Motion Extraction

Computer Vision

Motion is a basic cue

Motion can be the only cue for segmentation

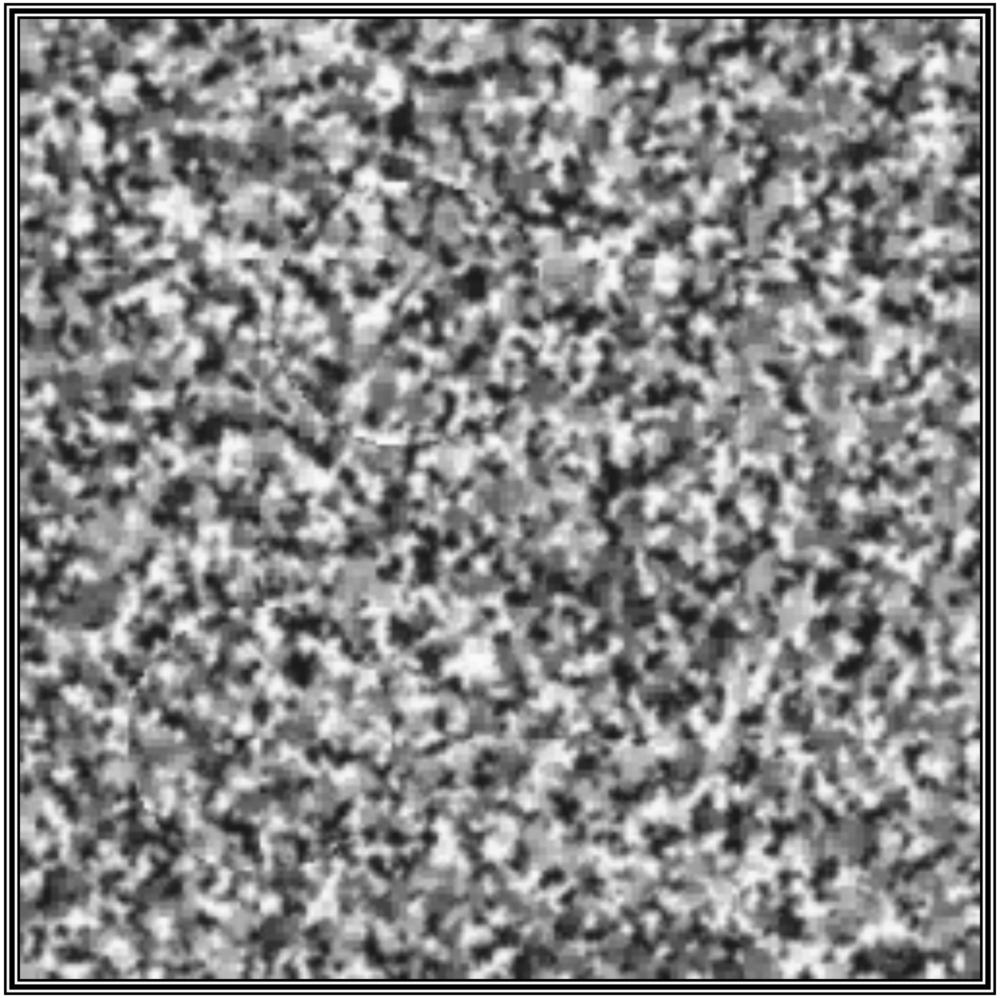
Biologically favoured because of camouflage

Motion is a basic cue

… which set in motion a constant, evolutionary race

Motion is a basic cue

Motion can be the only cue for segmentation



Computer Vision

Motion is a basic cue

Even impoverished motion data can elicit a strong percept

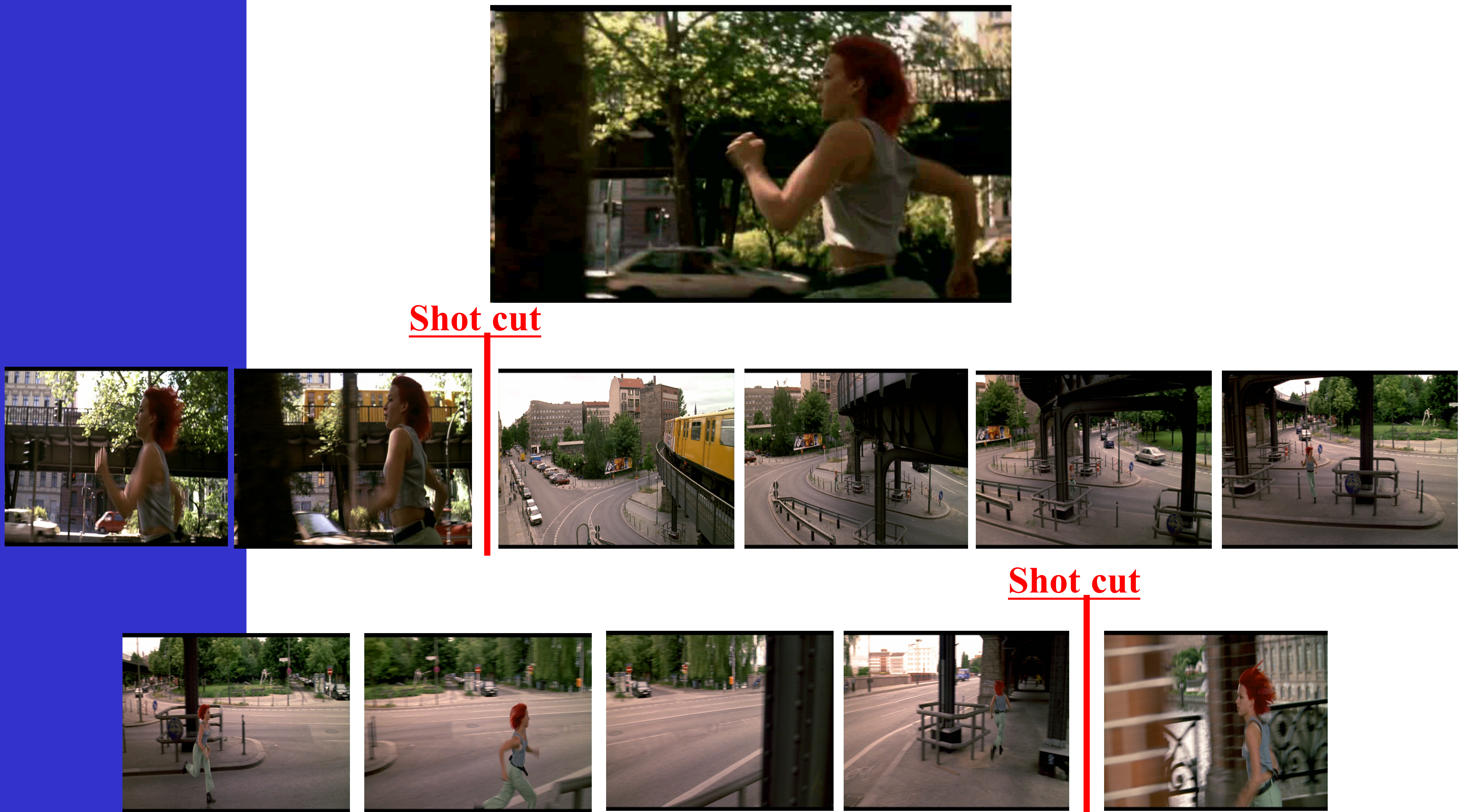
http://www.biomotionlab.ca/Demos/BMLwalker.html

