Video Analysis with Deep Learning and Without it Lecture 1. Motion estimation Pr. Jenny Benois-Pineau LABRI UMR 5800/Université Bordeaux



Summary of the cours

- 1. Introduction.
- 2. Motion Data. Motion Estimation
 - 2.3.1. Estimation of sparse motion fields
 - 2.3.2. Dense Optical flow estimation
- 3. Perception of Video Scenes
- 4. Incremental learning

1. Introduction(1)

- Video analysis is a part of a large area which is multimedia data mining including : images, video, data and text.

- The ever growing porgress in ML and specifically in DL allows for building efficient frameworks for video analysis and porcessing.

- Amongst the tasks we can cite :
- action recognition in video;
- - object recognition and localization in video;
- object tracking;
- - fake video detection
- agmented reality...

Introduction (2)



→ Images : pixel values in grey-level or colour spaces



How video differs from an image?



How video differes from an Image?



Motion in Classification Problems



2.1. Colour Systems (Reminder)(1)

 \rightarrow RGB – the system which is the closest to acquisition/: reproduction devices LCD, TV, scanner.



Colour Systems (Reminder)(2)

→ RGB is used e.g. in video cameras RGB-D (TOF, Microsoft Kinect).



- → Avantages : allows for a precise colour acquisition
- → Drawbacks :
 - > The RGB system is not efficient wrt the criterion efficiency/compactness.
- → Colour systems which separatly encode luminance and chrominance are efficient (YUV, YCrCB, YIQ,...). They are used for video.

Colour Systems (Reminder)(2)

Linear transformations -> RGB - > YUV -> RGB

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} 0,299 & 0,578 & 0,114 \\ -0,14713 & -0.28886 & 0.436 \\ 0,615 & -0.51498 & -0,10001 \end{bmatrix} \times \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1,13983 \\ 1 & -0,39465 & -0,58060 \\ 1 & 2,03211 & 0 \end{bmatrix} \times \begin{bmatrix} Y \\ U \\ V \end{bmatrix}$$

Origines : TV Pal (YUV), YCrCb – are independent from hardware.

2.2. Spatial sampling (video)



RGB

YUV

YUV: 4:0:0

YUV : 4:4:4 – the same resolution as a target RGB

YUV 4:2:2



26/11/2024 / IPB Deep and Non Deep Video Analysis

2.3 Motion Data. Motion Estimation

Problems: action recognition, moving object detection and tracking, violence recognition, activity caractérisation, interest detection....

Measuring motion: Frame difference (FD)

- \rightarrow *I*(*x*,*y*,*t*) intensity function)
- → Time instants: *t, t-dt*



$$FD(x, y, t) = I(x, y, t) - I(x, y, t - dt)$$

What is the quantity of information in FD vs to the original video sequence? 26/11/2024 / IPB Deep and Non Deep Video Analysis

Examples of FD





Quantity of Information. Entropy(I)

→ Entropy

- → Let S be a source of information and x_i (symbols of an alphabeth) possible values of information. In case of an image these are the values of intensity (0, ...255 in the case of 8 bit depth)
- → We can model the source S by a random variable X with values x_i following the pdf $p(x_i)$

\rightarrow

→ The Entropy of a random variable X is:

$$H(X) = -\sum_{i=0}^{N-1} p(x_i) \log_2 p(x_i)$$



→ Intensity depth 8 bit

→ Values $X = \{0, 1, ..., 255\}$

→ Maximal Entropy $H(X) = \log_2(K) = \log_2(256) = 8$



Motion detection: an elementary approach

→ If $|FD(x, y, t)| \ge Seuil$ the the pixel(x,y) is in motion (*)

→ Draw-backs : noise, difficulties to set-uo the threshold

→ Prise en compte du bruit et de variations de luminance: -Pour chaque pixel (*x*,*y*) apprentissage pendant Δt de

- $m(x, y) = \min_{t=t_0, \dots, t_0 + \Delta t} (I(x, y, t))$

$$M(x, y) = Max_{t=t_0, \dots, t_0 + \Delta t} (I(x, y, t))$$

Pour le couple d'images calcul de

$$D(x, y, t) = \max_{(x, y)} \left| FD(x, y, t) \right|$$

Détection d'un « pixel en mouvement » si

$$|m(x, y) - I(x, y, t)| > D(x, y, t) \quad ou$$
$$|M(x, y) - I(x, y, t)| > D(x, y, t)$$

Motion detection per block

 \rightarrow Consider 2 windows centered on a pixel (*x*,*y*)



- → Statistical Hypotheses:
- → H0 : distributions of the luminance in both windows have the same statistical characteristics at t and t+dt
- \rightarrow H1 : the distributions are different.

