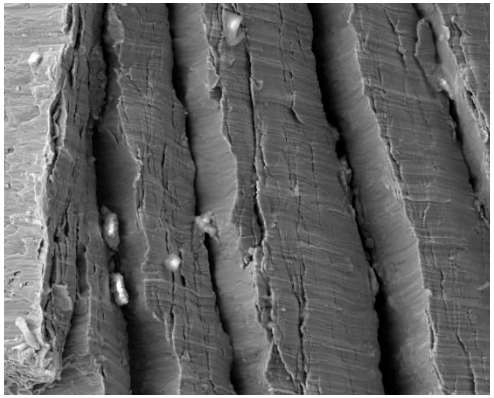
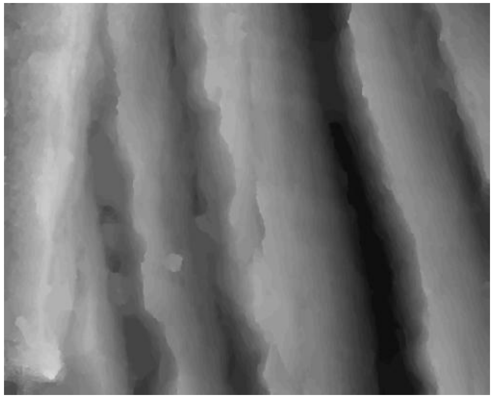
Optical Flow

Thomas Pock

$$\int_\Omega f(x, u(x), \nabla u(x))dx \Leftrightarrow \sup_{\phi \in K} \int_{\Omega \times R} \phi \cdot D\mathbf{1}_u$$



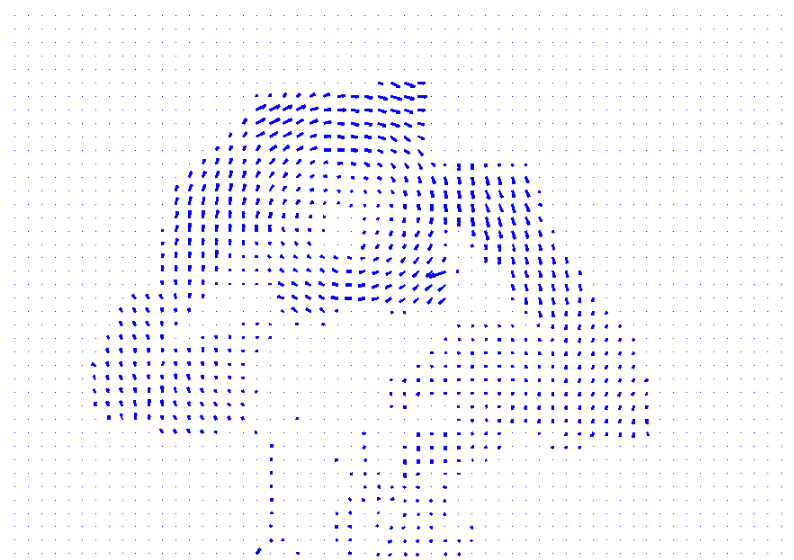


Optical Flow (I)

- Content
  - Introduction
  - Local approach (Lucas Kanade)

