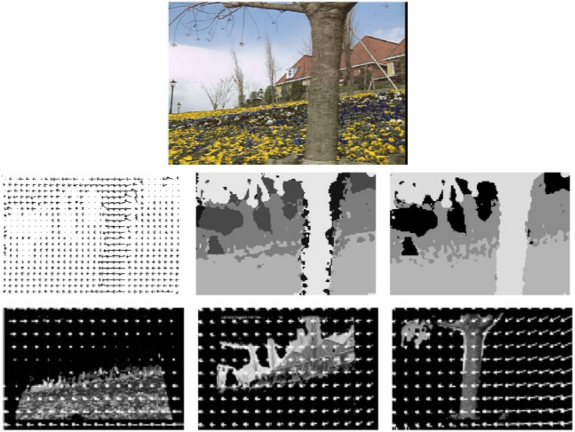
Motion Segmentation

How to estimate the layers ?

1. Obtain a set of hypotheses for the affine motions
   - Divide the image into blocks and estimate affine motion parameters in each block using the least squares method.
   - Eliminate hypotheses with a large residual error.
   - Transform into the motion parameter space.
   - K-means algorithm on the parameters : merge close groups and retain the largest groups to obtain a smaller set of hypotheses describing the scene motion.
2. Iterate until convergence :
   - Assign the best hypothesis to each pixel.
     - Pixels with the highest residual error remain unassigned.
   - Filter the regions to reinforce spatial constraints.
   - Re-estimate affine motion in each region.

Motion Segmentation

Example results



http://persci.mit.edu/demos/jwang/garden-layer/layer-demo.html

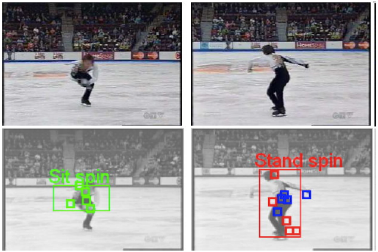
Motion Segmentation

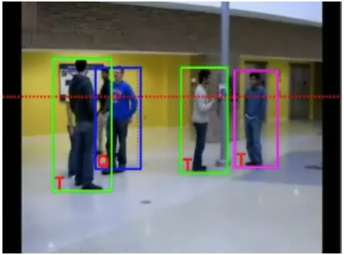
Example results

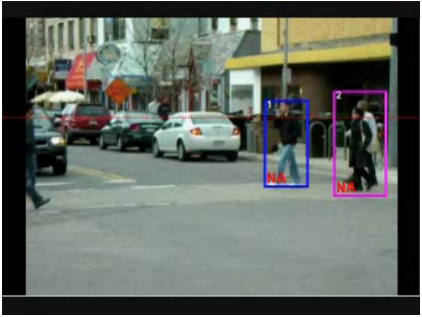
http://persci.mit.edu/demos/jwang/garden-layer/layer-demo.html

Plan

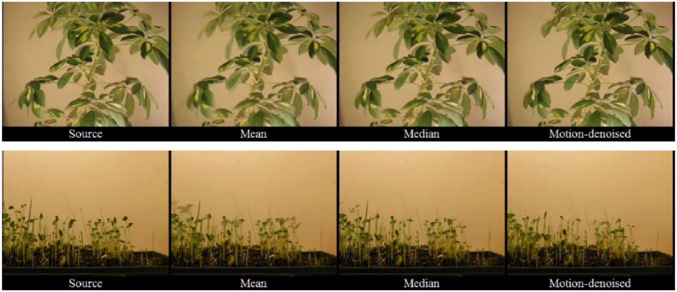
Activity Recognition







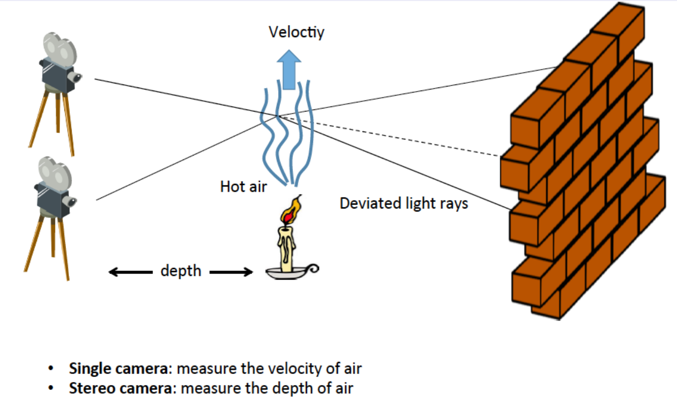
Stabilization - Video Denoising



- http://people.csail.mit.edu/mrub/timelapse/
- http://research.microsoft.com/en-us/um/people/jiansun/papers/CVPR14_SteadyFlow.pdf