→ Model of luminance

$$I(x,y) = \mu + N(0,\sigma^2)$$



→ Likelihood functions relative to hypotheses

$$L(H_0) = \prod_{(x_A, y_A) \in A_1 \cup A_2} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(I(x_A, y_A) - \mu_0)^2}{2\sigma^2}\right)$$

$$L(H_{1}) = \prod_{(x_{A}, y_{A}) \in A_{1}} \frac{1}{\sqrt{2\pi\sigma^{2}}} \exp\left(-\frac{(I(x_{A}, y_{A}) - \mu_{1})^{2}}{2\sigma^{2}}\right) \times \prod_{(x_{A}, y_{A}) \in A_{2}} \frac{1}{\sqrt{2\pi\sigma^{2}}} \exp\left(-\frac{(I(x_{A}, y_{A}) - \mu_{2})^{2}}{2\sigma^{2}}\right)$$

- → Maximum likelihood test:
- → Logarithmic likelihood ratio

$$R = \ln \frac{L(H_1)}{L(H_0)} \stackrel{>}{\underset{H_0}{\overset{\times}{\sim}}} \lambda$$

$$R = \frac{1}{2\sigma^{2}} \left\{ \begin{array}{c} \sum_{A_{1}\cup A_{2}} \left(I\left(x_{A}, y_{A}\right) - \mu_{0}\right)^{2} - \\ -\sum_{A_{1}} \left(I\left(x_{A1}, y_{A1}\right) - \mu_{1}\right)^{2} - \sum_{A_{2}} \left(I\left(x_{A2}, y_{A2}\right) - \mu_{2}\right)^{2} \end{array} \right\}$$
$$\hat{R} = \max \min_{\mu_{1}, \mu_{2}} \min_{\mu_{0}} R$$

 \rightarrow To simplify, consider the sqr of *R*

$$\frac{H_1}{\sqrt{R}} = \frac{n}{2\sigma} |\hat{\mu}_1 - \hat{\mu}_2| < \frac{\lambda'}{\kappa}$$

$$\frac{H_1}{\kappa} = \frac{n}{2\sigma} |\hat{\mu}_1 - \hat{\mu}_2| < \frac{\lambda'}{\kappa}$$

$$FD(x, y, t+dt) = I(x, y, t+dt) - I(x, y, t)$$

$$\frac{1}{2n\sigma} \left| \sum_{A} FD(x, y, t + dt) \right| \stackrel{>}{<} \lambda'$$

$$H_0$$

Algorithm

- → In a homogenious area
- → estimate μ and $\sigma = \sqrt{(I(x,y) \mu)^2 / N}$

 \rightarrow (suppose σ - constant in the sequence)

- → For each block-window A
 - > Compute the sum of *FD*

$$\frac{H_1}{\frac{1}{2n\sigma}} \left| \sum_{A} FD(x, y, t + dt) \right| \stackrel{>}{<} \lambda'$$

$$H_0$$
and compare with a threshold

Example of block-wise motion detection (variable size)



Motion, displacement field(I)

→ Projection of a rigid motion into an image plane



Motion, displacement field(II)

→ Projected motion/apparent motion



→ Insufficient spatial gradient

Motion, displacement field(III)

→ $d(x,y,t) = (dx(x,y,t), dy(x,y,t))^T$ is the projection of a real 3D displacement into the image plane



→ Vector field $\mathbf{D}(x, y) = \{(dx, dy)^T\}$ is called « dense motion field »