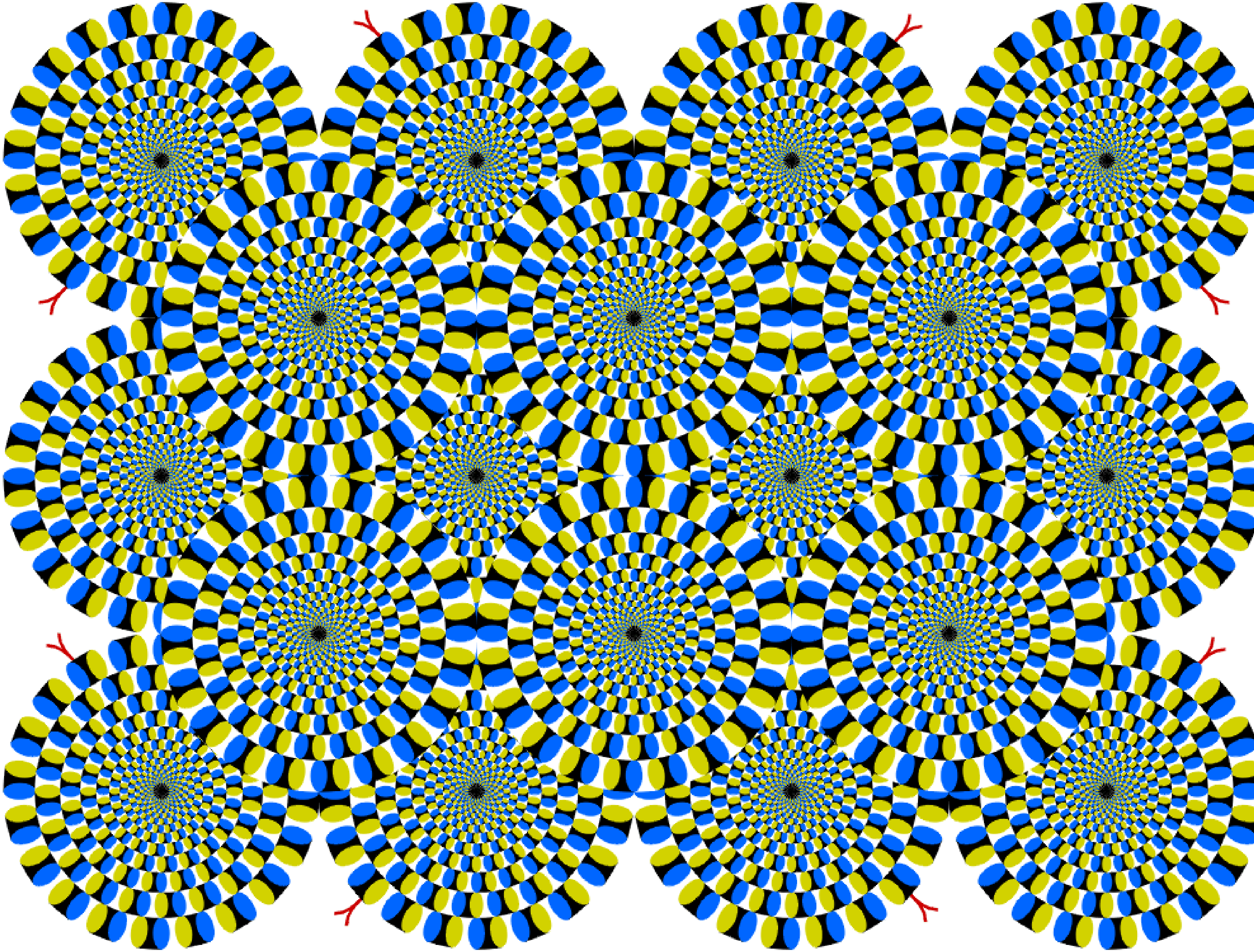
What is „Optical Flow“?

- Optical Flow is a major task of every biological and artificial visual system
- Is the aparent motion in images sequenzes.
- Can be seen as a velocity field that transforms one image to the next image in a sequence



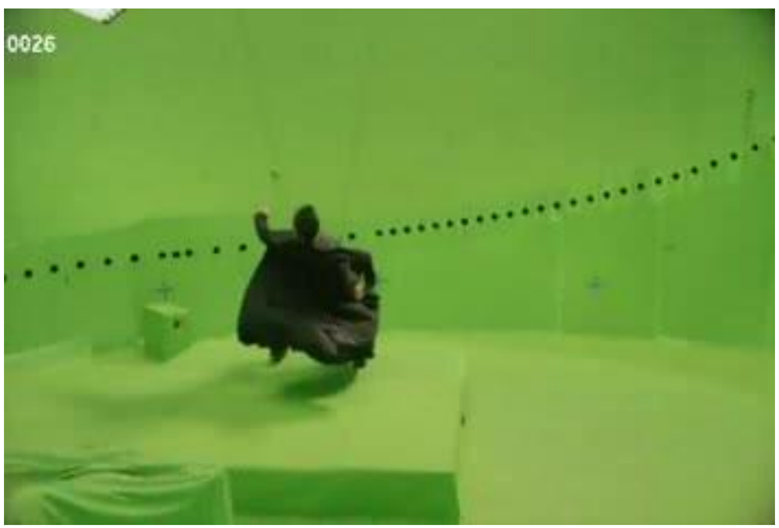
The apparent motion

- Optical flow is not the “true” 3D motion of the objects
- It is the 3D motion projected to the camera plane
- The “true” 3D motion is called the “scene flow” and additionally requires 3D information of the scene



