Optical Flow I

Guido Gerig

CS 6320, Spring 2012

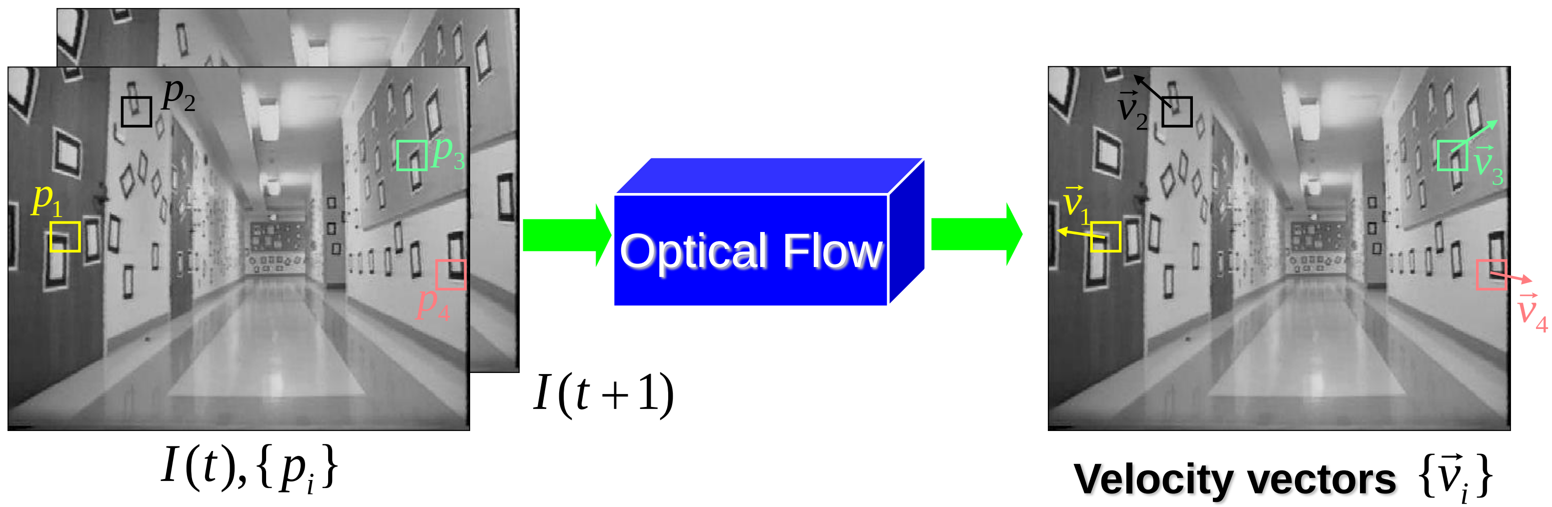
(credits: Marc Pollefeys UNC Chapel Hill, Comp 256 / K.H. Shafique, UCSF, CAP5415 / S. Narasimhan, CMU / Bahadir K. Gunturk, EE 7730 / Bradski&Thrun, Stanford CS223

Materials

- Gary Bradski & Sebastian Thrun, Stanford CS223 http://robots.stanford.edu/cs223b/index.html
- S. Narasimhan, CMU: http://www.cs.cmu.edu/afs/cs/academic/class/15385-s06/lectures/ppts/lec-16.ppt
- M. Pollefeys, ETH Zurich/UNC Chapel Hill: http://www.cs.unc.edu/Research/vision/comp256/vision10.ppt
- K.H. Shafique, UCSF: http://www.cs.ucf.edu/courses/cap6411/cap5415/
  - Lecture 18 (March 25, 2003), Slides: PDF/ PPT
- Jepson, Toronto: http://www.cs.toronto.edu/pub/jepson/teaching/vision/2503/opticalFlow.pdf
- Original paper Horn&Schunck 1981: http://www.csd.uwo.ca/faculty/beau/CS9645/PAPERS/Horn-Schunck.pdf
- MIT AI Memo Horn& Schunck 1980: http://people.csail.mit.edu/bkph/AIM/AIM-572.pdf
- Bahadir K. Gunturk, EE 7730 Image Analysis II
- Some slides and illustrations from L. Van Gool, T. Darell, B. Horn, Y. Weiss, P. Anandan, M. Black, K. Toyama

Tracking – Rigid Objects

What is Optical Flow (OF)?



Optical flow is the relation of the motion field:

- *the 2D projection of the physical movement of points relative to the observer* to 2D displacement of pixel patches on the image plane.

Common assumption:

The appearance of the image patches do not change (brightness constancy)

$$I(p_i, t) = I(p_i + \vec{v}_i, t+1)$$

Note: more elaborate tracking models can be adopted if more frames are process all at once

Optical Flow

- Brightness Constancy
- The Aperture problem
- Regularization
- Lucas-Kanade
- Coarse-to-fine
- Parametric motion models
- Direct depth
- SSD tracking
- Robust flow
- Bayesian flow

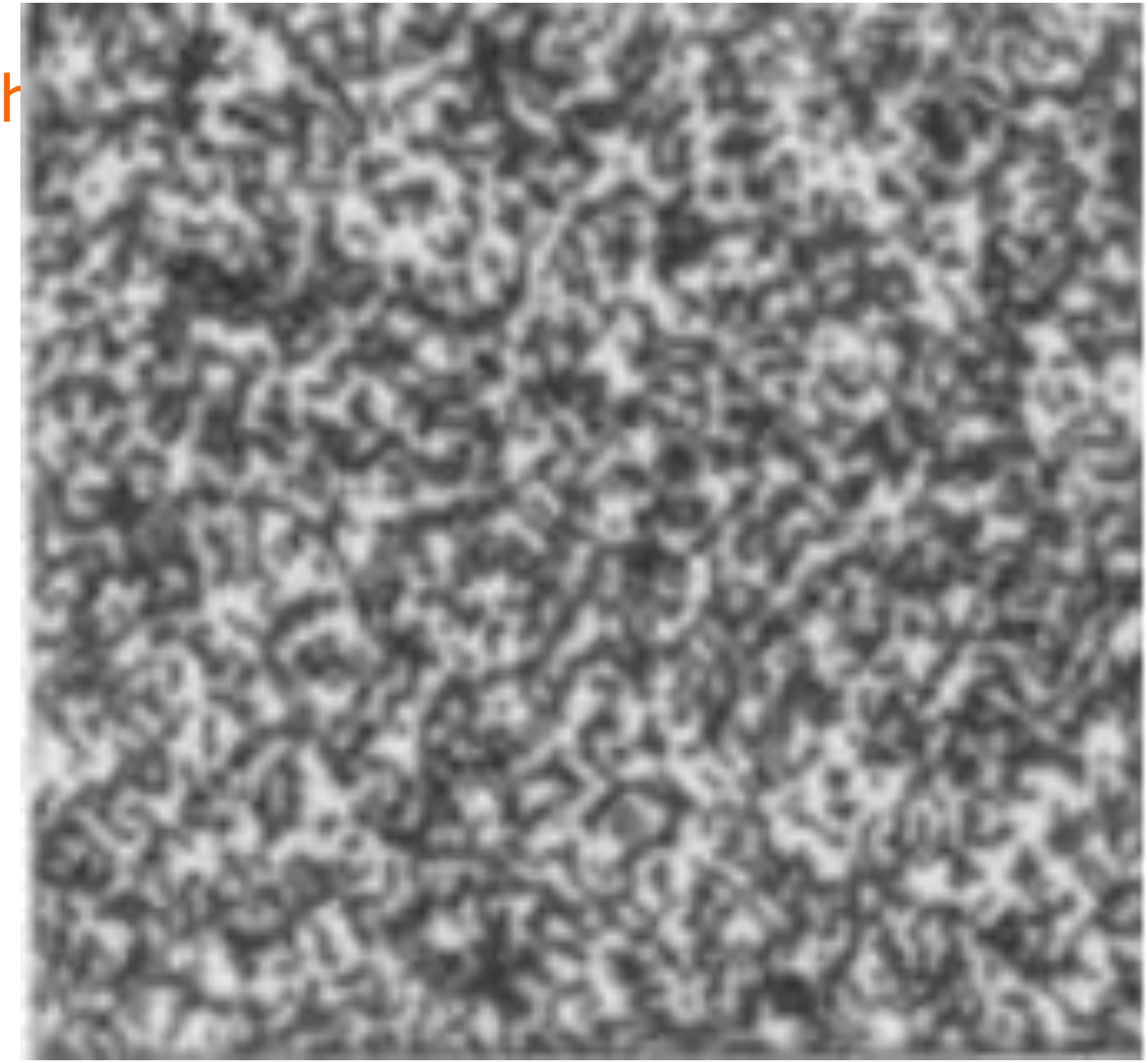
Optical Flow and Motion

- We are interested in finding the movement of scene objects from time-varying images (videos).
- Lots of uses
  - Motion detection
  - Track objects
  - Correct for camera jitter (stabilization)
  - Align images (mosaics)
  - 3D shape reconstruction
  - Special effects
  - Games: http://www.youtube.com/watch?v=JlLkkom6tWw
  - User Interfaces: http://www.youtube.com/watch?v=Q3gT52sHDI4
  - Video compression

Optical Flow: Where do pixels move to?

Related to: Optical flow

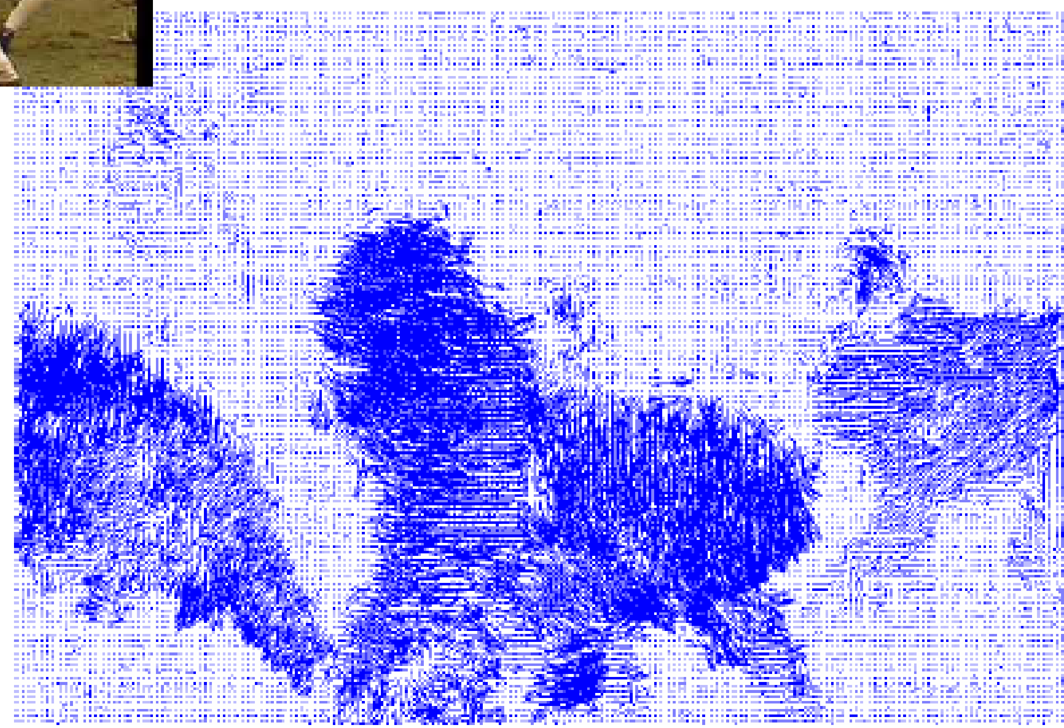
Wh



Tracking – Non-rigid Objects

(Comaniciu et al, Siemens)