→ Velocity vector $\mathbf{w}(x, y)$ corresponds up to *dt* to the displacement vector

$$\mathbf{w}(x, y, t) = \left(\frac{dx}{dt}, \frac{dy}{dt}\right)^T$$

Examples(I)



Examples(II)



Motion Estimation

 \rightarrow The goal of motion estimation : to obtain vector field Γ

$$\mathbf{D}(x,y) = \left\{ \left(dx, dy \right)^T \right\}$$

- \rightarrow i) dense, i.e. pixel-based
- → li) sparse, i.e; patch/block-based
- → Motion estimation methods :
- → i) direct direct estimation of motion field
- → Ii) parametric : supposing a global motion model in the image plane to estimate model parameters;

2.3.1. Estimation of a sparse motion fileds(I)

→ Block-based methods. Full search

 $\mathbf{d}_{i}(x, y, t) = \left(dx(x, y, t), dy(x, y, t)\right)_{i}^{T}$

→ Direct estimation



 $\mathbf{D}^{*}(x,y) = \{(dx, dy)^{T}\}^{*} = \underset{(dx, dy)^{T} \in F}{\operatorname{Argmin}}(dx, dy)^{T} (E\{(dx, dy)^{T}\})$

→ If $E \{(dx, dy)^{T}\}$ is additive, then → to minimize $E \{(dx, dy)^{T}\}$ is equivalent to the minimization of $\sum E_i \{(dx, dy)_i^{T}\}$

Motion estimation by block-matching(II)

→ Partition of each frame into blocks (regular grid or irregular grid)







 $I(t - \Delta t)$

I(t)

Reference frame

Current frame

→ « Backward » Motion estimation vs « Forward » Motion estimation

 \rightarrow The displacement field **D**(x,y) is supposed to be piece-wise constant

Optimisation criteria

→ Minimal Absolute Difference (MAD)

$$MAD(\mathbf{d}) = \min_{\vec{d}\in F} \sum_{\vec{p}\in B} |I(\mathbf{p},t) - I(\mathbf{p}+\mathbf{d},t-\Delta t)| \qquad \mathbf{p} = (x,y)^T$$

→ Mean squared error

$$MSE(\mathbf{d}) = \min_{\vec{d} \in F} \frac{1}{Card(B)} \sum_{\vec{p} \in B} \left(I(\mathbf{p}, t) - I(\mathbf{p} + \mathbf{d}, t - \Delta t) \right)^2$$

MAD and MSE are additive, hence to optimase a vector for each block means to optimize the motion field for a given frame

Full search method (I)

→ Estimation with « pixel precision »



 $I(t - \Delta t)$ I(t)

Principle: « move » the block with integer coordinates of displacement vector

Full search method (II)

- → Algorithm for search with « pixel accuracy »
- → Input : $(x, y), F, I(t), I(t \Delta t)$
- → Output: Optimal displacement vector (dx*,dy*)
- → For each block
 - E0 = MaxInt
 - $dx^* = -F/2$
 - $dy^* = -F/2$
 - > For dx from –F/2 to F/2 with step 1
 - > For dy from -F/2 to F/2 with step 1
 - E = ComputeCriterion(dx,dy,x,y, $I(x,y,t), I(x+dx, y+dy, t-\Delta t)$)
 - If E < E0
 - > then
 - $dx^* = dx$
 - \rightarrow dy^{*} = dy
 - > Endlf
 - > E0 = E
 - > EndFor
 - > EndFor

³⁵ → EndFor

26/11/2024 / IPB Deep and Non Deep Video Analysis

Full search method (III)

→ Advantage: yields an optimal vector in the window F

- → Major drawback: computational cost
 - > Elementary operation: computation of $f(I(x, y, t), I(x + dx, y + dy, t \Delta t))$
 - > # of operations for a block of size *NxN* pixels:

$$C(N,F) = F^2 N^2$$

> Maximal displacement:

$$F/\sqrt{2}$$

>

>
Example of motion field

→ Blocks 8x8



Diamond search (Zhu, Ma, MPEG4 VS, 1997)