Applications of Optical Flow

- Tracking
- Video Compression (Recent MPEG Standard)
- 3D Reconstruction (Stereo)
- Segmentation
- Object Detection
- Video Interpolation in time (The Matrix)

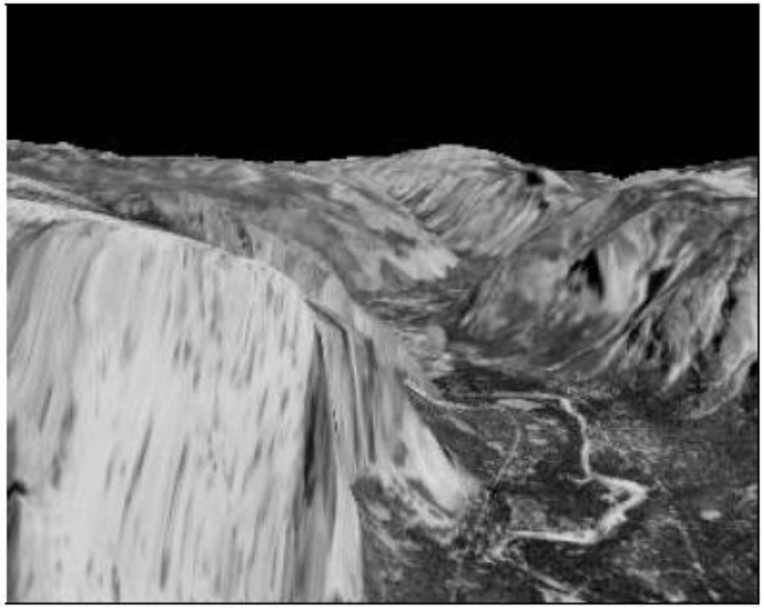


History

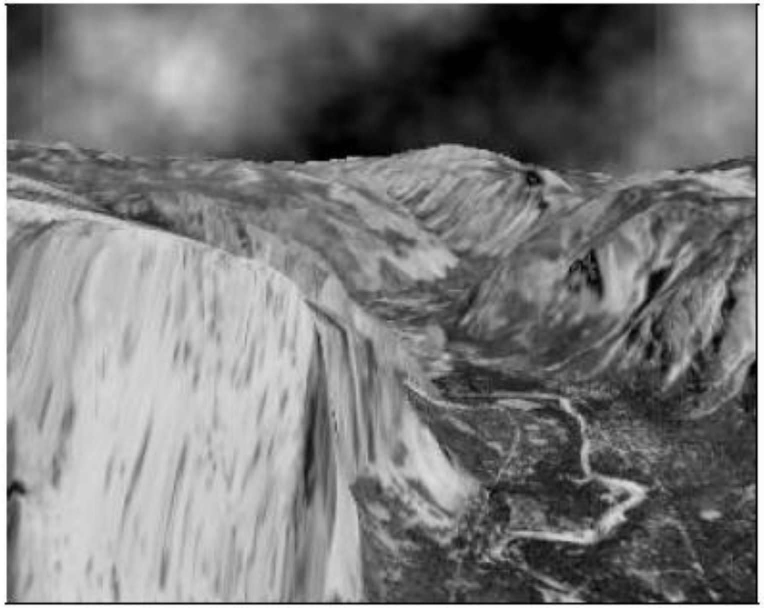
- Computing Optical Flow started in the early 80's and is still a hot research topic
  - 1980: Horn and Schunck (global approach)
  - 1981: Lucas and Kanade (local approach)
  - 1989: Shulman and Herve (discontinuity preserving)
  - 1993: Black and Anandan (robust optical flow)
  - 1999: Alvarez et al. (PDE model)
  - 2003: Bruhn et al. (realtime optical flow using mulitgrid)
  - 2004: Brox et al. (high accuracy using warping)
  - 2007: Zach, Pock, Bischof: (duality based minimization)
  - …