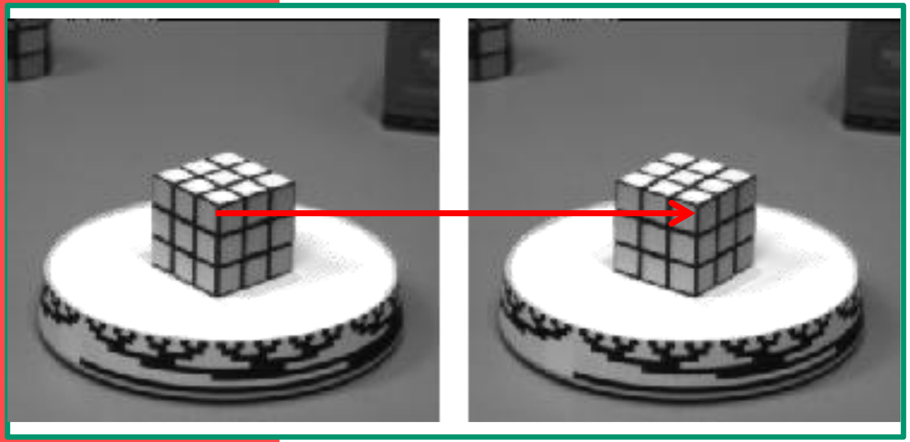
Tracking – Non-rigid Objects

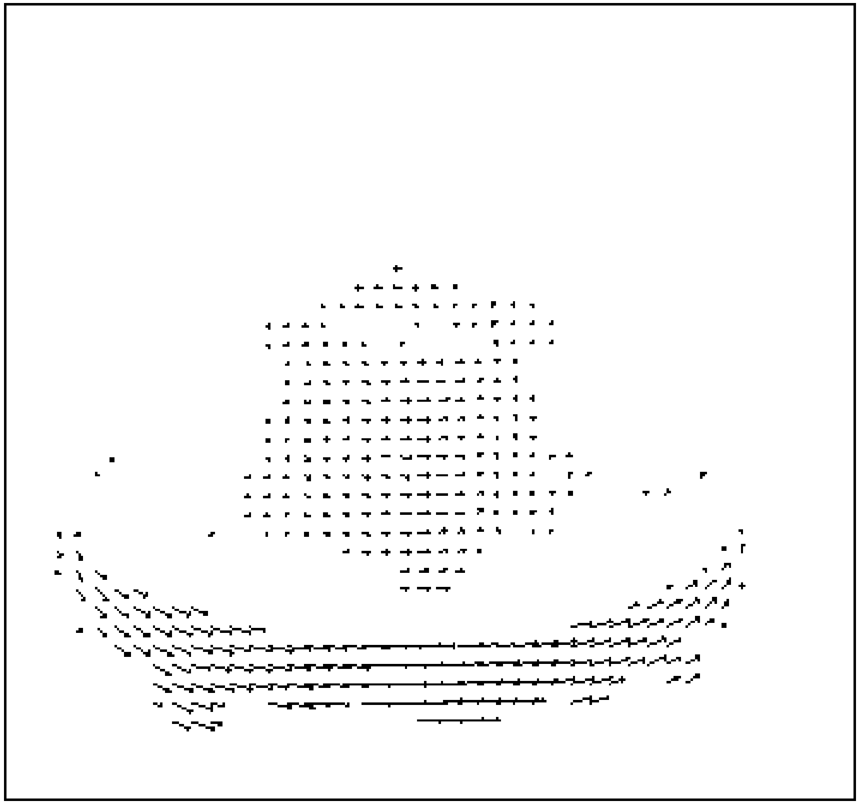


Alper Yilmaz, Fall 2005 UCF

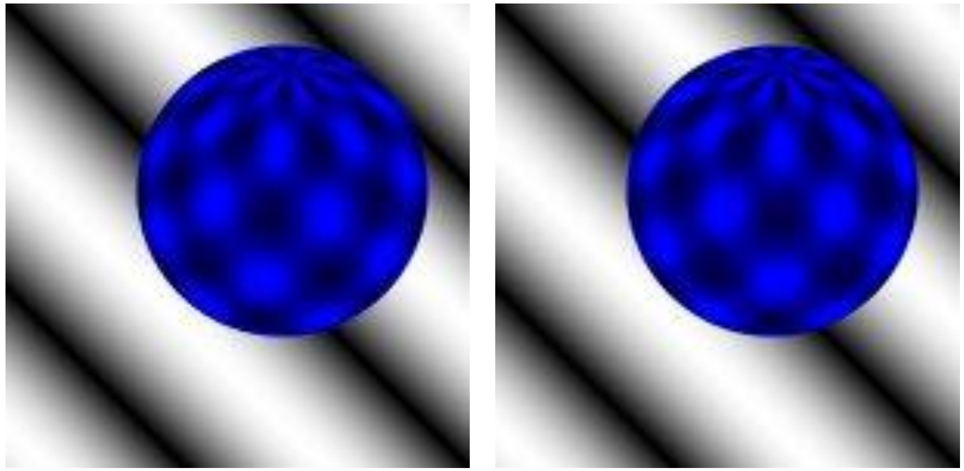
Optical Flow: Correspondence

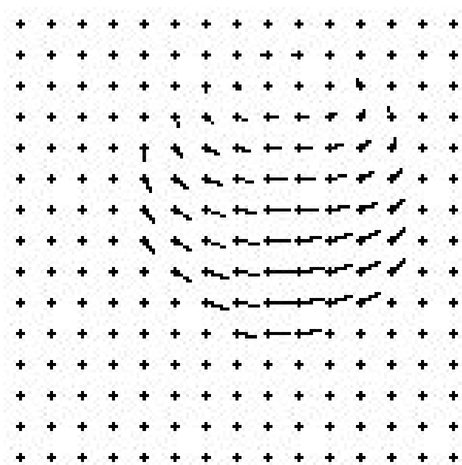
Basic question: Which Pixel went where?





Optical Flow is NOT 3D motion field

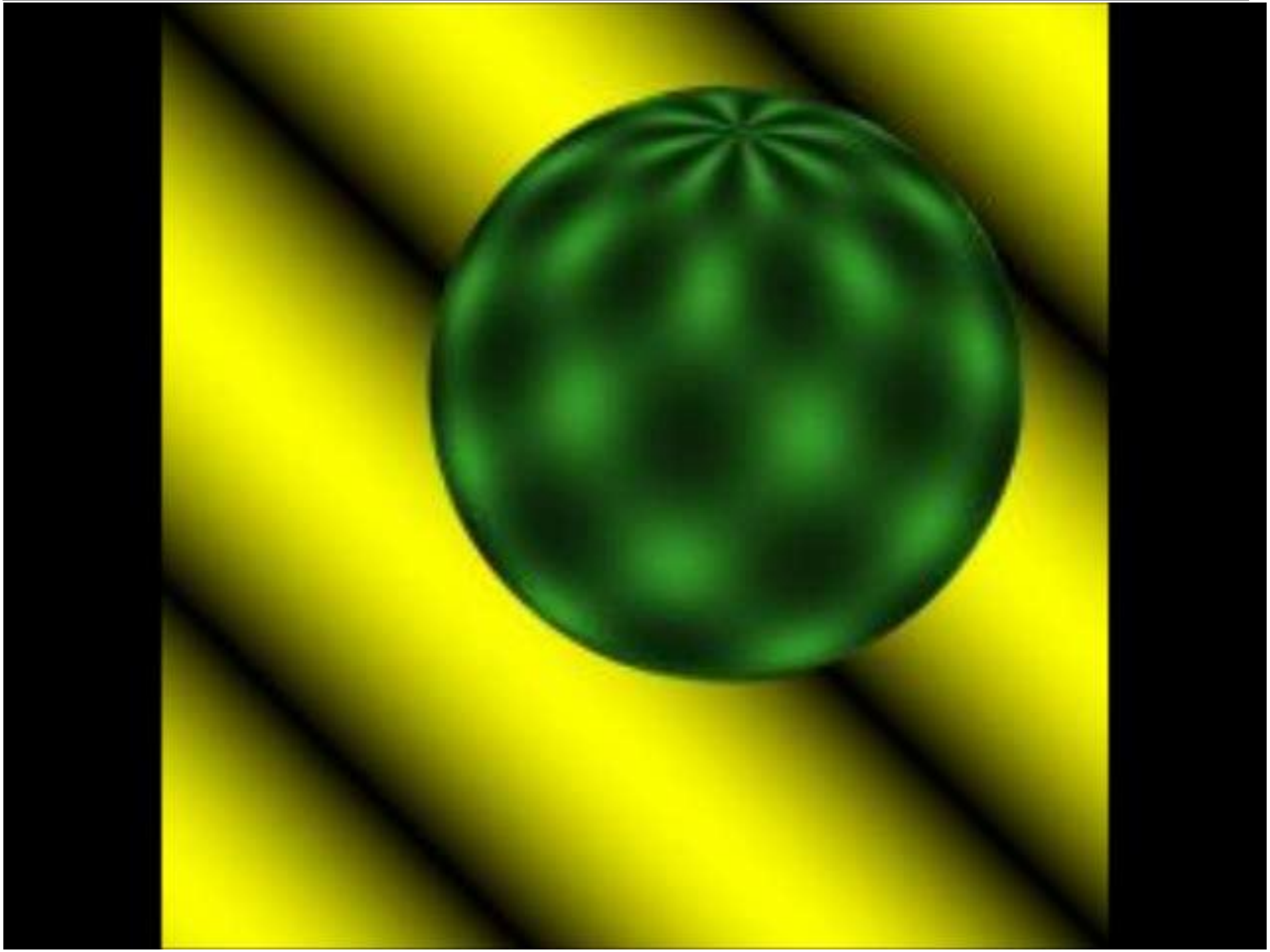




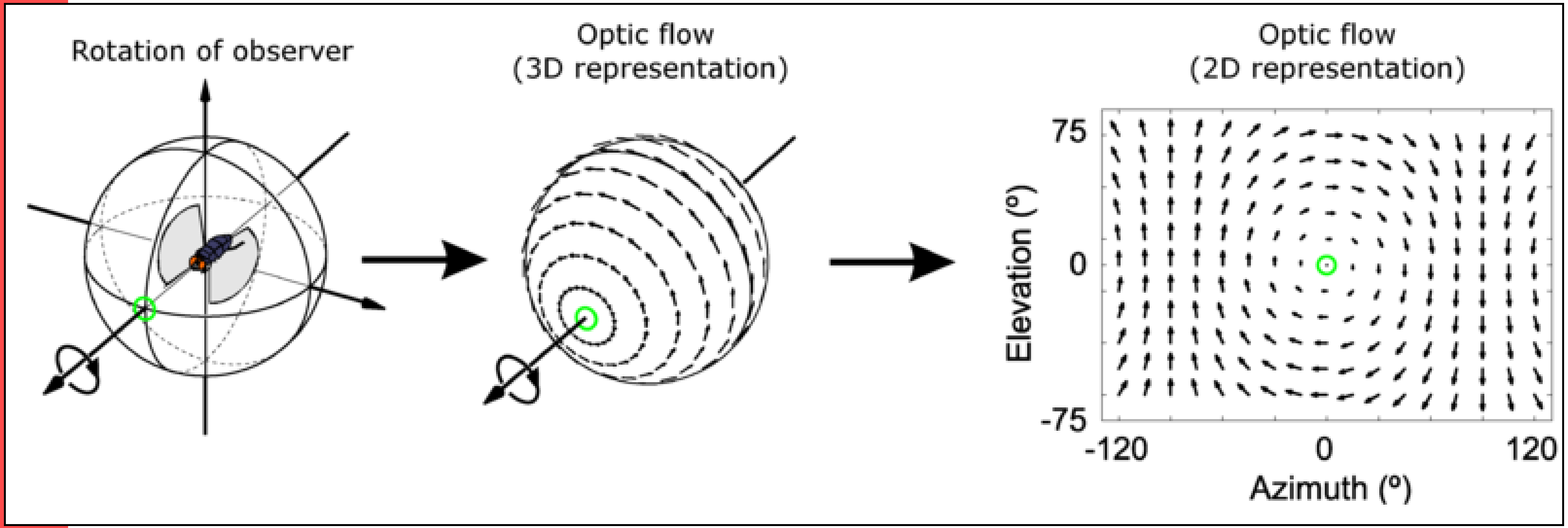
Optical flow: Pixel motion field as observed in image.

http://of-eval.sourceforge.net/

Structure from Motion?