→ Sub-optimal method with a « diamond » search pattern



- first search
 If the optimum is in the center of the
 « diamond »– then perform sear on the smallest « diamond »
- On the border of the search window
 - -On the corner

-In the two former cases the search is done on the diamond of the same size



- \rightarrow of results
- → Displacement
- → field



Diamond search

→ Example of results



Search with a half-pixel accuracy

 Between current frame and interpolated reference frame: refinement of the search



 $I(x + \Delta x, y + \Delta y) = I_1 (1 - \Delta x) (1 - \Delta y) + I_2 \Delta x (1 - \Delta y) + I_3 \Delta x \Delta y + I_4 (1 - \Delta x) \Delta y$

Example of a motion-compensated frame with variable block-size



352x288 pels, original frame,

Ian E.G. Richardson H.264 and MPEG-4 Video compression, John Wiley & Sons Ltd, 2003, http://lib.mdp.ac.id/ebook/Karya%20Umum/Video-Compression-Video-Coding-for-Next-⁴generation-Multimedia.pdf ^{26/11/2024} / IPB Deep and Non Deep Video Analysis

Example of a motion compensated frame with variable block-size



Motion compensation with 16x16 Blocks

2.3.2. Dense Optical flow estimation

→ Optical flow equation

→ Fondamental hypothesis of motion estimation: conservation of intensity of a point along the trajectory

$$DFD(x, y, \mathbf{d}) = I(x + dx, y + dy, t + dt) - I(x, y, t) = 0$$

→ Motion estimation problem is an ill-posed one:

(1) Problem of existence–occlusion









→ Development

$$\rightarrow DFD(x, y, \mathbf{d}) = I(x + dx, y + dy, t + dt) - I(x, y, t) = 0$$
(1)

- → Developing I(x+dx, y+dy, t+dt) around the point (x, y, t)
- → And supposing the linearity of I(x, y, t) (2nd derivatives are nulls)

→ From (1):
$$I(x + dx, y + dy, t + dt) = I(x, y, t) + dx \frac{\partial I}{\partial x} + dy \frac{\partial I}{\partial y} + dt \frac{\partial I}{\partial t}$$

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

→ Finally

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

26/11/2024 / IPB Deep and Non Deep Video Analysis

45

Different forms of OFE (I)

→ Let us denote the velocity vector $\mathbf{w} = (u, v)^T$

$$\frac{\partial I}{\partial x}u + \frac{\partial I}{\partial y}v = -\frac{\partial I}{\partial t}$$

 \rightarrow (2)The second reason why the problem of motion estimation is ill-posed.

→ Supposing independence of u and v – only one equation for two variables – the uniqueness of the solution is not ensured.

Different forms of OFE (II)

→ In a vector form

→ here
$$\nabla I = \left(\frac{\partial I}{\partial x}; \frac{\partial I}{\partial y}\right)^T$$
 is the image gradinet

→ Let us de-compose the velocity vector on a new orthogonal basis



Different forms of OFE (III)

→ OFE :

$$\nabla I \cdot (\mathbf{w}_{\perp} + \mathbf{w}_{\mathrm{II}}) = -I_t$$

→ \vec{w}_{\perp} is the component parallel to the image gradient $\vec{\nabla}I$ Therefore it is orthogonal to the local contour.

It is the only observable one: « normal optical flow ».

Uniqueness of the solution-pb, « problem of aperture »

(3) Continuity : Motion estimation is very sensitive to noise, unstable : a weak noise can yield a large difference in the solution

Estimation of Optical flow with regularization Horn & Schunk method (I)

- \rightarrow A local continuity of OF is hypothesized.
- Horn and Schunk's method : Add regularization constraint when estimating optimal OF (u,v) by minimisation of the error functional on the domain G in image plane:

$$E(u,v) = \int_{G} \left[\left(I_{x}u + I_{y}v + I_{t} \right)^{2} + \alpha^{2} \left(\left(\frac{\partial u}{\partial x} \right)^{2} + \left(\frac{\partial u}{\partial y} \right)^{2} + \left(\frac{\partial v}{\partial x} \right)^{2} + \left(\frac{\partial v}{\partial y} \right)^{2} \right] dxdy \rightarrow \min$$