Computer Vision

Some applications of motion extraction

- Change / shot cut detection
- Surveillance / traffic monitoring
- Autonomous driving
- Analyzing game dynamics in sports
- Motion capture / gesture analysis (HCI)
- Image stabilisation
- Motion compensation (e.g. medical robotics)
- Feature tracking for 3D reconstruction
- **Etc. !**

Shot cut detection & Keyframes



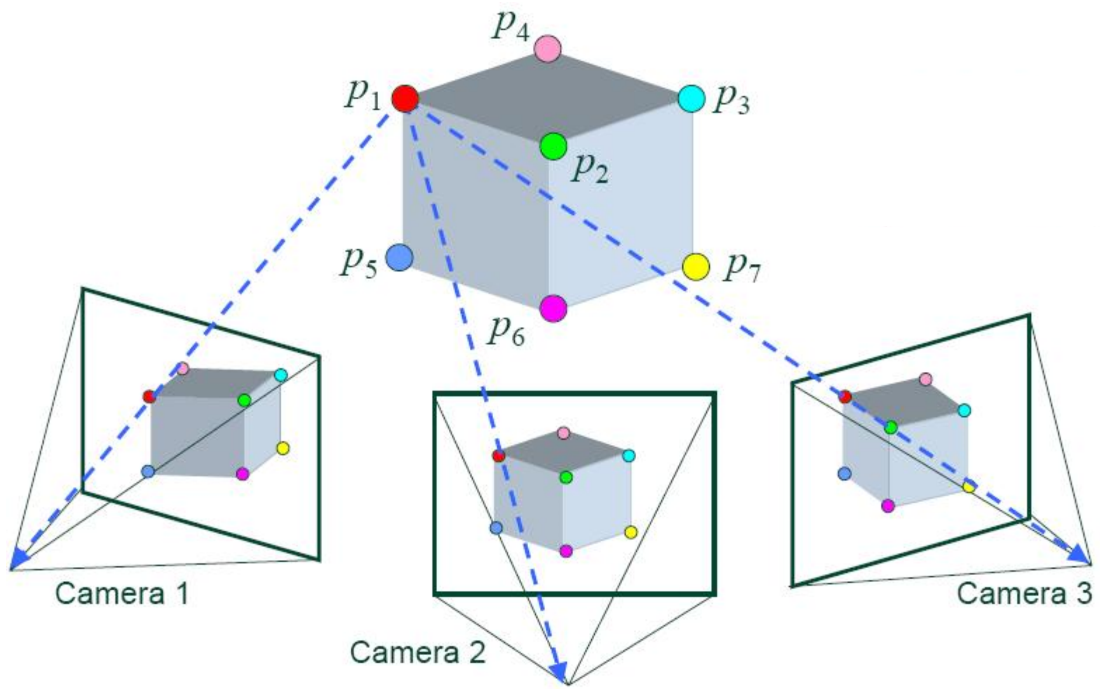
Computer Vision

Human-Machine Interfacing

KINECT for XBOX 360

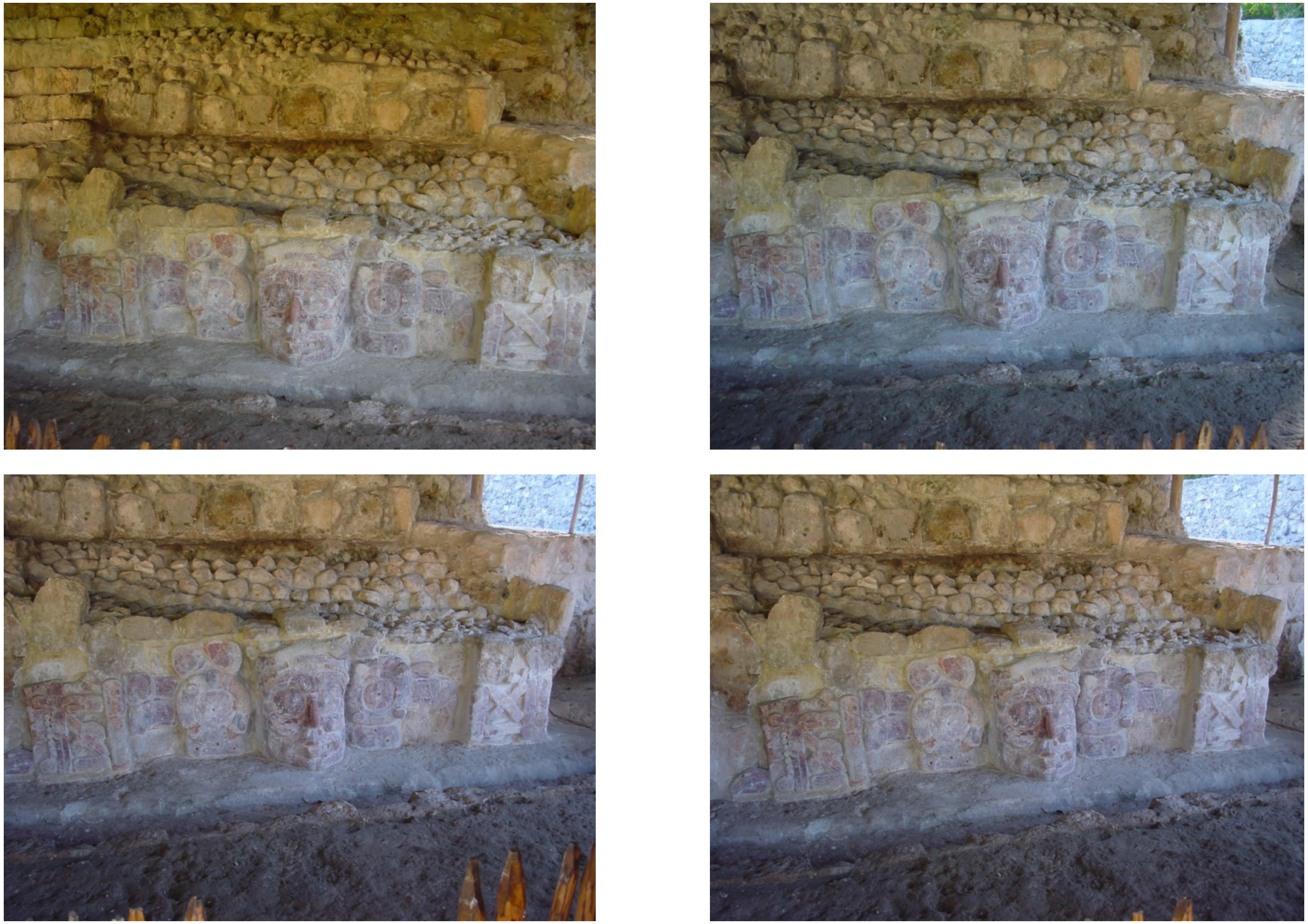
3D: Structure-from-Motion

Tracking points yields correspondences