Optical Flow is NOT 3D motion field

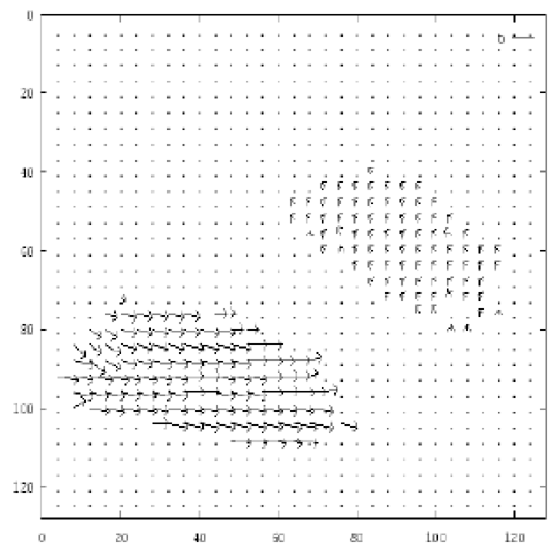


http://en.wikipedia.org/wiki/File:Opticfloweg.png

Definition of optical flow

OPTICAL FLOW = apparent motion of brightness patterns

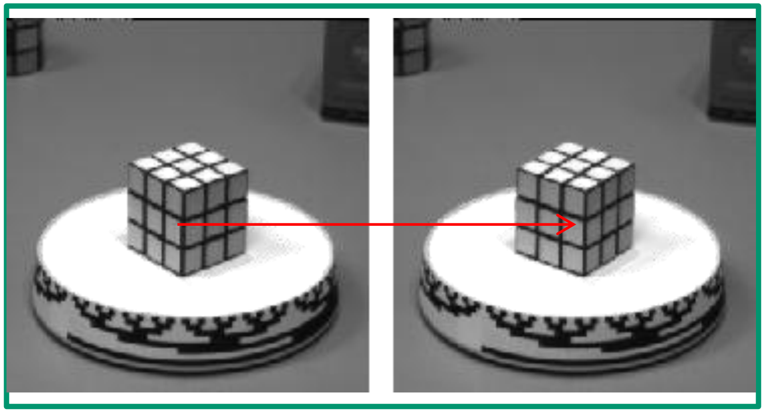
Ideally, the optical flow is the projection of the three-dimensional velocity vectors on the image



Optical Flow

- Brightness Constancy
- The Aperture problem
- Regularization
- Lucas-Kanade
- Coarse-to-fine
- Parametric motion models
- Direct depth
- SSD tracking
- Robust flow
- Bayesian flow

Start with an Equation: Brightness Constancy



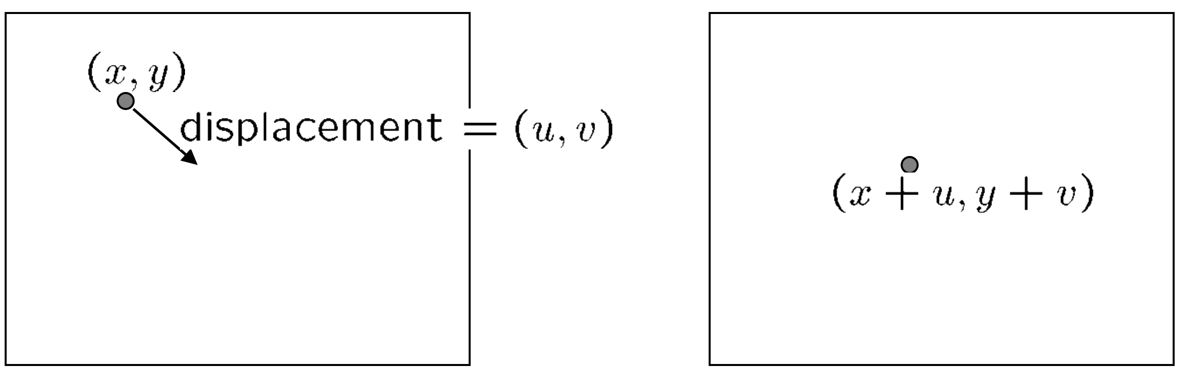
Time: t

Time: t + dt

Point moves (small), but its brightness remains constant:

$$I_{t1}(x, y) = I_{t2}(x + u, y + v)$$

$$I = constant \rightarrow \frac{dI}{dt} = 0$$



I_1 I_2

Mathematical formulation

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

Brightness constancy assumption (shift of location but brightness stays same):

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

Optical flow constraint equation (chain rule):

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

The aperture problem

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

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

$$\boxed{I_x u + I_y v + I_t = 0}$$

Horn and Schunck optical flow equation

1 equation in 2 unknowns

Optical Flow: 1D Case

Brightness Constancy Assumption:

$$f(t) \equiv I(\underbrace{x(t),t}) = I(x(t+dt),t+dt)$$

$$\frac{\partial f(x)}{\partial t} = 0$$ Because no change in brightness with time

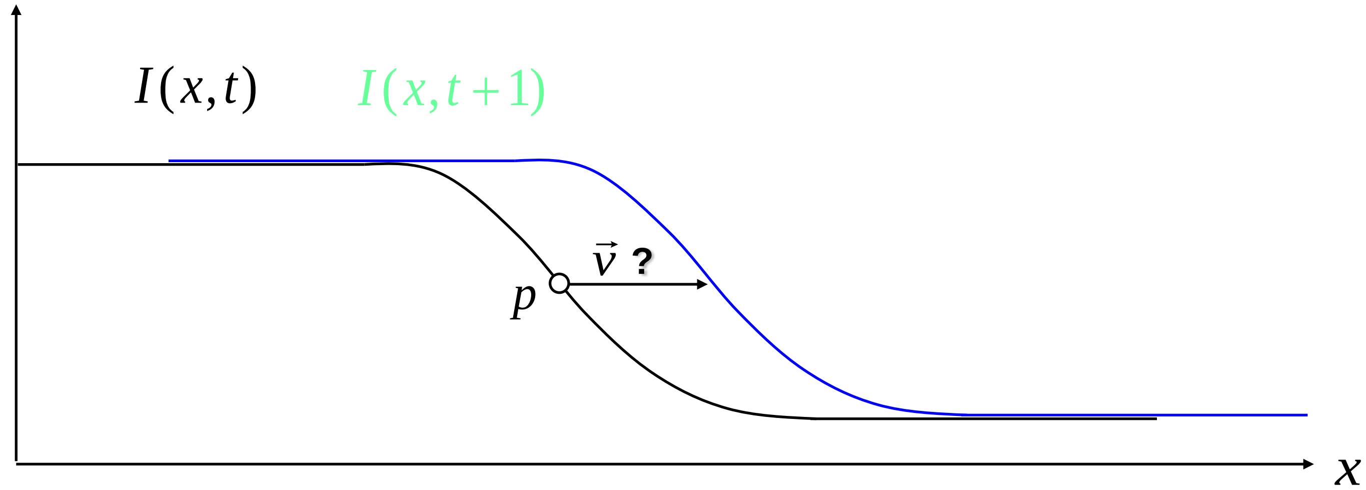
$$\left.\frac{\partial I}{\partial x}\right|_t \left(\frac{\partial x}{\partial t}\right) + \left.\frac{\partial I}{\partial t}\right|_{x(t)} = 0$$

I_x $\quad v \quad$ I_t

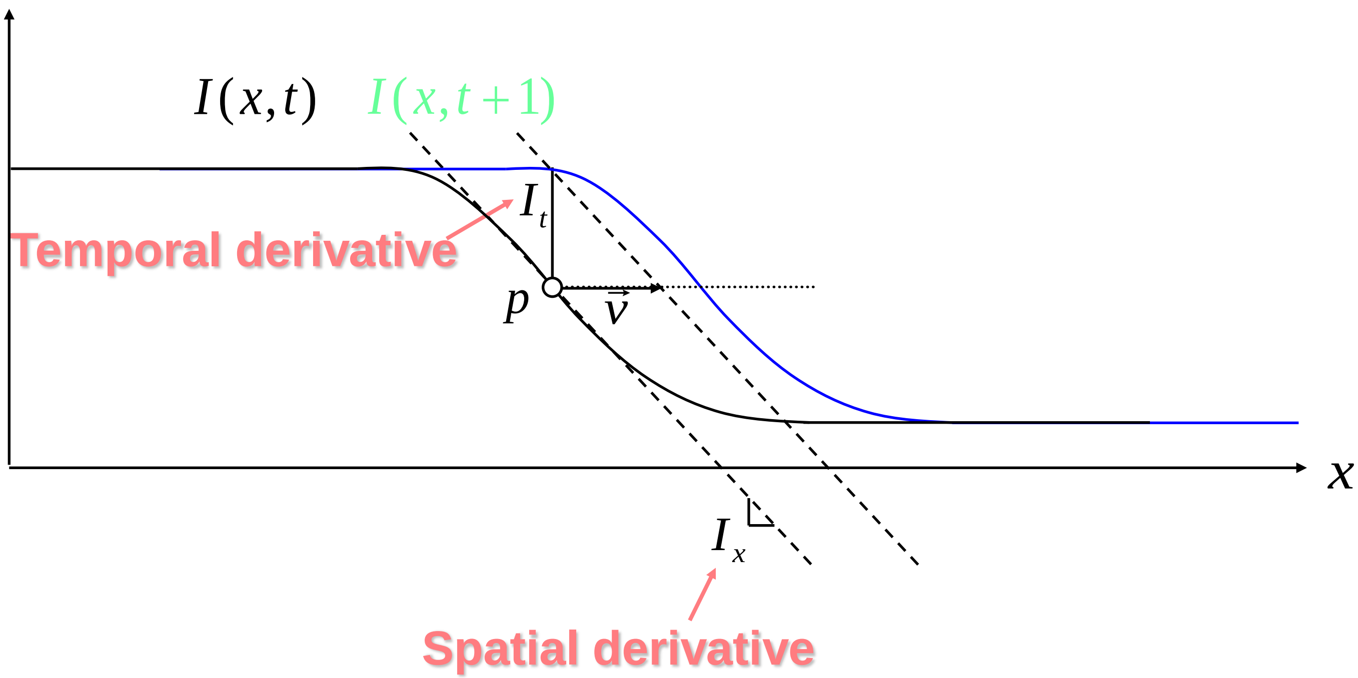
$$\Rightarrow v = -\frac{I_t}{I_x}$$

Gary Bradski & Sebastian Thrun, Stanford CS223 http://robots.stanford.edu/cs223b/index.html

Tracking in the 1D case:



Tracking in the 1D case:



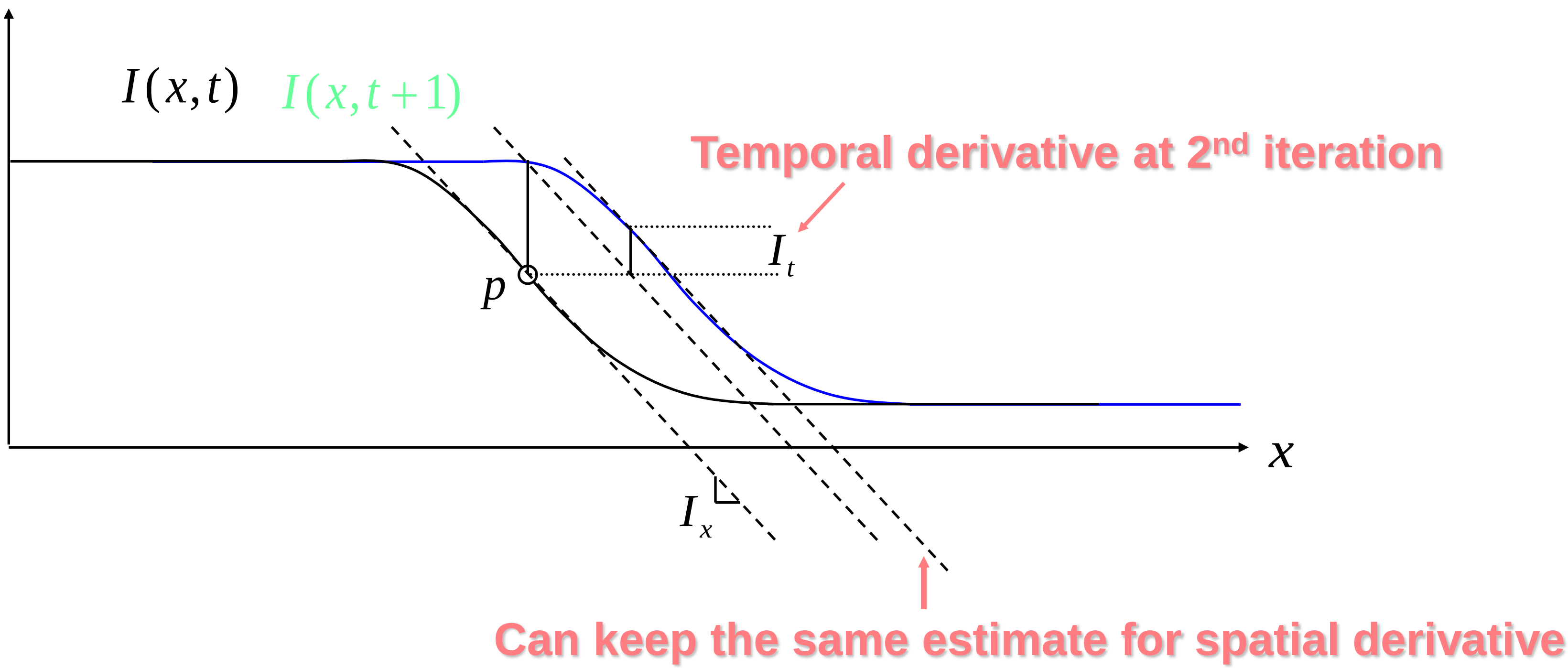
$$I_x = \left.\frac{\partial I}{\partial x}\right|_t \qquad I_t = \left.\frac{\partial I}{\partial t}\right|_{x=p} \quad \Rightarrow \quad \vec{v} \approx -\frac{I_t}{I_x}$$