Some Recent Work

Refraction Wiggles

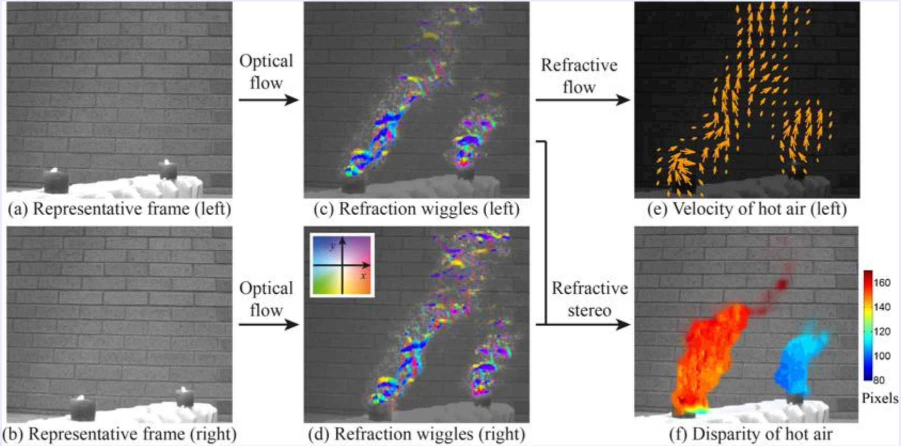


Refraction Wiggles for Measuring Fluid Depth and Velocity from Video *Tianfan Xue, Michael Rubinstein, Neal Wadhwa, Anat Levin, Fredo Durand, and William T. Freeman*, ECCV 2014. `http://people.csail.mit.edu/tfxue/proj/fluidflow/`

Some Recent Work

Refraction Wiggles

http://people.csail.mit.edu/tfxue/proj/fluidflow/

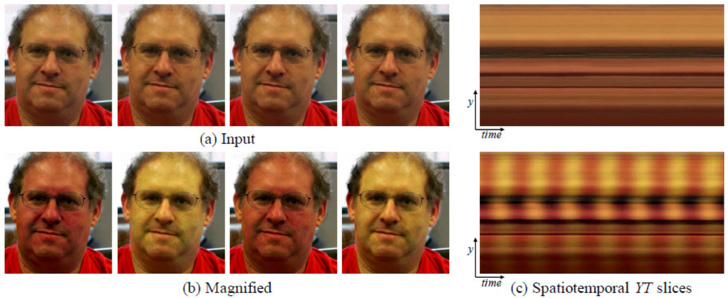


(a) Representative frame (left)
(c) Refraction wiggles (left)
(e) Velocity of hot air (left)
(b) Representative frame (right)
(d) Refraction wiggles (right)
(f) Disparity of hot air

Some Recent Work

Video Magnification

Revealing Invisible Changes In The World



(a) Input
(b) Magnified
(c) Spatiotemporal *YT* slices

`https://www.youtube.com/watch?v=e9ASH8IBJ2U`.
Hao-Yu Wu, Michael Rubinstein, Eugene Shih, John V. Guttag, Frédo Durand, William T. Freeman : Eulerian video magnification for revealing subtle changes in the world. ACM Trans. Graph. 31(4) : 65 (2012)

Some Recent Work

The Visual Microphone

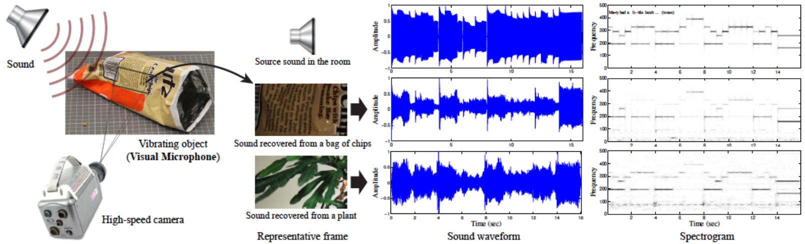


Figure 1: *Recovering sound from video. Left: when sound hits an object (in this case, an empty bag of chips) it causes extremely small surface vibrations in that object. We are able to extract these small vibrations from high speed video and reconstruct the sound that produced them - using the object as a visual microphone from a distance. Right: an instrumental recording of "Mary Had a Little Lamb" (top row) is played through a loudspeaker, then recovered from video of different objects: a bag of chips (middle row), and the leaves of a potted plant (bottom row). For the source and each recovered sound we show the waveform and spectrogram (the magnitude of the signal across different frequencies over time, shown in linear scale with darker colors representing higher energy). The input and recovered sounds for all of the experiments in the paper can be found on the project web page.*

`http://people.csail.mit.edu/mrub/VisualMic/`

Abe Davis, Michael Rubinstein, Neal Wadhwa, Gautham J. Mysore, Frédo Durand, William T. Freeman : The visual microphone : passive recovery of sound from video. ACM Trans. Graph. 33(4) : 79 (2014)

Plan

Conclusion

- Motion : an important feature in vision.
- Several approaches for :
  - Detecting.
  - Estimating.
  - Segmenting.
- Numerous applications.