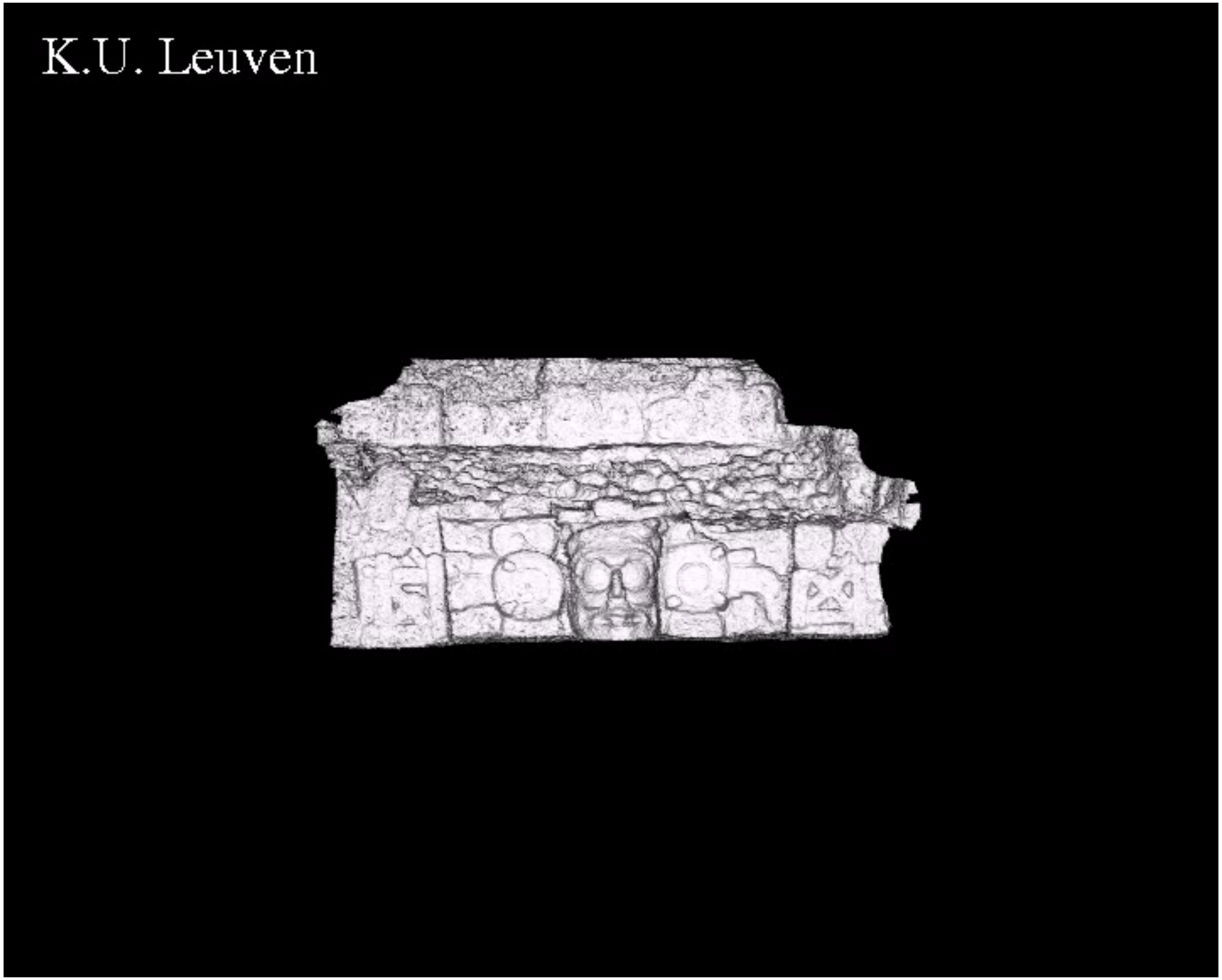


3D: Structure-from-Motion

Temple of the Masks, Edzna, Mexico



www.arc3d.b



in this lecture...

Several techniques, but…
this lecture is restricted to the

the detection of the “optical flow”

Computer Vision

Definition of optical flow

OPTICAL FLOW = apparent motion of brightness patterns

Ideally, the optical flow is the projection of the three-dimensional motion vectors on the image

Such 2D motion vector is sought at every pixel of the image (note: a motion vector here is a 2D translation vector)

Caution required !