→ Here $E_{OF} = I_x u + I_y v + I_t$ is the error term of OFE

$$\rightarrow E_w^2 = \|\vec{\nabla}u\|^2 + \|\vec{\nabla}v\|^2$$
 is the regularization term

Horn&Schunk's method(II)

$$\min_{w} \int_{G} (I_{x}u + I_{y}v + I_{t})^{2} + \alpha^{2} \left(\left(\frac{\partial u}{\partial x} \right)^{2} + \left(\frac{\partial u}{\partial y} \right)^{2} + \left(\frac{\partial v}{\partial x} \right)^{2} + \left(\frac{\partial v}{\partial y} \right)^{2} \right) dx dy$$

We thus have a functional of several functions of several independent variables

$$\iint_{G} f(x, y, u(x, y), v(x, y), \frac{\partial u}{\partial x}, \frac{\partial u}{\partial y}, \frac{\partial v}{\partial x}, \frac{\partial v}{\partial y}) dxdy$$

The solutions satisfy Euler – Ostrogradski system

$$\begin{cases} \frac{\partial f}{\partial u} - \frac{\partial}{\partial x} \left(\frac{\partial f}{\partial u_x} \right) - \frac{\partial}{\partial y} \left(\frac{\partial f}{\partial u_y} \right) = 0 \\ \frac{\partial f}{\partial v} - \frac{\partial}{\partial x} \left(\frac{\partial f}{\partial v_x} \right) - \frac{\partial}{\partial y} \left(\frac{\partial f}{\partial v_y} \right) = 0 \end{cases}$$

Horn&Schunk's method(III)

After the calculus

$$\left(\frac{\partial I}{\partial x}\right)^2 u + \frac{\partial I}{\partial y}\frac{\partial I}{\partial x}v = \alpha^2 \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) - \frac{\partial I}{\partial x}\frac{\partial I}{\partial t}$$
$$\frac{\partial I}{\partial x}\frac{\partial I}{\partial y}u + \left(\frac{\partial I}{\partial y}\right)^2v + \alpha^2 \left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2}\right) - \frac{\partial I}{\partial y}\frac{\partial I}{\partial t}$$

Here

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

Are the Laplacians of functions *u* and *v*

Horn&Schunk's method(IV)

H&S approximation of Laplacian

$$\Delta u = \left(\overline{u}_{i,j,t} - u_{i,j,t}\right)$$

$$\Delta v = \left(\overline{v}_{i,j,t} - v_{i,j,t}\right)$$
(*)
$$\overline{u} - u =$$

$$1/12 \quad 1/6 \quad 1/12$$

$$1/6 \quad -1 \quad 1/6$$

$$1/12 \quad 1/12$$

Using (*) the system can be re-written as

$$\begin{cases} \left(\alpha^2 + I_x^2\right)u + I_xI_yv = \alpha^2\overline{u} - I_xI_t\\ I_xI_yu + \left(\alpha^2 + I_y^2\right)v = \alpha^2\overline{v} - I_yI_t \end{cases}$$

The solution is either direct, either iterative

26/11/2024 / IPB Deep and Non Deep Video Analysis



Horn&Schunk's method(IV)

Iterative solution

$$\begin{cases} \left(\alpha^2 + I_x^2\right)u + I_xI_yv = \alpha^2\overline{u} - I_xI_t \\ I_xI_yu + \left(\alpha^2 + I_y^2\right)v = \alpha^2\overline{v} - I_yI_t \end{cases}$$

Iterative method of Gauss-Seidel:

$$\hat{u}^{(i+1)} = \overline{u}^{(i)} - \frac{\partial I}{\partial x} \frac{\nabla I \cdot \overline{\mathbf{w}}^{(i)} + \frac{\partial I}{\partial t}}{\alpha^2 + \left\|\nabla I\right\|^2}$$

Here $\overline{w}, \overline{u}, \overline{v}$, are the weighted means in the neighbourhood

$$\hat{v}^{(i+1)} = \overline{v}^{(i)} - \frac{\partial I}{\partial y} \frac{\nabla I \cdot \overline{\mathbf{w}}^{(i)} + \frac{\partial I}{\partial t}}{\alpha^2 + \left\| \nabla I \right\|^2}$$