Evaluation

- For a long time, the so-called Yosemite sequence (Barron et al. 1994) was used to evaluate the algorithms



Yosemite



Yosemite with clouds

Evaluation

- Middlebury optical flow benchmark (Baker et al. 2007)

	Hidden Texture				Synthetic			Stereo	High-speed camera (no GT)			
	Army	Mequon	Schefflera	Wooden	Grove	Urban	Yosemite	Teddy	Backyard	Basketball	Dumptruck	Evergreen
# frames	8	8	8	8	8	8	8	2	8	8	8	8
Flow Eval	yes	yes	yes	yes	yes	yes	yes	yes	--	--	--	--
Interp Eval	--	yes	yes	--	--	yes	--	yes	yes	yes	yes	yes

Optical flow evaluation results

Show images: below table above table in window

Statistics: Average SD R0.5 R1.0 R2.0 A50 A75 A95
Error type: endpoint angle interpolation normalized interpolation

Average endpoint error	avg. rank	Army (Hidden texture) GT im0 im1 all	disc	untext	Mequon (Hidden texture) GT im0 im1 all	disc	untext	Schefflera (Hidden texture) GT im0 im1 all	disc	untext	Wooden (Hidden texture) GT im0 im1 all	disc	untext
MDP-Flow2 [70]	5.7	0.08 5	0.21 2	0.07 11	**0.15** 1	**0.48** 1	**0.11** 1	0.20 2	0.40 2	**0.14** 1	0.15 12	0.80 19	0.08 7
NN-field [73]	6.4	0.08 5	0.22 9	**0.05** 1	0.17 4	0.55 4	0.13 7	**0.19** 1	**0.39** 1	0.15 4	**0.09** 1	0.48 2	**0.05** 1
TC/T-Flow [81]	11.9	**0.07** 1	0.21 2	**0.05** 1	0.19 10	0.68 19	0.12 5	0.28 14	0.66 20	**0.14** 1	0.14 5	0.86 24	0.07 2
ADF [67]	12.5	0.08 5	0.22 9	0.06 3	0.18 6	0.62 11	0.14 11	0.29 18	0.71 23	0.17 10	0.16 22	0.91 32	0.07 2
Layers++ [37]	12.5	0.08 5	0.21 2	0.07 11	0.19 10	0.56 5	0.17 21	0.20 2	0.40 2	0.18 14	0.13 3	0.58 4	0.07 2
LME [72]	12.7	0.08 5	0.22 9	0.06 3	**0.15** 1	0.49 2	**0.11** 1	0.30 21	0.64 15	0.31 56	0.15 12	0.78 17	0.09 18
IROF++ [58]	13.1	0.08 5	0.23 13	0.07 11	0.21 20	0.68 19	0.17 21	0.28 14	0.63 14	0.19 22	0.15 12	0.73 12	0.09 18
nLayers [57]	13.2	**0.07** 1	**0.19** 1	0.06 3	0.22 25	0.59 7	0.19 34	0.25 9	0.54 8	0.20 31	0.15 12	0.84 22	0.08 7