Two examples where following brightness patterns is misleading:

1. Untextured, rotating sphere

$$\Downarrow$$

$$O.F. = 0$$

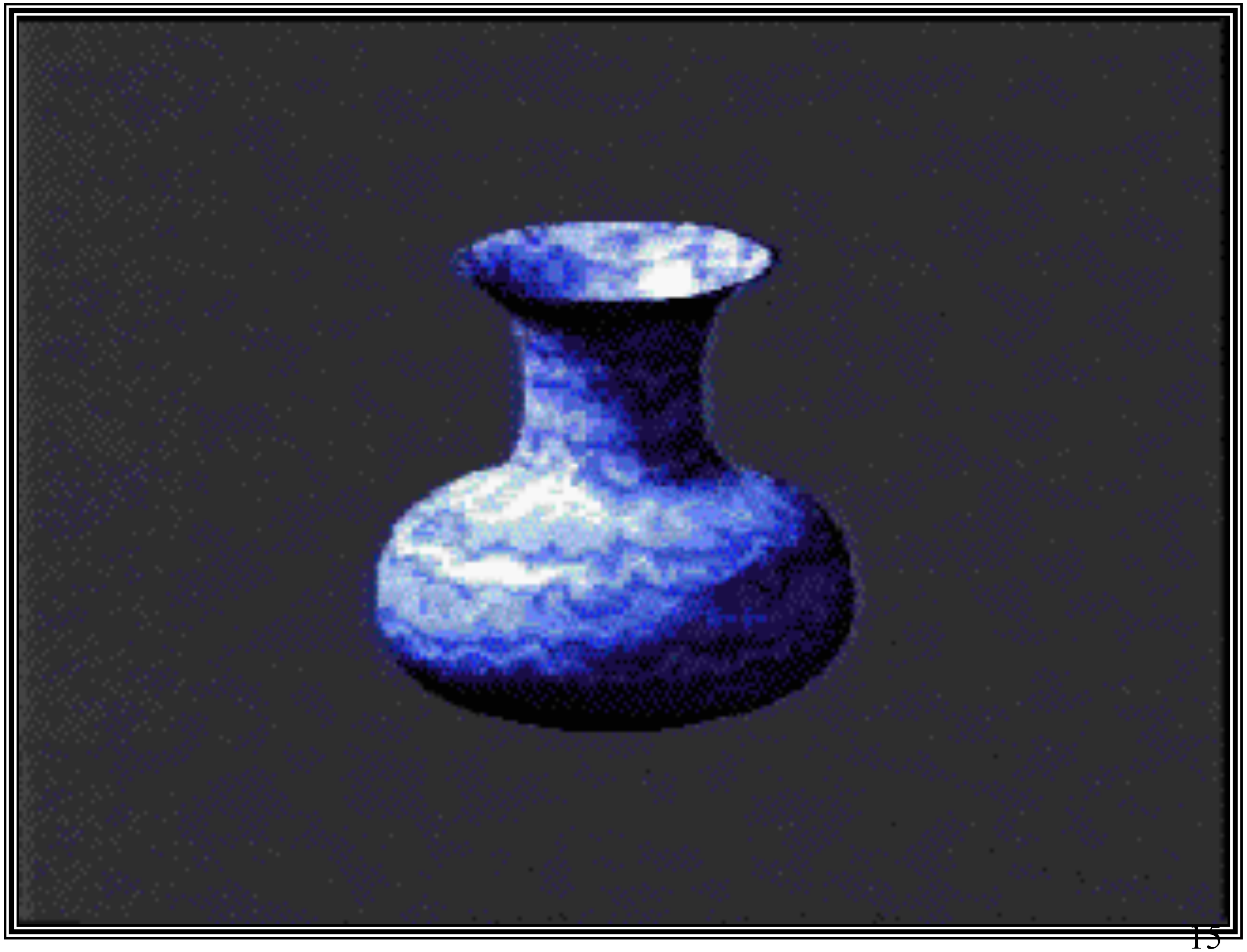
2. No motion, but changing lighting

$$\Downarrow$$

$$O.F. \neq 0$$

Computer Vision

Caution required !



Qualitative formulation

Suppose a *point of the scene* projects to a certain pixel of the current video frame. Our task is to figure out to which pixel in the next frame it moves…

That question needs answering *for all pixels* of the current image.

In order to find these corresponding pixels, we need to come up with a reasonable assumption on how we can detect them among the many.

We assume these corresponding pixels have the *same intensities* as the pixels the scene points came from in the previous frame.

That will only hold approximately…

Mathematical formulation

Our mathematical representation of a video:

$I\ (x,y,t)$ = brightness at (x,y) at time t

Optical flow constraint equation :

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

This equation states that if one were to track the image projections of a scene point through the video, it would not change its intensity. This tends to be true over short lapses of time.

Mathematical formulation

Our mathematical representation of a video:

$I\,(x,y,t)$ = brightness at (x,y) at time t

Optical flow constraint equation :

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

Note the different types of time derivatives !

Mathematical formulation

Our mathematical representation of a video:

$I\ (x,y,t)$ = brightness at (x,y) at time t

Optical flow constraint equation :

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

Change of intensity when following a physical point through the images

Change of intensity when looking at the same pixel (x,y) through the images

Mathematical formulation

We will use as shorthand notation for

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

$$u = \frac{dx}{dt}, \qquad v = \frac{dy}{dt}$$

$$I_x = \frac{\partial I}{\partial x}, \quad I_y = \frac{\partial I}{\partial y}, \qquad I_t = \frac{\partial I}{\partial t}$$

$$I_x u + I_y v + I_t = 0$$

1 equation per pixel

The aperture problem

$$u = \frac{dx}{dt}, \qquad v = \frac{dy}{dt}$$

$$I_x = \frac{\partial I}{\partial x}, \qquad I_y = \frac{\partial I}{\partial y}, \qquad I_t = \frac{\partial I}{\partial t}$$

$$I_x u + I_y v + I_t = 0$$

Note that we can measure the 3 derivatives of I, but that u and v are unknown

1 equation in 2 unknowns… the `aperture problem'