Example of motion estimation by H&S method (I)

- → Sequence HD, zoom. Initialisation with null vectors
- → Estimation in mono-resolution



Example of motion estimation by H&S method (II)



 MSE^{0} , MSE, #iter = 6



Multi-resolution motion estimation(I)

- → Principle
- → For a pair of consecutive images I(t), I(t-dt)
- → repeat L times
- → Lowpass filter

$$I(t)^{f} = g \circ I(t)$$
$$I(t - dt)^{f} = g \circ I(t - dt)$$

- → Subsample.
- → Result : $\tilde{I}(t)^{l}$ $\tilde{I}(t-dt)^{l}, l=0,...,L-1$
- « Top –down » motion estimation with initialisation by projection of OF accros the resolution levels

Multi-resolution motion estimation(II)



$$\mathbf{w}(\rho x, \rho y)^{l_0} = \rho \mathbf{w}(x, y)^l \qquad \mathbf{w}(x, y)^{L-1,0} = \vec{0}$$

Interpolate coordinates for pixels – Linear interpolation is sufficient $\mathbf{w}(\rho x, \rho y)^{li} = \rho \mathbf{w}(x, y)^{l} + \mathbf{e}(x, y)^{i}$

2.3.2. Parametric Motion Estimation

→ Hierarchy of motion models

$$\begin{cases} dx = t_x \\ dy = t_y \end{cases}$$

$$\begin{cases} dx = t_x + k(x - x_g) \\ dy = t_y + k(y - y_g) \end{cases}$$

$$\int dx = x_2 - x_1 = t_x + k(x_1 - x_{g1})$$

$$\begin{cases} dx = x_2 - x_1 = t_x + k(x_1 - x_{g1}) - \theta(y_1 - y_{g1}) \\ dy = y_2 - y_1 = t_y + \theta(x_1 - x_{g1}) + k(y_1 - y_{g1}) \end{cases}$$

$$\begin{cases} dx \\ dy \end{cases} = \begin{pmatrix} a_0 \\ b_0 \end{pmatrix} + \begin{pmatrix} a_1 & a_2 \\ b_1 & b_2 \end{pmatrix} \begin{pmatrix} x - x_g \\ y - y_g \end{pmatrix}$$

$$\begin{cases} dx \\ dy \end{cases} = \begin{pmatrix} a_0 \\ b_0 \end{pmatrix} + \frac{1}{2} \begin{pmatrix} div \cdot (x - x_g) - rot \cdot (y - y_g) + hyp1 \cdot (x - x_g) + hyp2 \cdot (y - y_g) \\ div \cdot (y - y_g) + rot \cdot (x - x_g) - hyp1 \cdot (y - y_g) + hyp2(x - x_g) \end{pmatrix}$$

Motion models. Homography (8 parameters)

→ 1. Homography (3D->2D)

$$\begin{bmatrix} x_2 \\ y_2 \\ z_2 \end{bmatrix} = \begin{bmatrix} H_{11} & H_{12} & H_{13} \\ H_{21} & H_{22} & H_{23} \\ H_{31} & H_{32} & H_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} \Leftrightarrow \mathbf{x}_2 = H\mathbf{x}_1$$

→ In inhomogeneous coordinates $(x'_2 = x_2/z_2 \text{ and } y'_2 = y_2/z_2)$,

. . . .

$$\begin{split} x_2' &= \frac{H_{11}x_1 + H_{12}y_1 + H_{13}z_1}{H_{31}x_1 + H_{32}y_1 + H_{33}z_1} \\ y_2' &= \frac{H_{21}x_1 + H_{22}y_1 + H_{23}z_1}{H_{31}x_1 + H_{32}y_1 + H_{33}z_1} \end{split}$$

Homography(2)