Average endpoint error	Grove (Synthetic) GT im0 im1 all	disc	untext	Urban (Synthetic) GT im0 im1 all	disc	untext	Yosemite (Synthetic) GT im0 im1 all	disc	untext	Teddy (Stereo) GT im0 im1 all	disc	untext
MDP-Flow2 [70]	0.63 10	0.93 11	0.43 11	0.26 2	0.76 2	0.23 4	0.11 9	0.12 7	0.17 10	0.38 2	0.79 2	0.44 3
NN-field [73]	**0.41** 1	**0.61** 1	**0.20** 1	0.52 34	**0.64** 1	0.26 6	0.13 24	0.13 22	0.20 19	**0.35** 1	0.83 3	**0.21** 1
TC/T-Flow [81]	0.67 18	0.98 17	0.49 19	**0.22** 1	0.82 3	**0.19** 1	0.11 9	**0.11** 1	0.30 60	0.50 20	1.02 20	0.64 13
ADF [67]	0.69 20	1.03 20	0.47 14	0.43 14	0.91 6	0.28 8	0.12 16	0.12 7	0.20 19	0.43 5	0.88 7	0.63 11
Layers++ [37]	0.48 2	0.70 2	0.33 4	0.47 23	1.01 10	0.33 24	0.15 42	0.14 37	0.24 35	0.46 10	0.88 7	0.72 23
LME [72]	0.66 14	0.96 14	0.53 22	0.33 4	1.18 22	0.28 8	0.12 16	0.12 7	0.18 13	0.44 6	0.91 9	0.61 9
IROF++ [58]	0.60 7	0.89 7	0.42 10	0.43 14	1.08 16	0.31 17	0.10 4	0.12 7	0.12 4	0.47 12	0.98 15	0.68 20
nLayers [57]	0.53 3	0.78 4	0.34 5	0.44 18	0.84 4	0.30 14	0.13 24	0.13 22	0.20 19	0.47 12	0.97 14	0.67 17

Evaluation

- SINTEL open source synthetic movie, Butler et al. 2012

Basic assumptions

- Brightness constancy assumption
  - The intensities remain constant, although the location might change.
  - Problems by changing illumination
  - Can be generalized to a „feature constancy“ assumption
- Spatial coherence assumption
  - Neighbouring pixels are likely to have the same motion
  - Difficult to find a good model
- Temporal persistence
  - Motion changes gradually over time
  - Only useful in case of high frame rates (small motion)

Brightness constancy assumption

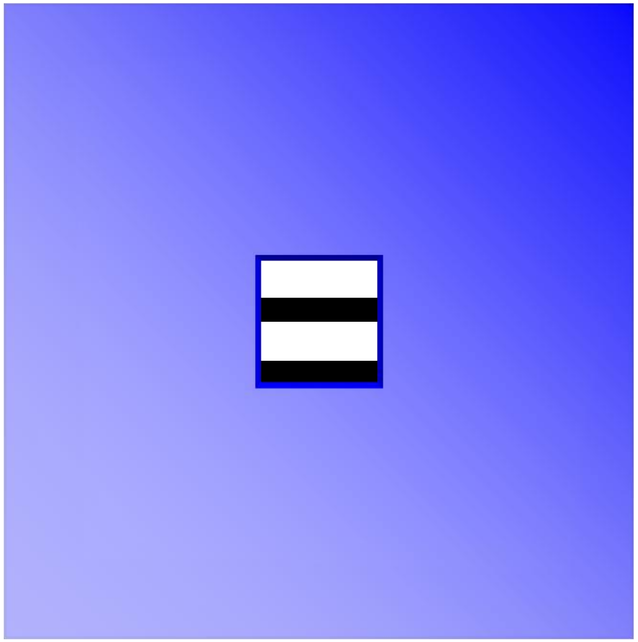
- We assume that intensity patterns only change their positions

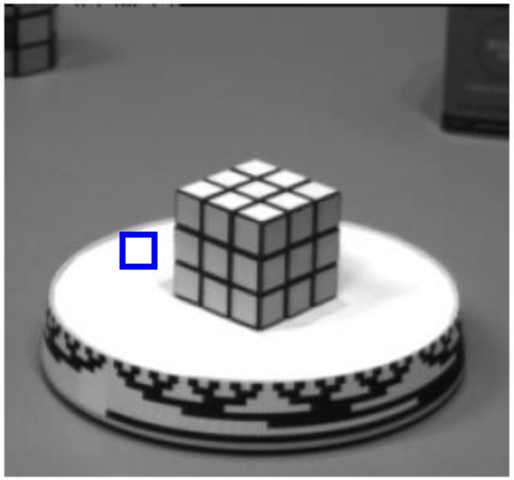
frame t frame t+1

$$I(x,y,t) - I(x+u, y+v, t+1) \approx 0$$

Difficulties of the brightnes constancy assumption

- Aperture Problem: Only the normal flow can be estimated
- Untextured areas: No information in untextured areas





Changing illumination

- A changing illumination induces an optical flow that does not correspond to the motion of the object

(a) (b)

- A changing illumination causes an optical flow although the object does not move

What to do if the brightness constancy assumption is violated?