The aperture problem

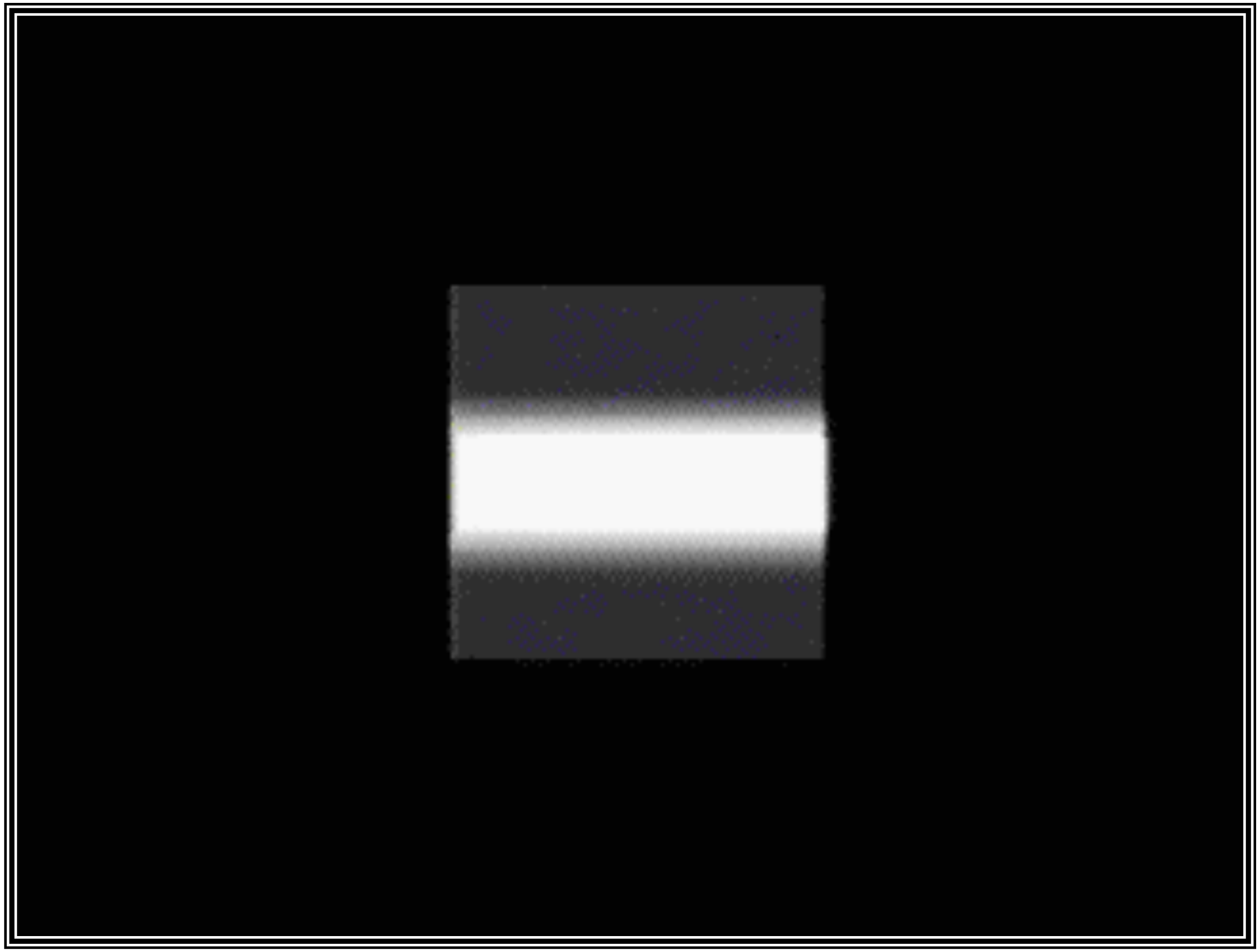
$$I_x u + I_y v + I_t = 0 \Rightarrow \left(I_x, I_y\right) \cdot \left(u, v\right) = -I_t$$

Aperture problem : only the component along the gradient can be retrieved

$$\frac{I_t}{\sqrt{I_x^2 + I_y^2}}$$

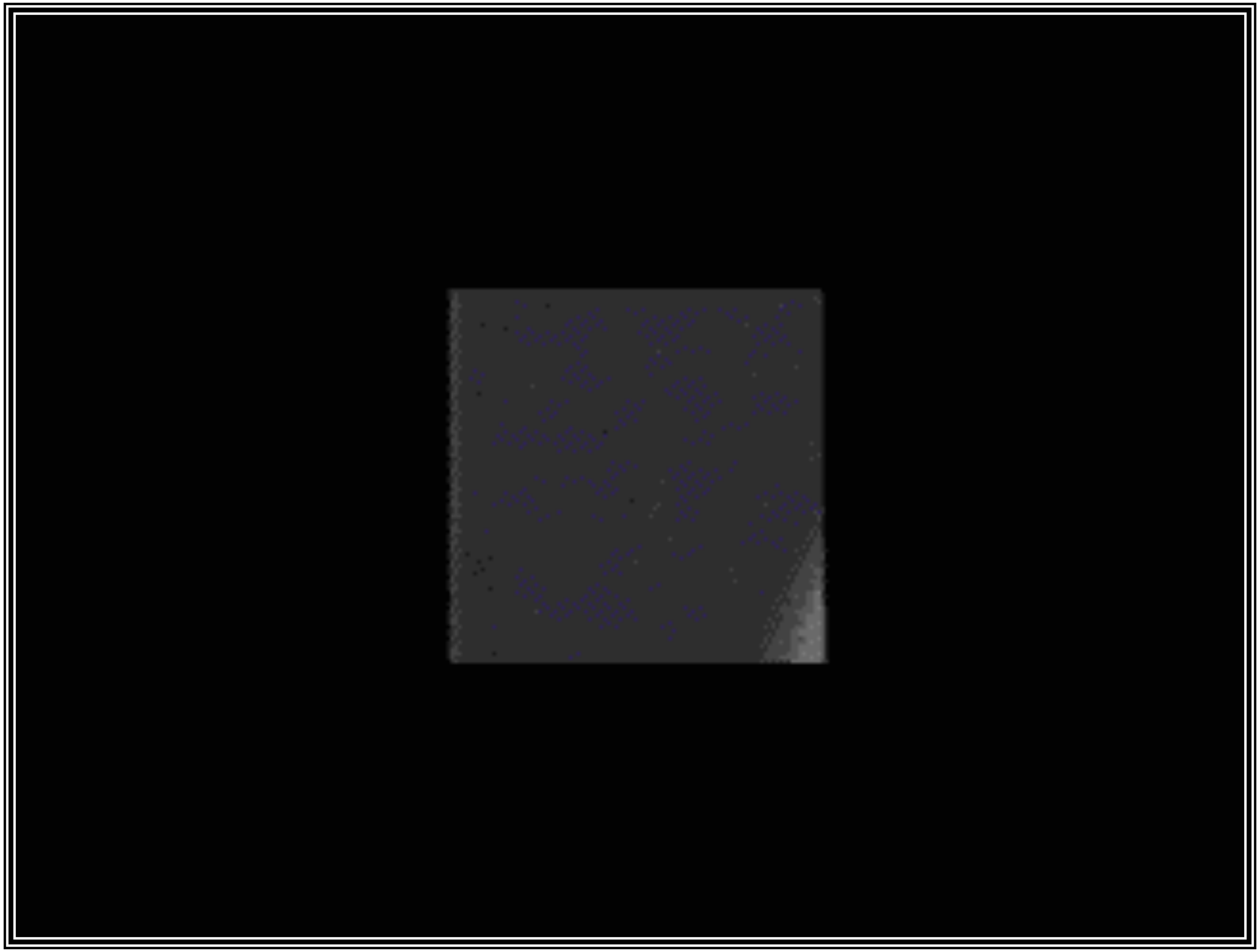
Computer Vision

The aperture problem



Computer Vision

Remarks



Remarks

1. The underdetermined nature could be solved using higher derivatives of intensity

2. for some intensity patterns, e.g. patches with a planar intensity profile, the aperture problem cannot be resolved anyway.