Assumptions:

- **Brightness constancy**
- **Small motion**

Tracking in the 1D case:

Iterating helps refining the velocity vector



$$\vec{v} \leftarrow \vec{v}_{previous} - \frac{I_t}{I_x}$$

Converges in about 5 iterations

From 1D to 2D tracking

1D: $$\left.\frac{\partial I}{\partial x}\right|_{t}\left(\frac{\partial x}{\partial t}\right)+\left.\frac{\partial I}{\partial t}\right|_{x(t)}=0$$

2D: $$\left.\frac{\partial I}{\partial x}\right|_{t}\left(\frac{\partial x}{\partial t}\right)+\left.\frac{\partial I}{\partial y}\right|_{t}\left(\frac{\partial y}{\partial t}\right)+\left.\frac{\partial I}{\partial t}\right|_{x(t)}=0$$

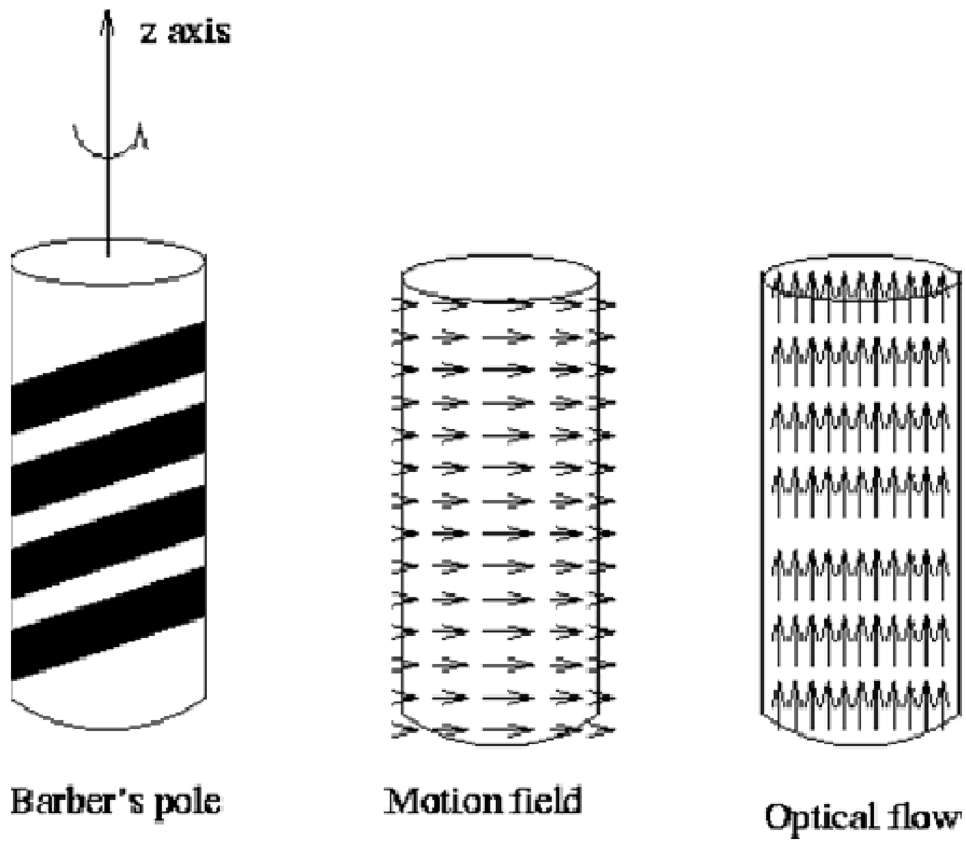
$$\left.\frac{\partial I}{\partial x}\right|_{t}u+\left.\frac{\partial I}{\partial y}\right|_{t}v+\left.\frac{\partial I}{\partial t}\right|_{x(t)}=0$$

Shoot! One equation, two velocity *(u,v)* unknowns…

Optical Flow vs. Motion: Aperture Problem

Barber shop pole:
http://www.youtube.com/watch?v=VmqQs613SbE

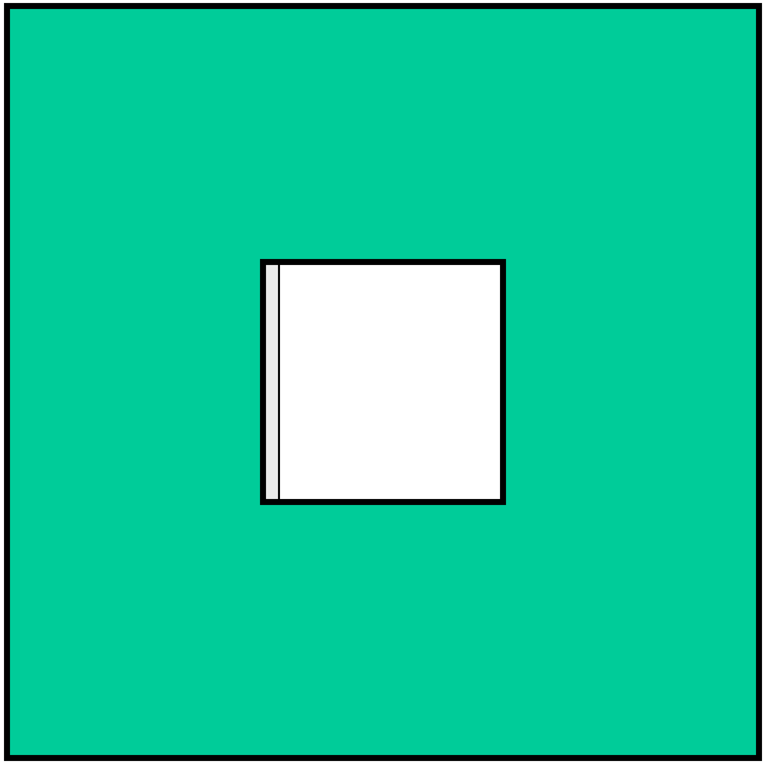
Barber pole illusion



Optical Flow

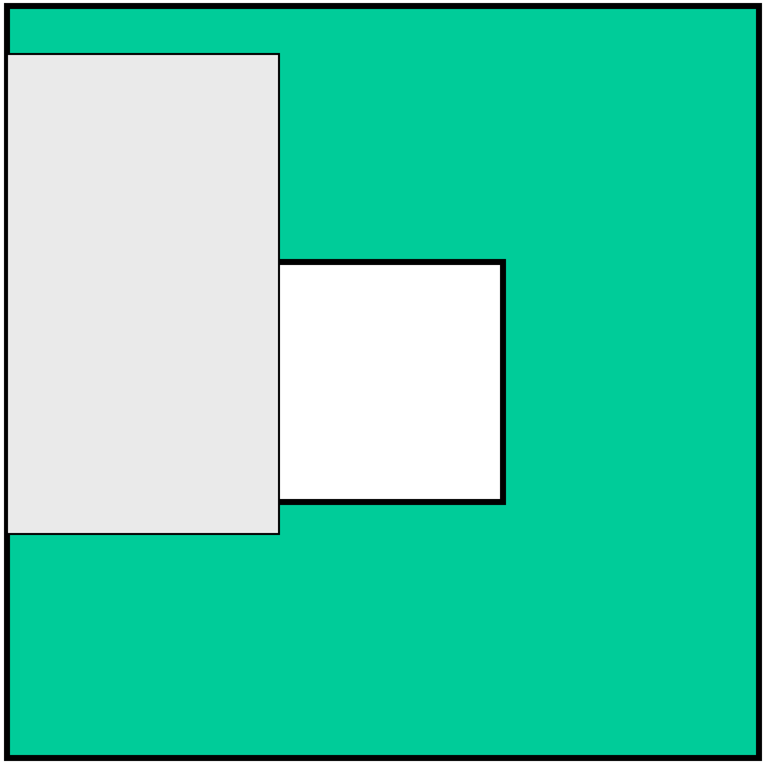
- Brightness Constancy
- The Aperture problem
- Regularization
- Lucas-Kanade
- Coarse-to-fine
- Parametric motion models
- Direct depth
- SSD tracking
- Robust flow
- Bayesian flow

How does this show up visually? Known as the “Aperture Problem”



Gary Bradski & Sebastian Thrun, Stanford CS223
http://robots.stanford.edu/cs223b/index.html

Aperture Problem Exposed



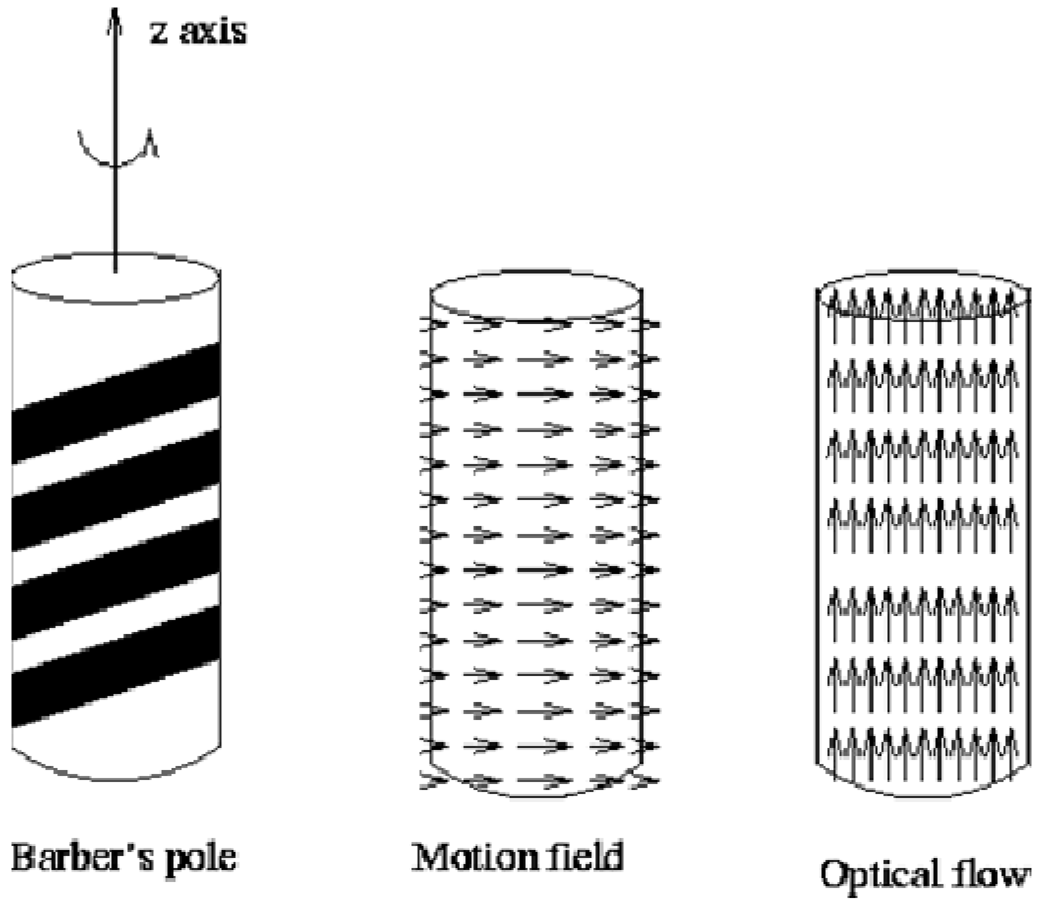
Motion along just an edge is ambiguous

Gary Bradski & Sebastian Thrun, Stanford CS223
http://robots.stanford.edu/cs223b/index.html