 \rightarrow Without loss of generality setting $z_1 = 1$:

$$\begin{aligned} x_2'(H_{31}x_1 + H_{32}y_1 + H_{33}) &= H_{11}x_1 + H_{12}y_1 + H_{13} \\ y_2'(H_{31}x_1 + H_{32}y_1 + H_{33}) &= H_{21}x_1 + H_{22}y_1 + H_{23} \end{aligned}$$

→ Re-arranging

$$\mathbf{a}_x^T \mathbf{h} = \mathbf{0}$$
$$\mathbf{a}_y^T \mathbf{h} = \mathbf{0}$$

$$\Rightarrow With \qquad \mathbf{h} = (H_{11}, H_{12}, H_{13}, H_{21}, H_{22}, H_{23}, H_{31}, H_{32}, H_{33})^T \\ \mathbf{a}_x = (-x_1, -y_1, -1, 0, 0, 0, x'_2 x_1, x'_2 y_1, x'_2)^T \\ \mathbf{a}_y = (0, 0, 0, -x_1, -y_1, -1, y'_2 x_1, y'_2 y_1, y'_2)^T.$$

Homography(3)

→ Finally, given a set of corresponding points

 $A\mathbf{h} = \mathbf{0}$

$$A = \begin{pmatrix} \mathbf{a}_{x1}^T \\ \mathbf{a}_{y1}^T \\ \vdots \\ \mathbf{a}_{xN}^T \\ \mathbf{a}_{yN}^T \end{pmatrix}.$$

→ Solution wrt H – Least squares

Inconvenience : 9 parameters, sensitivity to outliers when solving by least squares

David Kriegman

http://cseweb.ucsd.edu/classes/wi07/cse252a/homography_estimation/

RANSAC for global motion models estimation

Random Sample Consensus

- Principle : iterative algorithm which allows for discarding outliers
- \rightarrow 1. Randomly choose a set of measures
- \rightarrow 2. Estimate the model for the set of selected measures(MC)
- → 3. Find the outliers amongst all available values and discard them (consensus set)

$$\left\| \mathbf{d}(\boldsymbol{\theta}, \mathbf{x}) - \mathbf{d}_0(\mathbf{x}) \right\| \le Th \qquad (r_i = z_i - \hat{z}_i)$$

 \rightarrow 4. Repeat 1 – 3 Nbrlt times

Quality criteria for motion estimation. How to choose the best estimator(1)

- \rightarrow (1) Without ground trouth optical flow
- → Mean Square Motion compensation error

$$MSE = rac{\sum_{x=0}^{W_I-1} \ \sum_{y=0}^{H_I-1} (I(x,y,t+1) - I(x+u,y+v,t))^2}{W_I * H_I} \ = rac{\sum_{p\in\Omega} (I(p) - I_C(p))^2}{N_\Omega}$$

→ Priniciple: comparison of original frame I(x,y,t+1) with motioncompensated frame I(x+u,y+v)

Quality criteria for motion estimation. How to choos the best estimator(2)

- → If the reference optical flow (u^*, v^*) is available:
- \rightarrow Angular error (AE) and average Angular Error (aAE)

$$egin{aligned} AE_{x,y} &= \cos^{-1}(rac{\epsilon+u^{*}\,*\,v+v^{*}\,*\,v}{\sqrt{\epsilon+u^{*2}+v^{*2}}\,*\,\sqrt{\epsilon+u^{2}+v^{2}}}) \ aAE &= rac{\sum_{x=0}^{W_{I}-1}\;\sum_{y=0}^{H_{I}-1}AE_{x,y}}{\sqrt{W_{I}\,*\,H_{I}}} \end{aligned}$$

 \rightarrow End-Point Error (*EPE*) and average EPE (*aEPE*)

$$EPE = \|V^* - V\|_2 \ = \sqrt{(u^* - u)^2 + (v^* - v)^2} \ aEPE = rac{\sum_{x=0}^{W_I - 1} \ \sum_{y=0}^{H_I - 1} EPE_{x,y}}{W_I * H_I}$$

To benchmark



Sintel Benchmark

- A synthetic dataset with available GT optical flow
- https://ps.is.mpg.de/research_projects/mpi-sintel-flow

Optical Flow with Deep Learning: Challenges and Advances. Summary

- What is Optical Flow?
- Traditional approaches: based on energy models.
- Challenges: computational complexity, low accuracy in difficult conditions.
- Advantages of deep learning methods: accuracy, speed, adaptability.