For many images, large parts have planar intensity profiles… higher-order derivatives than 1st order are typically not used (also because they are noisy)

Horn & Schunck algorithm

Breaking the spell via an …
additional smoothness constraint :

$$e_s = \iint ((u_x^2 + u_y^2) + (v_x^2 + v_y^2))dxdy,$$

to be minimized,
besides the OF constraint equation term

$$e_c = \iint (I_x u + I_y v + I_t)^2 dxdy,$$

The integrals are over the image.

Horn & Schunck algorithm

Breaking the spell via an …
additional smoothness constraint :

$$e_s = \iint ((u_x^2 + u_y^2) + (v_x^2 + v_y^2))dxdy,$$

to be minimized,
besides the OF constraint equation term

$$e_c = \iint (I_x u + I_y v + I_t)^2 dxdy,$$

minimize $e_s + \lambda e_c$ (also reduces influence of noise)

The calculus of variations

look for functions that extremize *functionals*

(a functional is a function that takes a vector as its input argument, and returns a scalar)

like for our functional:

$$\iint ((u_x^2 + u_y^2) + (v_x^2 + v_y^2))dxdy$$
$$+ \lambda \iint (I_x u + I_y v + I_t)^2 dxdy$$

what are the optimal *u(x,y)* and *v(x,y)* ?

The calculus of variations

look for functions that extremize *functionals*

$$I = \int_{x_1}^{x_2} F(x, f, f')\,dx \qquad \text{with } f = f(x)\ ,\ f' = \frac{df}{dx}$$

$$f(x_1) = f_1 \quad \text{and} \quad f(x_2) = f_2$$

Calculus of variations

Suppose

1. $f(x)$ is a solution

2. $\eta(x)$ is a test function with $\eta(x_1) = 0$ and $\eta(x_2) = 0$

We then consider

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

Rationale: suppose f is the solution, then any deviation should result in a worse I; when applying classical optimization over the values of ε the optimum should be $\varepsilon = 0$

Computer Vision

Calculus of variations

Suppose

1. $f(x)$ is a solution

2. $\eta(x)$ is a test function with $\eta(x_1) = 0$ and $\eta(x_2) = 0$

We then consider

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

With this trick, we reformulate an optimization over a function into a classical optimization over a scalar… a problem we know how to solve

Calculus of variations

Suppose

1. $f(x)$ is a solution

2. $\eta(x)$ is a test function with $\eta(x_1)= 0$ and $\eta(x_2) = 0$

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

for the optimum :

$$\frac{dI}{d\varepsilon}\Big|_{\varepsilon=0} = 0$$

Around the optimum, the derivative should be zero

Computer Vision

Calculus of variations

Suppose

1. $f(x)$ is a solution

2. $\eta(x)$ is a test function with $\eta(x_1) = 0$ and $\eta(x_2) = 0$

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

for the optimum :

$$\int_{x_1}^{x_2} (\eta(x) F_f + \eta'(x) F_{f'})\,dx = 0$$

$f + \varepsilon\eta$ with $\varepsilon = 0$ $\quad$ $f' + \varepsilon\eta'$ with $\varepsilon = 0$

Computer Vision

Calculus of variations

$$\int_{x_1}^{x_2} (\eta(x)F_f + \eta'(x)F_{f'})dx = 0$$

Using integration by parts:

$$\int_{x_1}^{x_2} \frac{d}{dx}(g\ h)\ dx = \int_{x_1}^{x_2} (\frac{dg}{dx}h + \frac{dh}{dx}g)dx = [gh]_{x_1}^{x_2}$$

where

$$[gh]_{x_1}^{x_2} = g(x_2)h(x_2) - g(x_1)h(x_1)$$

Computer Vision

Calculus of variations

$$\int_{x_1}^{x_2} \left(\eta(x)F_f + \eta'(x)F_{f'}\right)dx = 0$$

Using integration by parts $\int_{x_1}^{x_2} \frac{d}{dx}(\eta(x)F_{f'})\, dx$:

$$\int_{x_1}^{x_2} \eta'(x)F_{f'} + \eta(x)\frac{d}{dx}F_{f'}dx = \left[\eta(x)F_{f'}\right]_{x_1}^{x_2}$$

Computer Vision

Calculus of variations

$$\int_{x_1}^{x_2} \left(\eta(x)F_f + \eta'(x)F_{f'}\right)dx = 0$$

Using integration by parts $\int_{x_1}^{x_2} \frac{d}{dx}(\eta(x)F_{f'})\, dx$:

$$\int_{x_1}^{x_2} \eta'(x)F_{f'} + \eta(x)\frac{d}{dx}F_{f'}dx = \left[\eta(x)F_{f'}\right]_{x_1}^{x_2}$$

Computer Vision

Calculus of variations

$$\int_{x_1}^{x_2} (\eta(x)F_f + \eta'(x)F_{f'})dx = 0$$

Using integration by parts $\int_{x_1}^{x_2} \frac{d}{dx}(\eta(x)F_{f'})\, dx$:

$$\int_{x_1}^{x_2} \eta'(x)F_{f'}dx = \left[\eta(x)F_{f'}\right]_{x_1}^{x_2} - \int_{x_1}^{x_2} \eta(x)\frac{d}{dx}F_{f'}\, dx,$$

Computer Vision

Calculus of variations

$$\int_{x_1}^{x_2} (\eta(x)F_f + \eta'(x)F_{f'})dx = 0$$

Using integration by parts $\int_{x_1}^{x_2} \frac{d}{dx}(\eta(x)F_{f'})\,dx$:

$$\int_{x_1}^{x_2} \eta'(x)F_{f'}dx = \qquad - \int_{x_1}^{x_2} \eta(x)\frac{d}{dx}F_{f'}\,dx,$$