Aperture Problem in Real Life

Aperture Problem

Barber pole illusion



Normal Flow

Notation

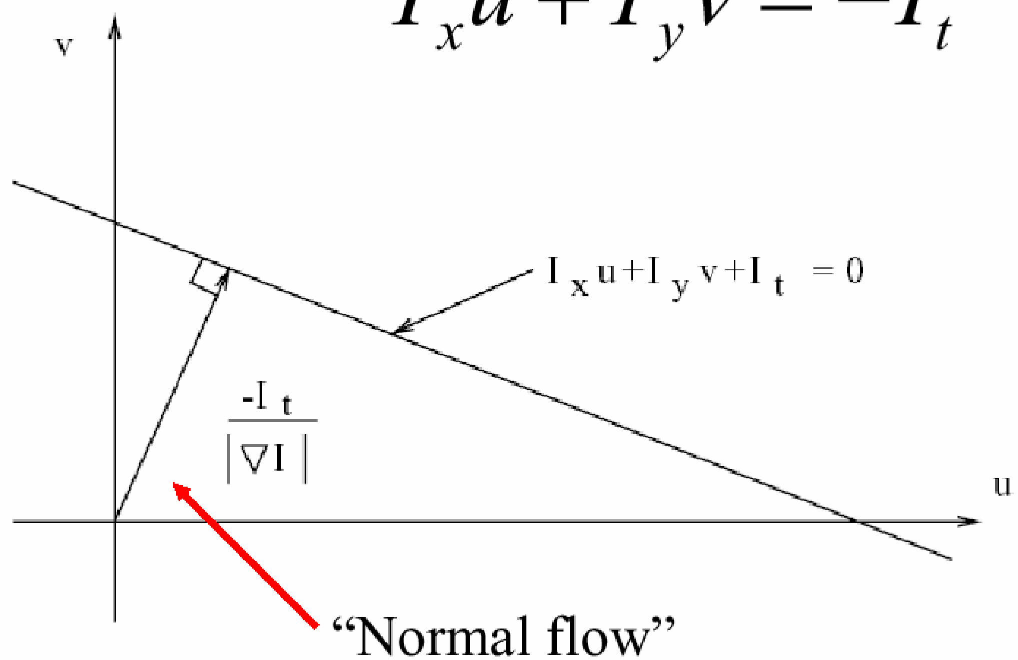
$$I_x u + I_y v + I_t = 0$$

$$\nabla I^T \mathbf{u} = -I_t$$

$$\mathbf{u} = \begin{bmatrix} u \\ v \end{bmatrix} \quad \nabla I = \begin{bmatrix} I_x \\ I_y \end{bmatrix}$$

At a single image pixel, we get a line:

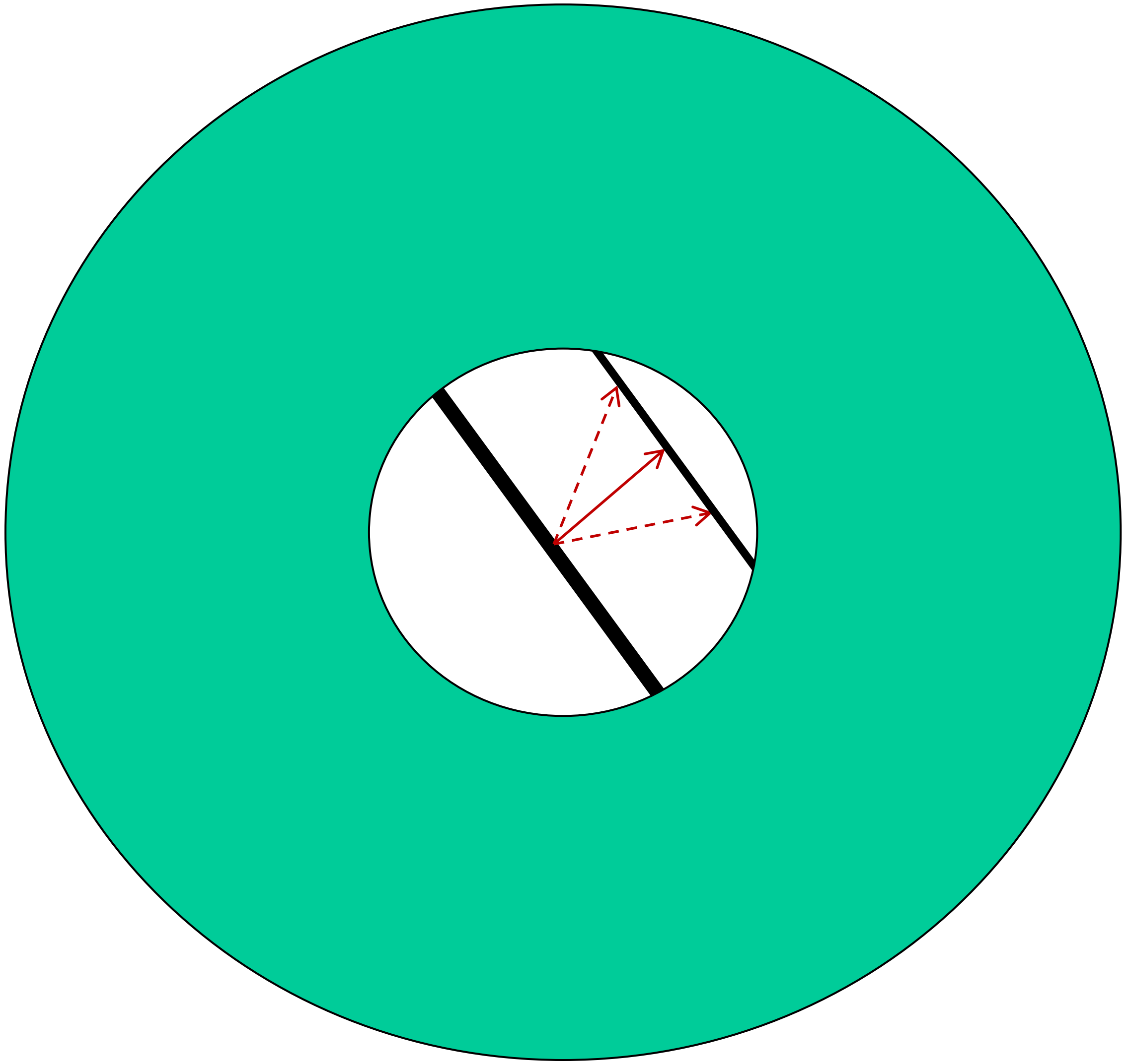
$$I_x u + I_y v = -I_t$$



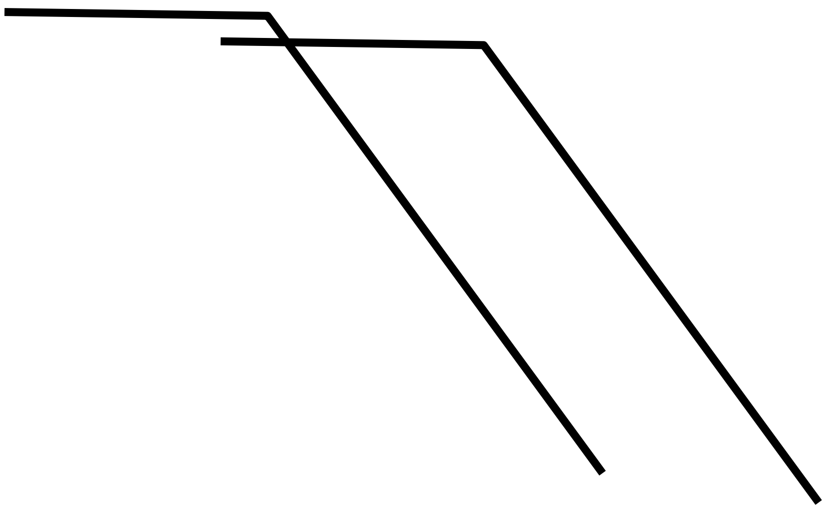
We get at most "Normal Flow" – with one point we can only detect movement perpendicular to the brightness gradient. Solution is to take a patch of pixels Around the pixel of interest.

* Slide from Michael Black, CS143 2003

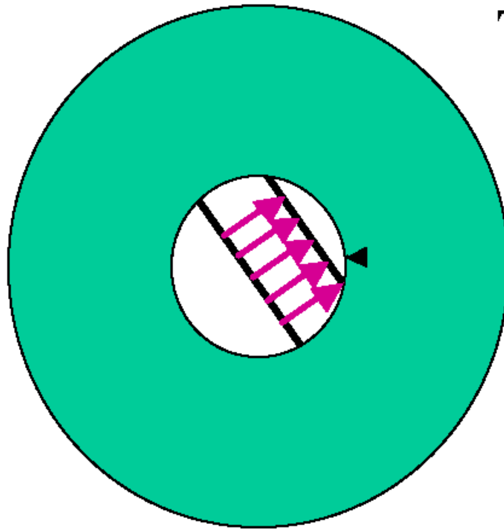
Aperture Problem



Aperture Problem



Aperture Problem and Normal Flow

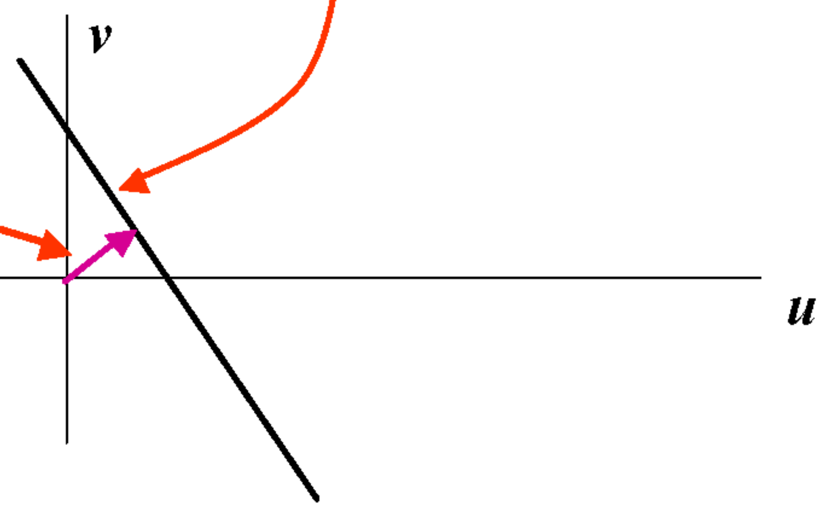


The gradient constraint:

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

$$\nabla I \bullet \vec{U} = 0$$

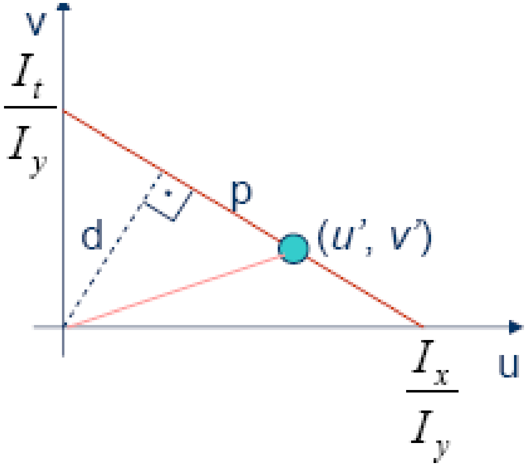
Defines a line in the *(u,v)* space



Normal Flow:

$$u_{\perp} = -\frac{I_t}{|\nabla I|}\frac{\nabla I}{|\nabla I|}$$

Aperture Problem and Normal Flow



$$v = u\frac{I_x}{I_y} + \frac{I_t}{I_y}$$

- Let (u', v') be true flow
- True flow has two components
  - Normal flow: *d*
  - Parallel flow: *p*
- Normal flow **can be computed**
- Parallel flow **cannot**

Computing True Flow

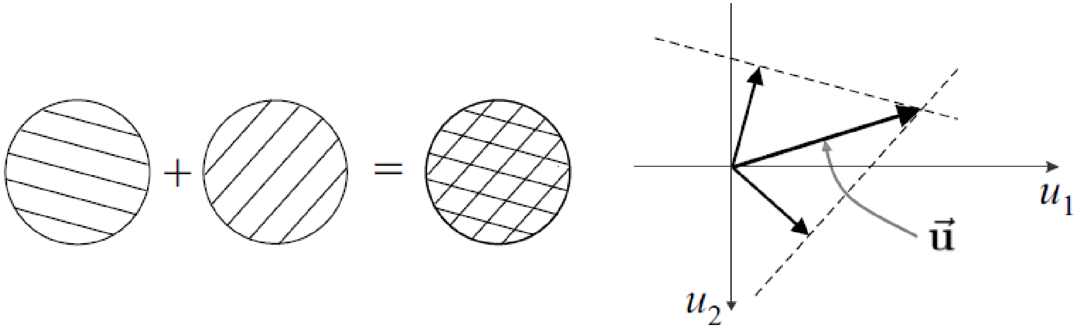
- Horn & Schunck
- Schunck
- Lukas and Kanade

Possible Solution: Neighbors

Two adjacent pixels which are part of the same rigid object:

- we can calculate normal flows $\mathbf{v}_{n1}$ and $\mathbf{v}_{n2}$
- Two OF equations for 2 parameters of flow: $\bar{v} = \binom{v}{u}$