- If the illumination changes from frame to frame, the brightness constancy assumption may be violated
- A simple idea is to perform a structure-texture decomposition
- Although the absolute intensity values might change, the texture part stays the same

1. Low-pass filter the images (e.g. total variation smoothing)
2. Subtract the low-pass filtered images from the original images
3. Work with the resulting images

Structure - Texture Decomposition

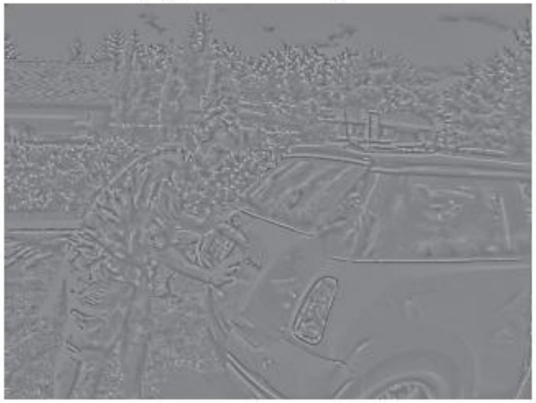
(a) original

(b) structure part

(c) texture part

(d) original

(e) structure part



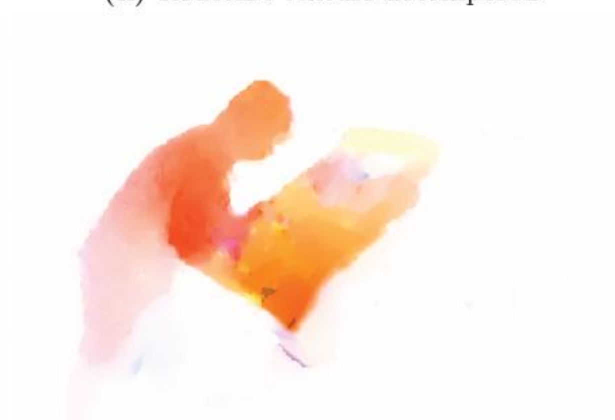
(f) texture part

Structure - Texture Decomposition

(g) original

(h) structure-texture decomposed