Calculus of variations

$$\int_{x_1}^{x_2} (\eta(x)F_f + \eta'(x)F_{f'})dx = 0$$

Using integration by parts $\int_{x_1}^{x_2} \frac{d}{dx}(\eta(x)F_{f'})\, dx$:

$$\int_{x_1}^{x_2} \eta'(x)F_{f'}dx = \qquad -\int_{x_1}^{x_2} \eta(x)\frac{d}{dx}F_{f'}\, dx,$$

Therefore

$$\int_{x_1}^{x_2} \eta(x)(F_f - \frac{d}{dx}F_{f'})dx = 0$$

regardless of $\eta(x)$, then $F_f - \frac{d}{dx}F_{f'} = 0$

Euler-Lagrange equation

Computer Vision

Calculus of variations

Generalizations

- 1. $$I = \int_{x_1}^{x_2} F(x, f_1, f_2, \ldots, f_1', f_2', \ldots) dx$$

  Simultaneous Euler-Lagrange equations, i.c. one for *u* and one for *v* :

  $$F_{fi} - \frac{d}{dx} F_{f_i'} = 0$$

Computer Vision

Calculus of variations

Generalizations

- 1. $I = \int_{x_1}^{x_2} F(x, f_1, f_2, \ldots, f_1', f_2', \ldots) dx$

Simultaneous Euler-Lagrange equations, i.c. one for *u* and one for *v* :

$$F_{fi} - \frac{d}{dx} F_{f_i'} = 0$$

We add $\varepsilon_1 \eta_1$ to f_1 , $\varepsilon_2 \eta_2$ to f_2, etc.
then repeat, once deriving w.r.t. ε_1,
once w.r.t. ε_2 , …
thus obtaining a system of PDEs

Computer Vision

Calculus of variations

Generalizations

- 1. $I = \int_{x_1}^{x_2} F(x, f_1, f_2, \ldots, f_1', f_2', \ldots) dx$

  Simultaneous Euler-Lagrange equations, i.c. one for *u* and one for *v* :

$$F_{fi} - \frac{d}{dx} F_{f_i'} = 0$$

- 2. 2 independent variables x and y

$$I = \iint_D F(x, y, f + \varepsilon\eta, f_x + \varepsilon\eta_x, f_y + \varepsilon\eta_y) dxdy$$

Calculus of variations

Hence

$$0=\iint_D (\eta F_f + \eta_x F_{f_x} + \eta_y F_{f_y})\,dxdy$$

Now by Gauss's integral theorem,

$$\iint_D \left(\frac{\partial Q}{\partial x} + \frac{\partial P}{\partial y} \right) dxdy = \int_{\partial D} (Q\,dy - P\,dx),$$

such that

$$\iint_D \frac{\partial(\eta F_{f_x})}{\partial x} + \frac{\partial(\eta F_{f_y})}{\partial y}\,dxdy = \int_{\partial D} (\eta\, F_{f_x} dy - \eta\, F_{f_y} dx)$$

= 0

Computer Vision

Calculus of variations

$$\iint_D \frac{\partial(\eta F_{f_x})}{\partial x} + \frac{\partial(\eta F_{f_y})}{\partial y}\, dxdy = 0$$

$$\iint_D (\eta_x F_{f_x} + \eta_y F_{f_y})\, dxdy + \iint_D (\eta \frac{\partial F_{f_x}}{\partial x} + \eta \frac{\partial F_{f_y}}{\partial y})\, dxdy = 0$$

Computer Vision

Calculus of variations

$$0 = \int\int_D (\eta F_f + \eta_x F_{f_x} + \eta_y F_{f_y}) dxdy$$

$$\int\int_D \frac{\partial(\eta F_{f_x})}{\partial x} + \frac{\partial(\eta F_{f_y})}{\partial y} dxdy = 0$$

$$\int\int_D (\eta_x F_{f_x} + \eta_y F_{f_y})\, dxdy = -\int\int_D \eta \left(\frac{\partial F_{f_x}}{\partial x} + \frac{\partial F_{f_y}}{\partial y} \right) dxdy$$

Consequently,

$$0 = \int\int_D \eta \left(F_f - \frac{\partial}{\partial x} F_{f_x} - \frac{\partial}{\partial y} F_{f_y} \right) dxdy$$

for all test functions η , thus

$$F_f - \frac{\partial}{\partial x} F_{f_x} - \frac{\partial}{\partial y} F_{f_y} = 0$$

is the Euler-Lagrange equation

Computer Vision

Horn & Schunck

The Euler-Lagrange equations :

$$F_u - \frac{\partial}{\partial x} F_{u_x} - \frac{\partial}{\partial y} F_{u_y} = 0$$

$$F_v - \frac{\partial}{\partial x} F_{v_x} - \frac{\partial}{\partial y} F_{v_y} = 0$$

In our case ,

$$F = (u_x^2 + u_y^2) + (v_x^2 + v_y^2) + \lambda (I_x u + I_y v + I_t)^2 ,$$

so the Euler-Lagrange equations are

$$\Delta u = \lambda (I_x u + I_y v + I_t) I_x ,$$

$$\Delta v = \lambda (I_x u + I_y v + I_t) I_y ,$$

$$\Delta = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$$ is the Laplacian operator

Horn & Schunck

Remarks :