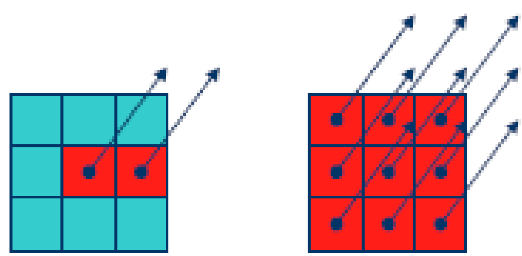
$$\nabla I_1 . \bar{v} - I_{t1} = 0$$
$$\nabla I_2 . \bar{v} - I_{t2} = 0$$

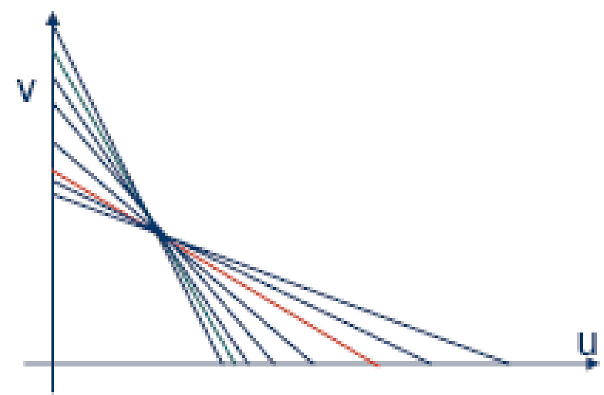


Considering Neighbor Pixels

Schunck

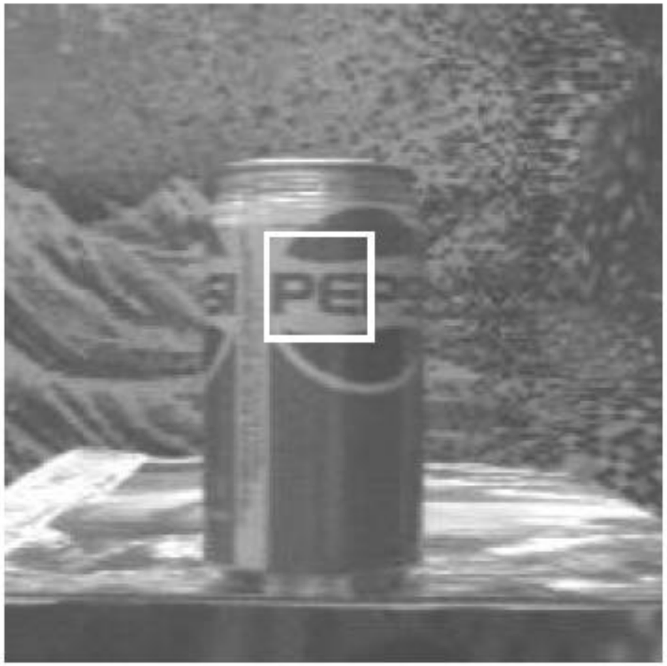
- If two neighboring pixels move with same velocity
  - Corresponding flow equations intersect at a point in (u,v) space
  - Find the intersection point of lines
  - If more than 1 intersection points find clusters
  - Biggest cluster is true flow

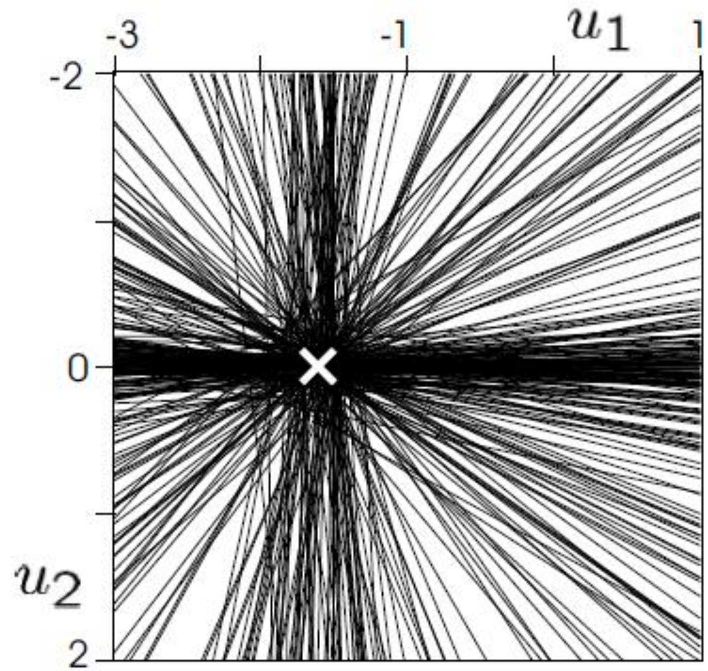




Alper Yilmaz, Fall 2005 UCF

Considering Neighbor Pixels





Cluster center provides velocity vector common for all pixels in patch.

Jepson, Toronto: http://www.cs.toronto.edu/pub/jepson/teaching/vision/2503/opticalFlow.pdf

Optical Flow

- Brightness Constancy
- The Aperture problem
- Regularization: Horn & Schunck
- Lucas-Kanade
- Coarse-to-fine
- Parametric motion models
- Direct depth
- SSD tracking
- Robust flow
- Bayesian flow

Horn & Schunck algorithm

Horn and Schunck's approach — Regularization

Two terms are defined as follows:

- Departure from smoothness

$$e_s = \int\int_{\Omega}((u_x^2 + u_y^2) + (v_x^2 + v_y^2))dxdy$$

- Error in optical flow constaint equation

$$e_c = \int\int_{\Omega}(E_x u + E_y v + E_t)^2 dxdy$$

The formulation is to minimize the linear combination of e_s and e_c,

$$e_s + \lambda e_c$$

where λ is a parameter.
Note: In this formulation, u and v are functions of x and y. Physically, u is the x-component of the motion, and v is the y-component of the motion.

Horn & Schunck algorithm

$$\int_D (\nabla I \cdot \vec{v} + I_t)^2 + \lambda^2 \left[\left(\frac{\partial v_x}{\partial x}\right)^2 + \left(\frac{\partial v_x}{\partial y}\right)^2 + \left(\frac{\partial v_y}{\partial x}\right)^2 + \left(\frac{\partial v_y}{\partial y}\right)^2 \right] dxdy$$

Additional smoothness constraint
(usually motion field varies smoothly in the image
→ penalize departure from smoothness) :

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

OF constraint equation term
(formulate error in optical flow constraint) :

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

minimize $e_s + \lambda e_c$

Horn & Schunck algorithm

Variational calculus: Pair of second order differential equations that can be solved iteratively.

- Define an energy function and minimize

$$E(x,y) = (uI_x + vI_y + I_t)^2 + \lambda \overbrace{(u_x^2 + u_y^2 + v_x^2 + v_y^2)}^{f}$$

- Differentiate w.r.t. unknowns *u* and *v*

$$\frac{\partial E}{\partial u} = 2I_x(uI_x + vI_y + I_t) + \frac{\partial f}{\partial u}$$

$$\frac{\partial f}{\partial u} = \frac{\partial}{\partial u}\frac{\partial u}{\partial x} + \frac{\partial}{\partial u}\frac{\partial u}{\partial y} = 2(u_{xx} + u_{yy})$$

laplacian of *u*

$$\frac{\partial E}{\partial v} = 2I_y(uI_x + vI_y + I_t) + 2(v_{xx} + v_{yy})$$

laplacian of *v*

Horn & Schunck algorithm

$$I_x(uI_x + vI_y + I_t) + \Delta^2 u = 0 \qquad I_y(uI_x + vI_y + I_t) + \Delta^2 v = 0$$

- Laplacian controls smoothness of optical flow
  - A particular choice can be $\Delta^2 u = u - u_{avg}$, $\Delta^2 v = v - v_{avg}$.
- Rearranging equations

$$u\left(\lambda + I_x^2\right) + vI_xI_y + I_xI_t - \lambda u_{avg} = 0$$

$$v\left(\lambda + I_y^2\right) + uI_xI_y + I_yI_t - \lambda v_{avg} = 0$$

- 2 equations 2 unknowns
- Write v in terms of u
- Plug it in the other equation

$$u = u_{avg} - I_x\left(\frac{I_x u_{avg} + I_y v_{avg} + I_t}{I_x^2 + I_y^2 + \lambda}\right) \qquad v = v_{avg} - I_y\left(\frac{I_x u_{avg} + I_y v_{avg} + I_t}{I_x^2 + I_y^2 + \lambda}\right)$$

- Iteratively compute u and v
  - Assume initially u and v are 0
  - Compute u_{avg} and v_{avg} in a neighborhood

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

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

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

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

3. Information spreads from corner-type patterns

Discrete Optical Flow Algorithm

Consider image pixel (i, j)

- Departure from Smoothness Constraint:

$$s_{ij} = \frac{1}{4}[(u_{i+1,j} - u_{i,j})^2 + (u_{i,j+1} - u_{i,j})^2 +$$
$$(v_{i+1,j} - v_{i,j})^2 + (v_{i,j+1} - v_{i,j})^2]$$

- Error in Optical Flow constraint equation:

$$c_{ij} = (E^{ij}{}_x\, u_{ij} + E^{ij}{}_y\, v_{ij} + E^{ij}{}_t)^2$$

- We seek the set $\{u_{ij}\}$ & $\{v_{ij}\}$ that minimize:

$$e = \sum_i \sum_j (s_{ij} + \lambda\, c_{ij})$$

NOTE: $\{u_{ij}\}$ & $\{v_{ij}\}$ show up in more than one term

Discrete Optical Flow Algorithm

- Differentiating e w.r.t v_{kl} & u_{kl} and setting to zero:

$$\frac{\partial e}{\partial u_{kl}} = 2\,(u_{kl} - \overline{u_{kl}}) + 2\lambda\,(E_x^{kl} u_{kl} + E_y^{kl} v_{kl} + E_t^{kl})E_x^{kl} = 0$$

$$\frac{\partial e}{\partial v_{kl}} = 2\,(v_{kl} - \overline{v_{kl}}) + 2\lambda\,(E_x^{kl} u_{kl} + E_y^{kl} v_{kl} + E_t^{kl})E_y^{kl} = 0$$

- $\overline{v_{kl}}$ & $\overline{u_{kl}}$ are averages of (u,v) around pixel (k,l)

Update Rule:

$$u_{kl}^{n+1} = \overline{u_{kl}^{n}} - \frac{E_x^{kl}\,\overline{u_{kl}^{n}} + E_y^{kl}\,\overline{v_{kl}^{n}} + E_t^{kl}}{1 + \lambda\,[(E_x^{kl})^2 + (E_y^{kl})^2]} E_x^{kl}$$

$$v_{kl}^{n+1} = \overline{v_{kl}^{n}} - \frac{E_x^{kl}\,\overline{u_{kl}^{n}} + E_y^{kl}\,\overline{v_{kl}^{n}} + E_t^{kl}}{1 + \lambda\,[(E_x^{kl})^2 + (E_y^{kl})^2]} E_y^{kl}$$

Horn-Schunck Algorithm : Discrete Case

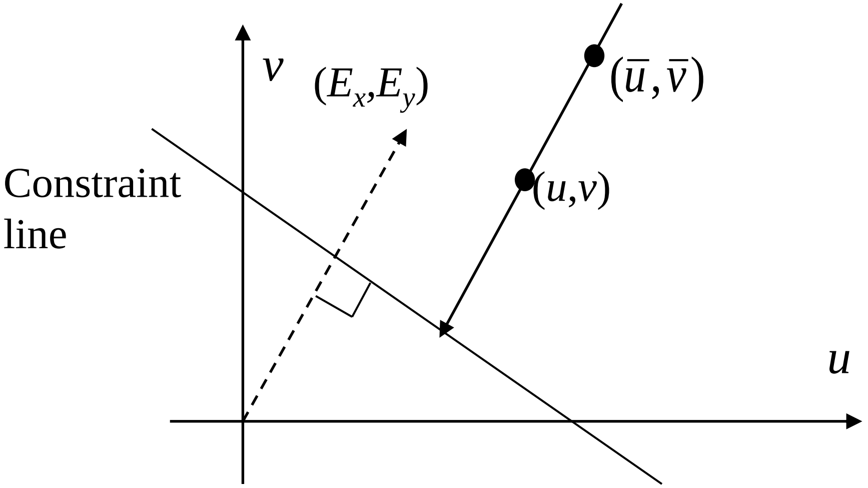
- Derivatives (and error functionals) are approximated by difference operators
- Leads to an iterative solution:

$$u_{ij}^{n+1} = \bar{u}_{ij}^{n} - \alpha E_x \qquad \alpha = \frac{E_x \bar{u}_{ij}^{n} + E_y \bar{v}_{ij}^{n} + E_t}{1 + \lambda (E_x^2 + E_y^2)}$$

$$v_{ij}^{n+1} = \bar{v}_{ij}^{n} - \alpha E_y$$

$\bar{u}, \bar{v}$ is the average of values of neighbors

Intuition of the Iterative Scheme



The new value of (u,v) at a point is equal to the average of surrounding values minus an adjustment in the direction of the brightness gradient