State-of-the-Art. FlowNet



[1] Butler D. J. et al. A naturalistic open source movie for optical flow evaluation //Computer Vision–ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part VI 12. – Springer Berlin Heidelberg, 2012. – C. 611-625.

[2] Dosovitskiy A. et al. Flownet: Learning optical flow with convolutional networks //Proceedings of the IEEE international conference on computer vision. – 2015. – C. 2758-2766.
 [3] Alfarano A. et al. Estimating optical flow: A comprehensive review of the state of the art //Computer Vision and Image Understanding. – 2024. – C. 104160.

State-of-the-Art. RAFT

RAFT: high accuracy due to recursive processing [2]



[1] Butler D. J. et al. A naturalistic open source movie for optical flow evaluation //Computer Vision–ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part VI 12. – Springer Berlin Heidelberg, 2012. – C. 611-625.

[2] Teed Z., Deng J. Raft: Recurrent all-pairs field transforms for optical flow //Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16. – Springer International Publishing, 2020. – C. 402-419.

[3] Alfarano A. et al. Estimating optical flow: A comprehensive review of the state of the art //Computer Vision and Image Understanding. - 2024. - C. 104160.

State-of-the-Art. DDVM



Denoising Diffusion Vision Model (DDVM): surprisingly effective for depth and flow estimation [3]

[1] Butler D. J. et al. A naturalistic open source movie for optical flow evaluation //Computer Vision–ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part VI 12. – Springer Berlin Heidelberg, 2012. – C. 611-625.

[2] Saxena S. et al. The surprising effectiveness of diffusion models for optical flow and monocular depth estimation //Advances in Neural Information Processing Systems. – 2024. – T. 36. [3] Alfarano A. et al. Estimating optical flow: A comprehensive review of the state of the art //Computer Vision and Image Understanding. – 2024. – C. 104160.

References

Horn, B., Schunck, B.: 'Determining optical 691 flow: a retrospective', Artif. Intell., 59, (1–2), pp. 81– 692 87, (1993).

Fortun D., Bouthemy P., Kervrann Ch., "Optical flow modelling and computation: a Survey", in Computer and Vision Image Understanding, 134, 2015, pp. 1-21

Martin, P. E., Benois-Pineau, J., Péteri, R., & Morlier, J. (2019, September). Optimal choice of motion estimation methods for fine-grained action classification with 3d convolutional networks. In 2019 IEEE International Conference on Image Processing (ICIP) (pp. 554-558).

Roujol, S., Benois-Pineau, J., Denis de Senneville, B., Ries, M., Quesson, B., Moonen, Ch. T. W. :

Robust Real-Time-Constrained Estimation of Respiratory Motion for Interventional MRI on Mobile Organs. IEEE Trans. Inf. Technol. Biomed. 16(3): 365-374 (2012)

Krämer, P., Benois-Pineau, J. : Camera Motion Detection in the Rough Indexing Paradigm. TRECVID 2005

Bouthemy,P., Mayela, B., Acosta,T., Delyon, B.: Robust selection of parametric motion models in image sequences. ICIP 2016: 3743-3747

Alfarano, A., Maiano, L., Papa, L., & Amerini, I. (2024). Estimating optical flow: A comprehensive review of the state of the art. *Computer Vision and Image Understanding*, 104160.

Wang, Y., Wang, W., Li, Y., Guo, J., Xu, Y., Ma, J., ... & Jia, Y. (2024). Research on traditional and deep learning strategies based on optical flow estimation-a review. *Journal of King Saud University-Computer and Information Sciences*, 102029.

Pan, Z., Geng, D., & Owens, A. (2024). Self-supervised motion magnification by backpropagating through optical flow. *Advances in Neural Information Processing Systems*, *36*.

Hur, J., & Roth, S. (2020). Optical flow estimation in the deep learning age. *Modelling human motion: from human perception to robot design*, 119-140.