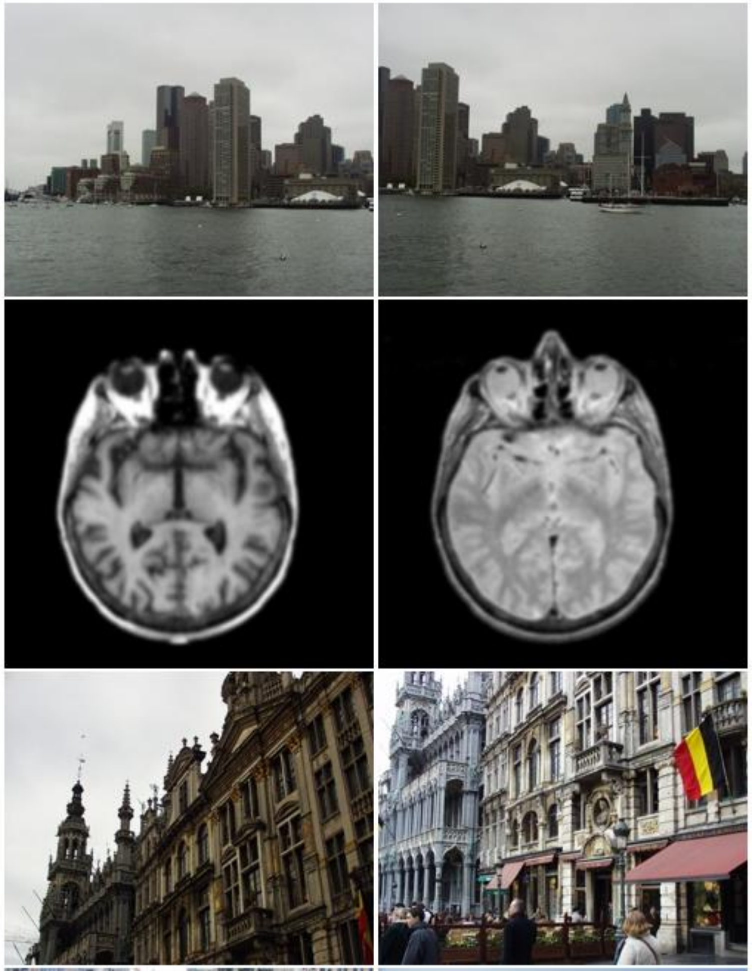
(i) original



(j) structure-texture decomposed

Beyond the brightness constancy assumption

- What can we do if we want to compute the optical flow between such images?
- We can work on feature transforms such as SIFT
- Each pixel in the images is replaced by its SIFT feature.
- SIFT-Flow [Liu, Yuen, Torralba, 2011]



The optical flow constraint (OFC)

- Brightness constancy assumption

$$I(x+\Delta x, y+\Delta y, t+\Delta t) = I(x,y,t)\,, \quad \Delta x,\ \Delta y,\ \Delta t \quad \text{small}$$

- Taylor development leads to the linearized Brightness Constancy Assumption

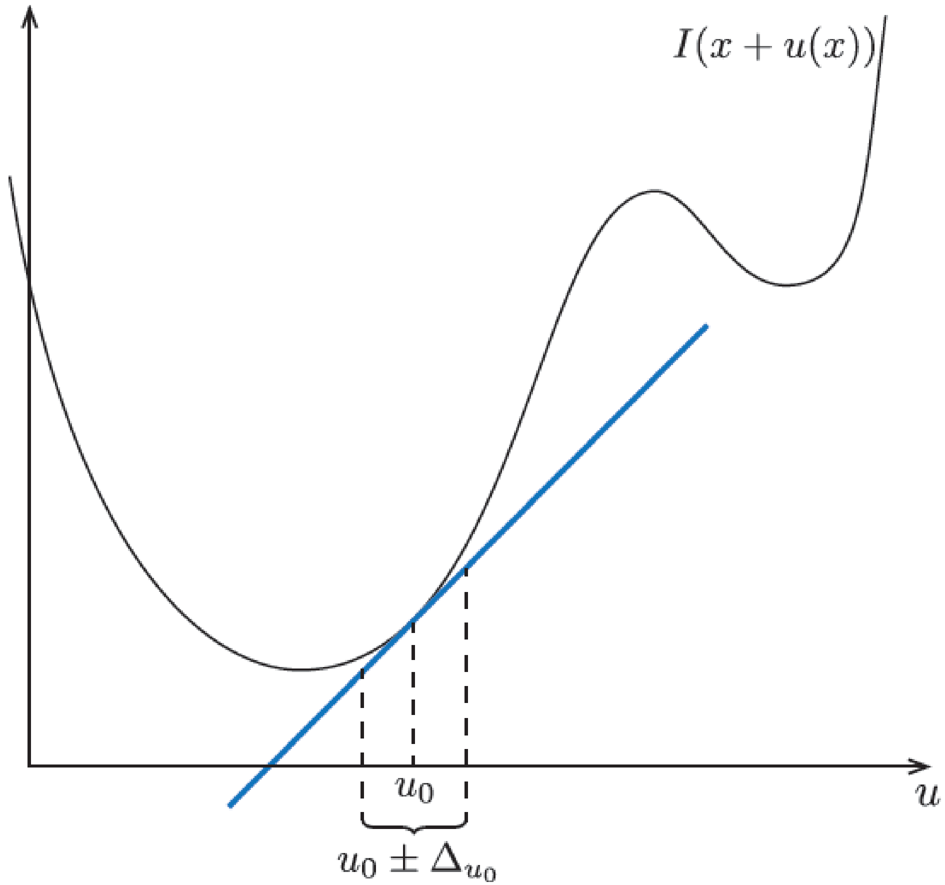
$$I_t + (\nabla I)^T \begin{pmatrix} u-u_0 \\ v-v_0 \end{pmatrix} = 0$$