Horn - Schunck Algorithm

begin

for $j := 1$ **to** N **do** **for** $i := 1$ **to** M **do** **begin**

calculate the values $E_x(i, j, t)$, $E_y(i, j, t)$, and $E_t(i, j, t)$ using a selected approximation formula;

{ special cases for image points at the image border have to be taken into account }

initialize the values $u(i, j)$ and $v(i, j)$ with zero

end {for};

choose a suitable weighting value λ; { e.g $\lambda = 10$ }

choose a suitable number $n_0 \geq 1$ of iterations; { $n_0 = 8$ }

$n := 1$; { iteration counter }

while $n \leq n_0$ **do begin**

for $j := 1$ **to** N **do** **for** $i := 1$ **to** M **do** **begin**

$\overline{u} := \frac{1}{4}\left(u(i-1, j) + u(i+1, j) + u(i, j-1) + u(i, j+1)\right)$;

$\overline{v} := \frac{1}{4}\left(v(i-1, j) + v(i+1, j) + v(i, j-1) + v(i, j+1)\right)$;

{ treat image points at the image border separately }

$$\alpha := \frac{E_x(i,j,t)\overline{u} + E_y(i,j,t)\overline{v} + E_t(i,j,t)}{1 + \lambda\left(E_x^2(i,j,t) + E_y^2(i,j,t)\right)} \cdot \lambda \ ;$$

$u(i, j) := \overline{u} - \alpha \cdot E_x(i, j, t)$; $v(i, j) := \overline{v} - \alpha \cdot E_y(i, j, t)$

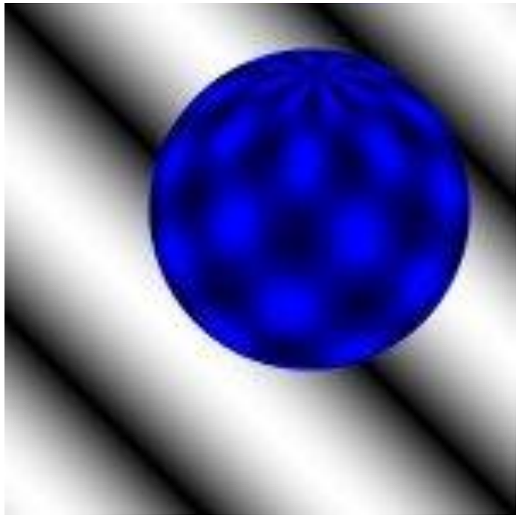
end {for};

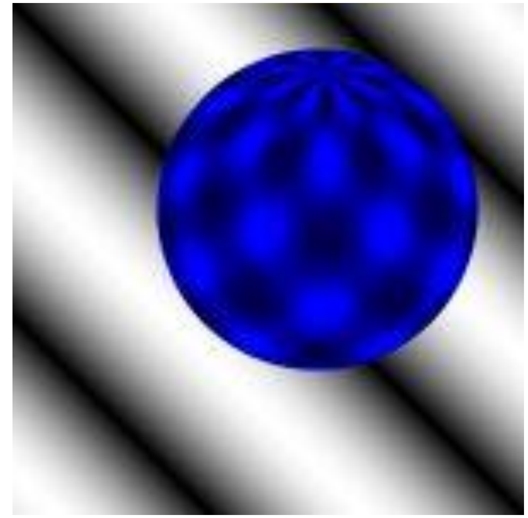
$n := n + 1$

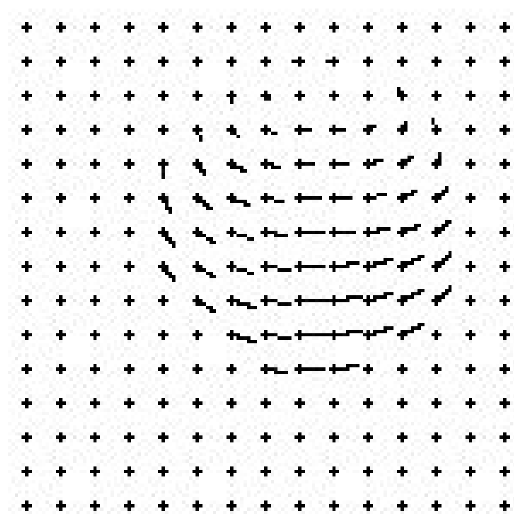
end {while}

end;

Example







http://of-eval.sourceforge.net/

Results

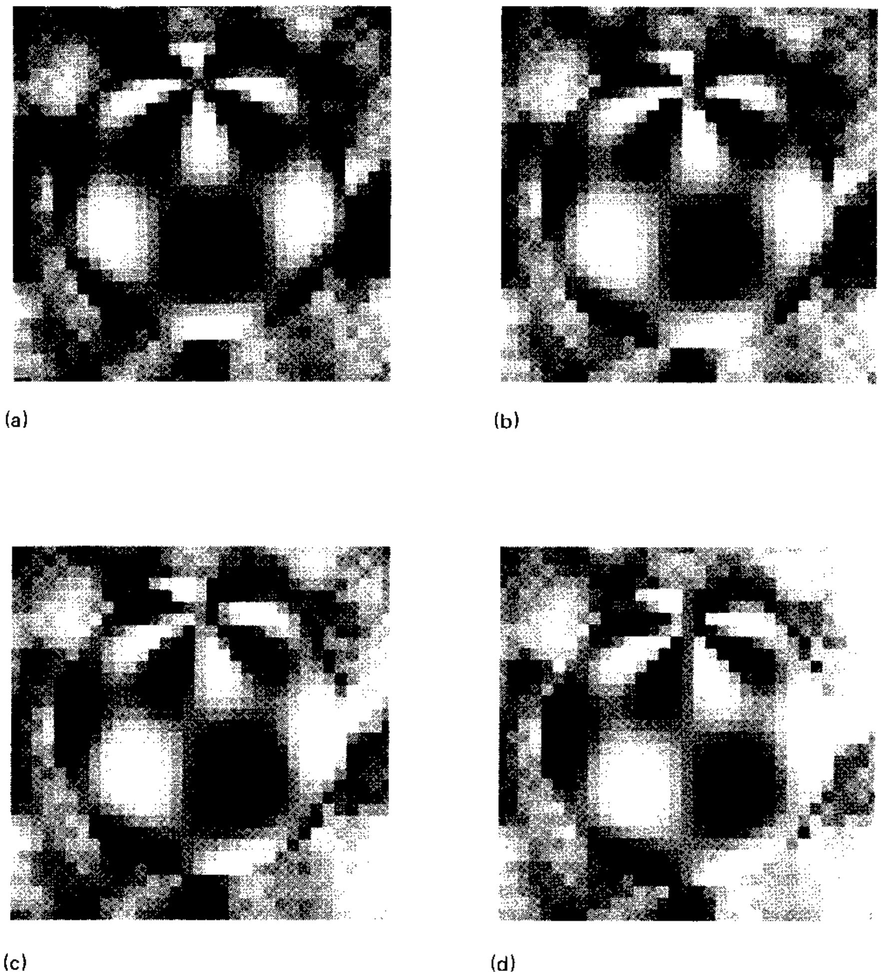


Figure 12-8. Four frames of a synthetic image sequence showing a sphere slowly rotating in front of a randomly patterned background.

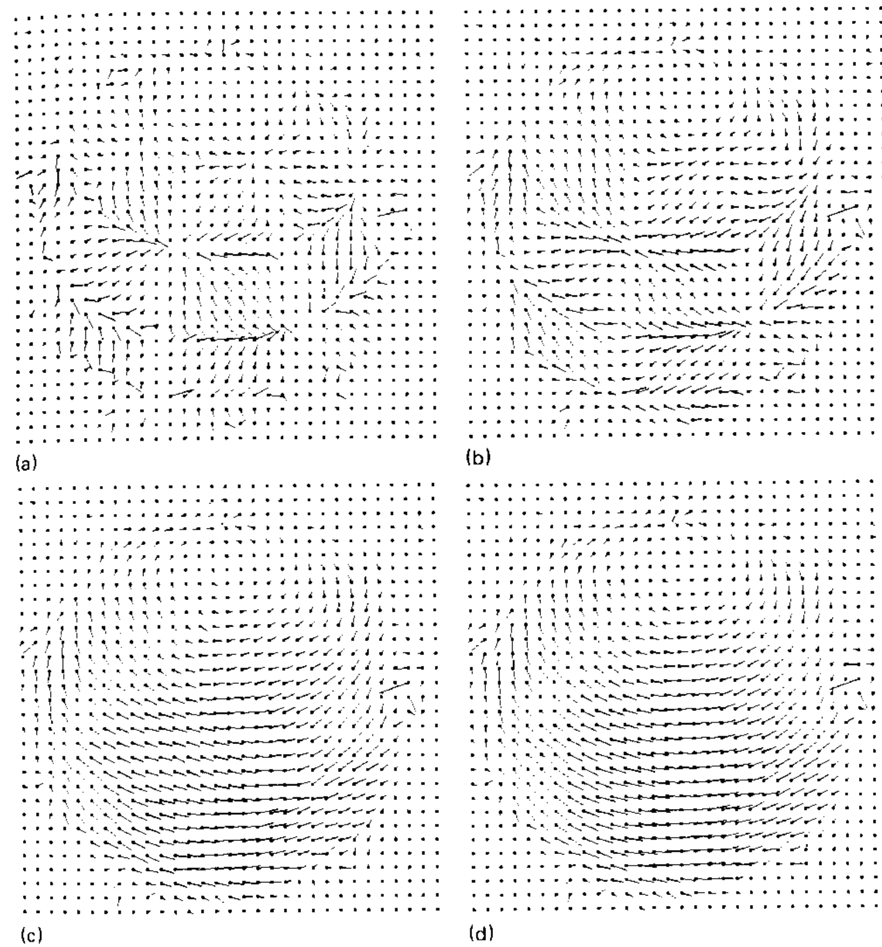
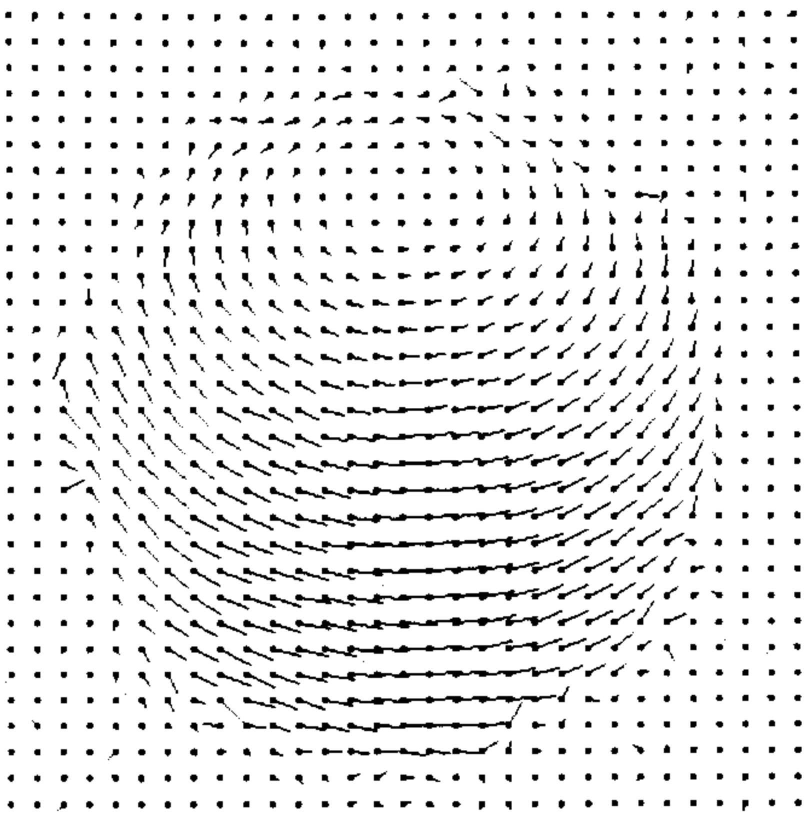
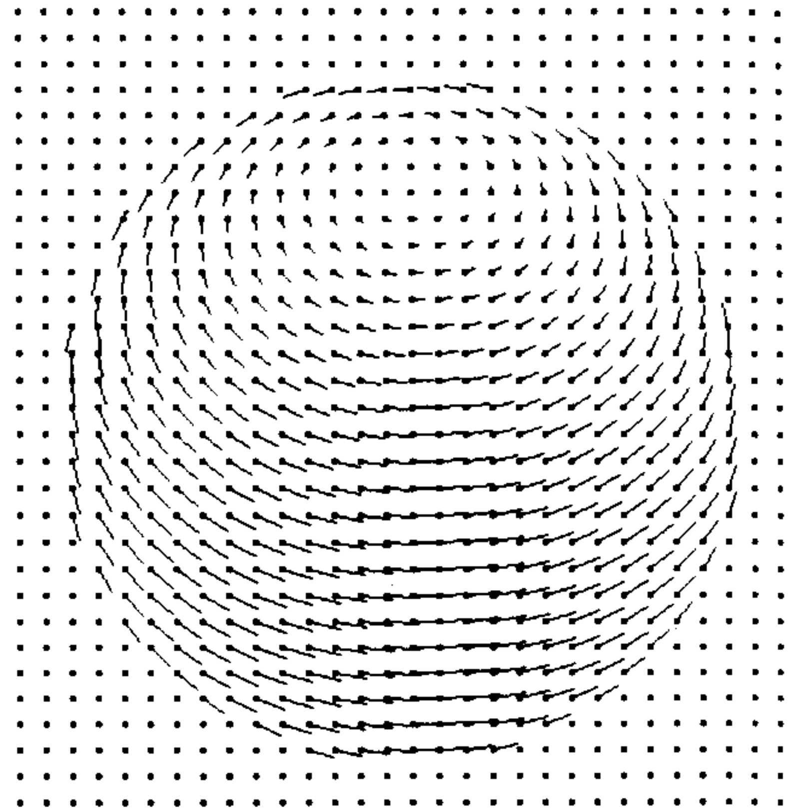


Figure 12-9. Estimates of the optical flow shown in the form of needle diagrams after 1, 4, 16, and 64 iterations of the algorithm.