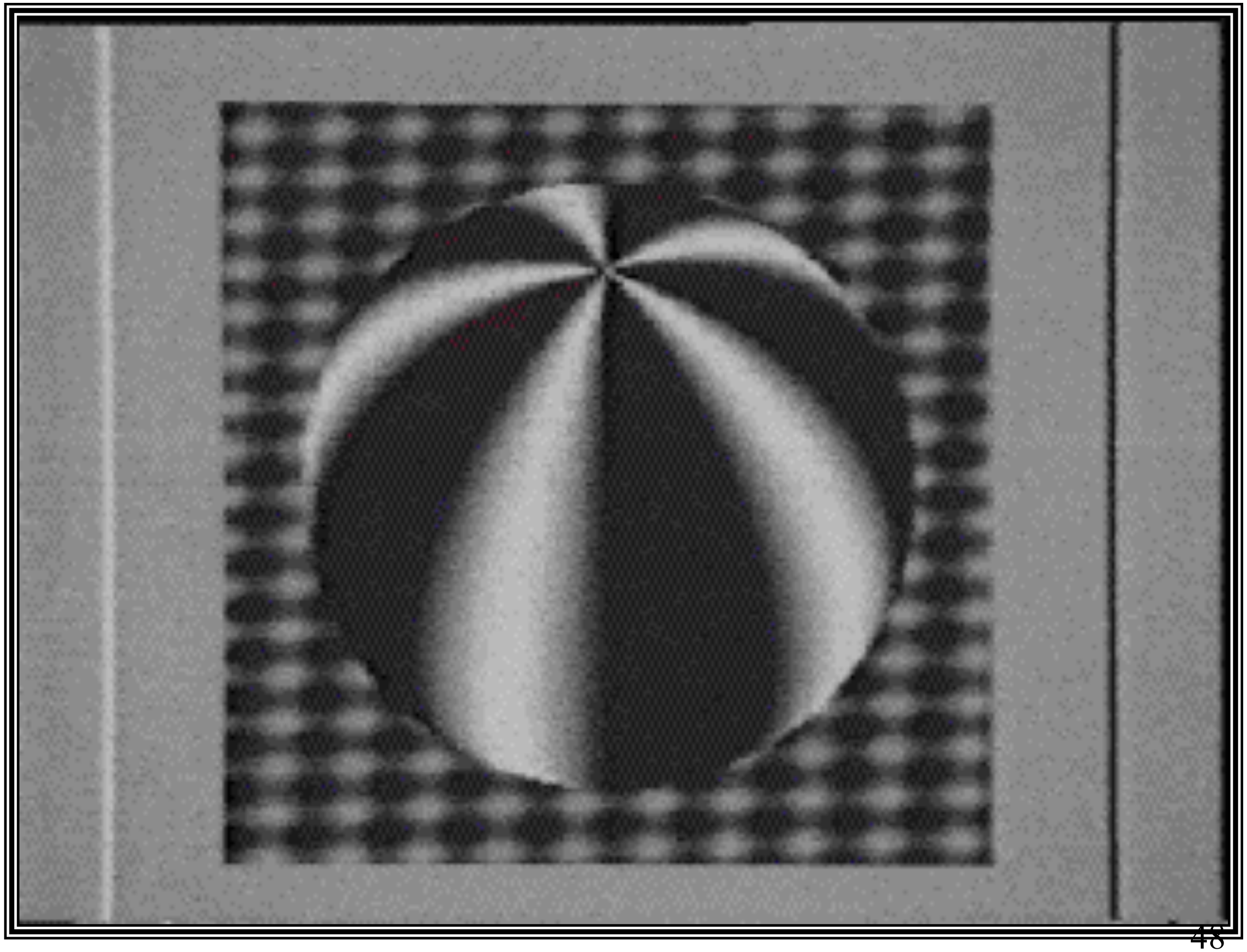
1. Coupled PDEs solved using iterative methods and finite differences (iteration i)

$$\frac{\partial u}{\partial i} = \Delta u - \lambda(I_x u + I_y v + I_t) I_x ,$$

$$\frac{\partial v}{\partial i} = \Delta v - \lambda(I_x u + I_y v + I_t) I_y ,$$

2. More than two frames allow for a better estimation of I_t

3. Information spreads from edge- and corner-type patterns



Horn & Schunck, remarks

1. Errors at object boundaries

   (where the smoothness constraint is no longer valid)

2. Example of *regularisation*

   (selection principle for the solution of ill-posed problems by imposing an extra generic constraint, like here smoothness)

Other approaches

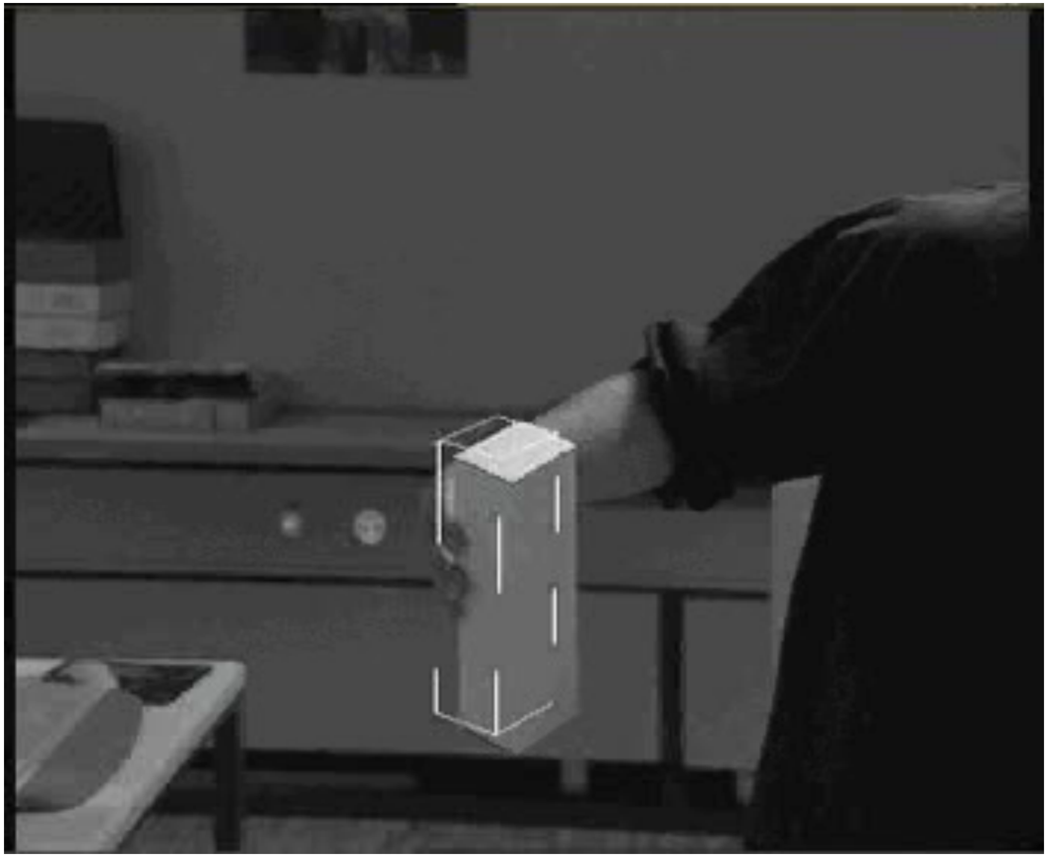
1. Model-based tracking (application-specific)
   - active contours
   - analysis/synthesis schemes

2. Feature tracking (more generic)
   - corner tracking
   - blob/contour tracking
   - intensity profile tracking
   - region tracking

Condensation tracker

Computer Vision

Model-based tracker



(EPFL)

Computer Vision

Model-based tracker

(EPFL)

Computer Vision

Motion capture for special effects