Results

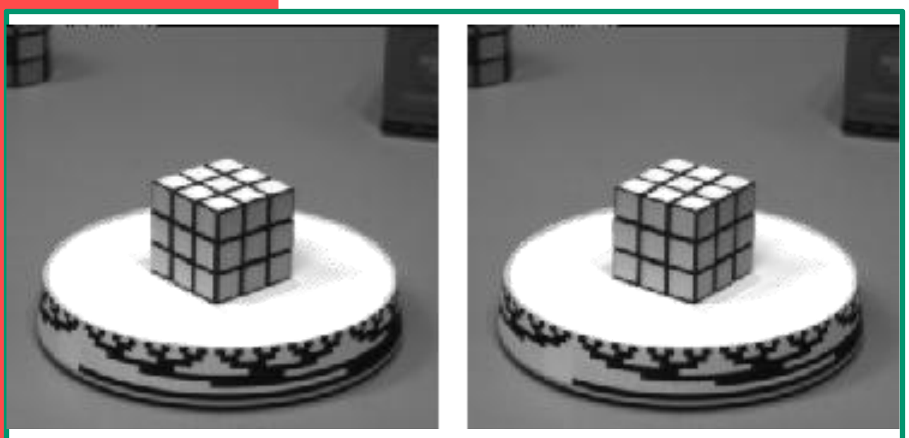


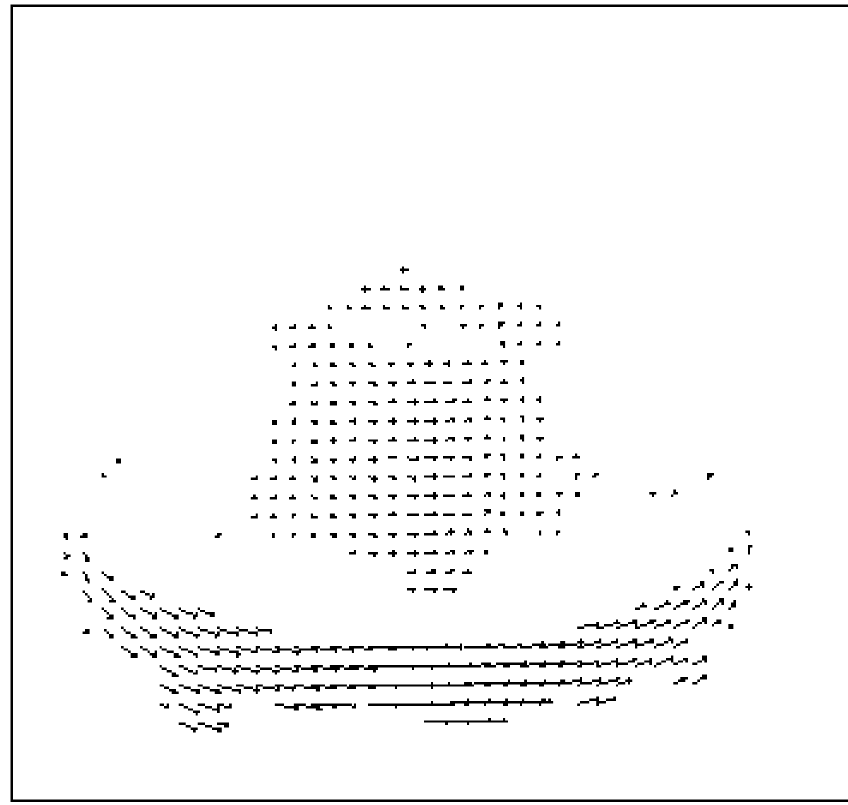


(a) (b)

Figure 12-10. (a) The estimated optical flow after several more iterations. (b) The computed motion field.

Optical Flow Result



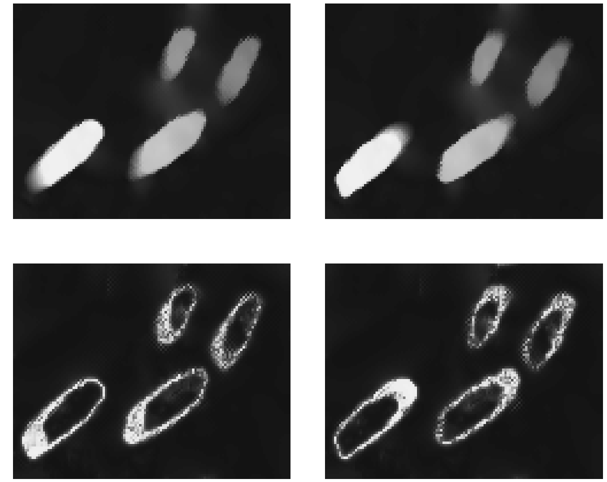


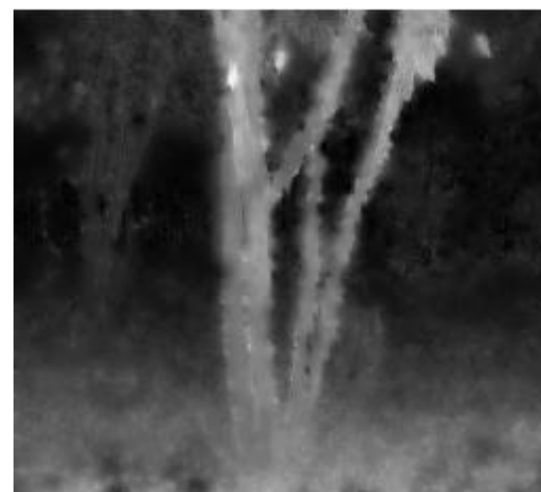
Horn & Schunck, remarks

1. Errors at boundaries

2. Example of *regularisation*
   (selection principle for the solution of illposed problems)

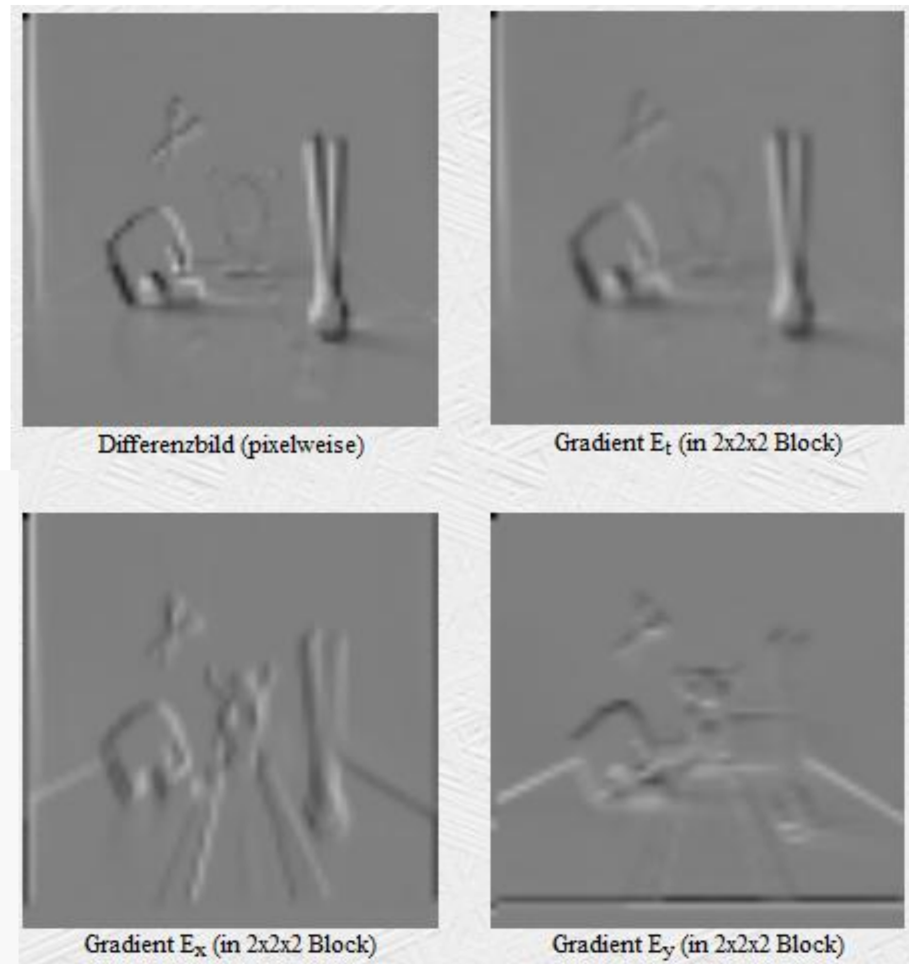
Results of an enhanced system





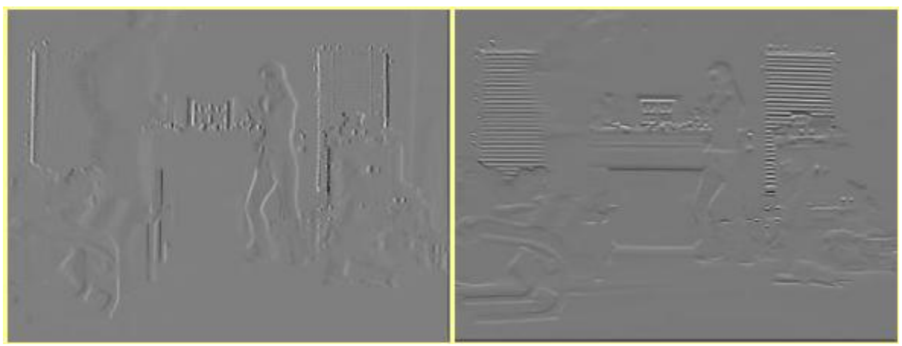
Results

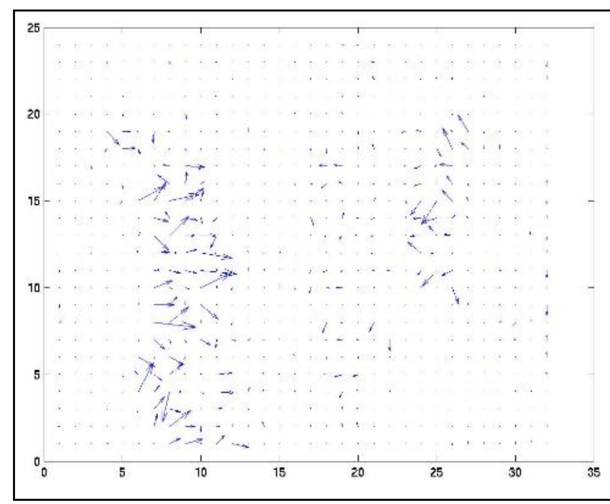
http://www-student.informatik.uni-bonn.de/~gerdes/OpticalFlow/index.html



Results

http://www.cs.utexas.edu/users/jmugan/GraphicsProject/OpticalFlow/





Optical Flow

- Brightness Constancy
- The Aperture problem
- Regularization
- Lucas-Kanade
- Coarse-to-fine
- Parametric motion models
- Direct depth
- SSD tracking
- Robust flow
- Bayesian flow