- Error function over the whole image

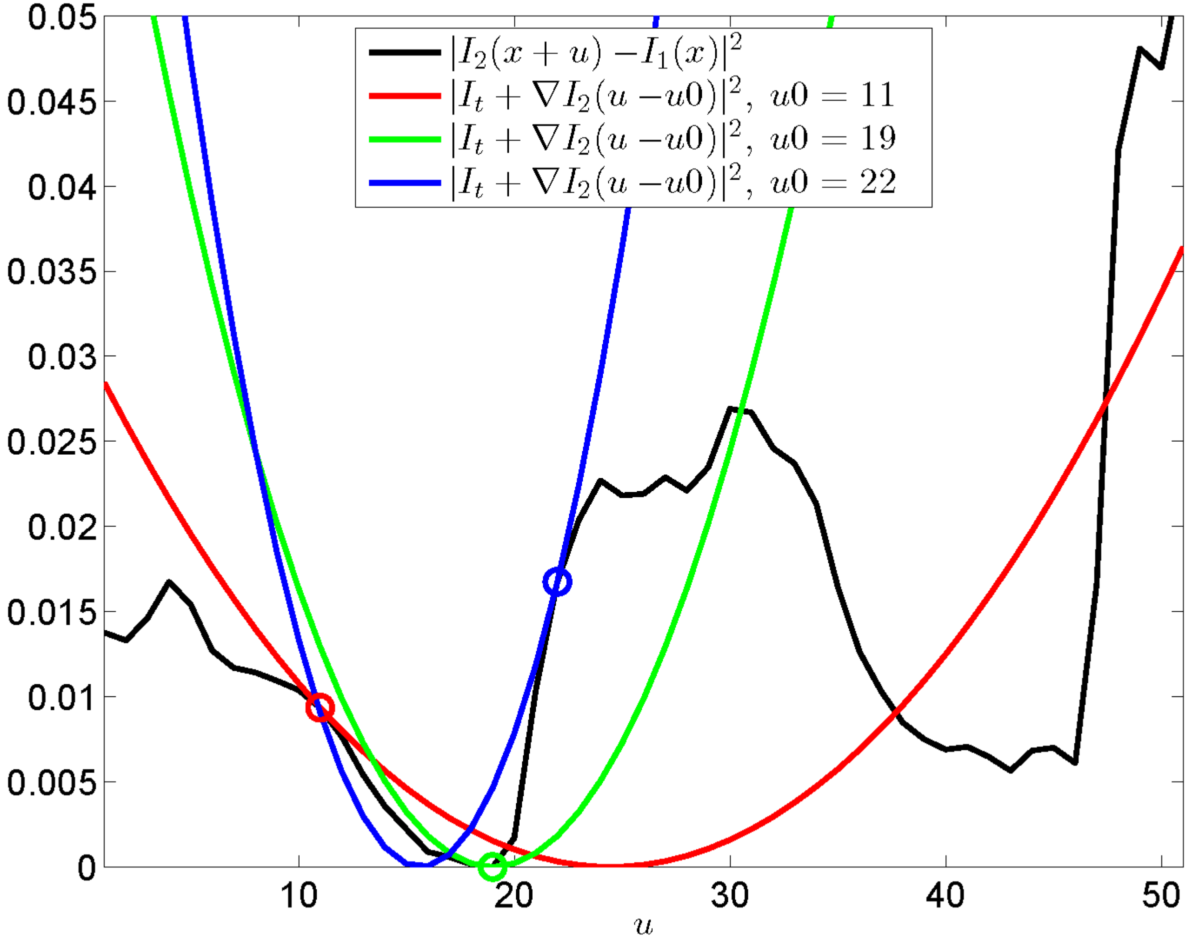
$$E_{FC} = \tfrac{1}{p} \| I_t + (\nabla I)^T \begin{pmatrix} u-u_0 \\ v-v_0 \end{pmatrix} \|_p^p\,, \quad p \in \{1,2\}$$

- Underdetermined problem

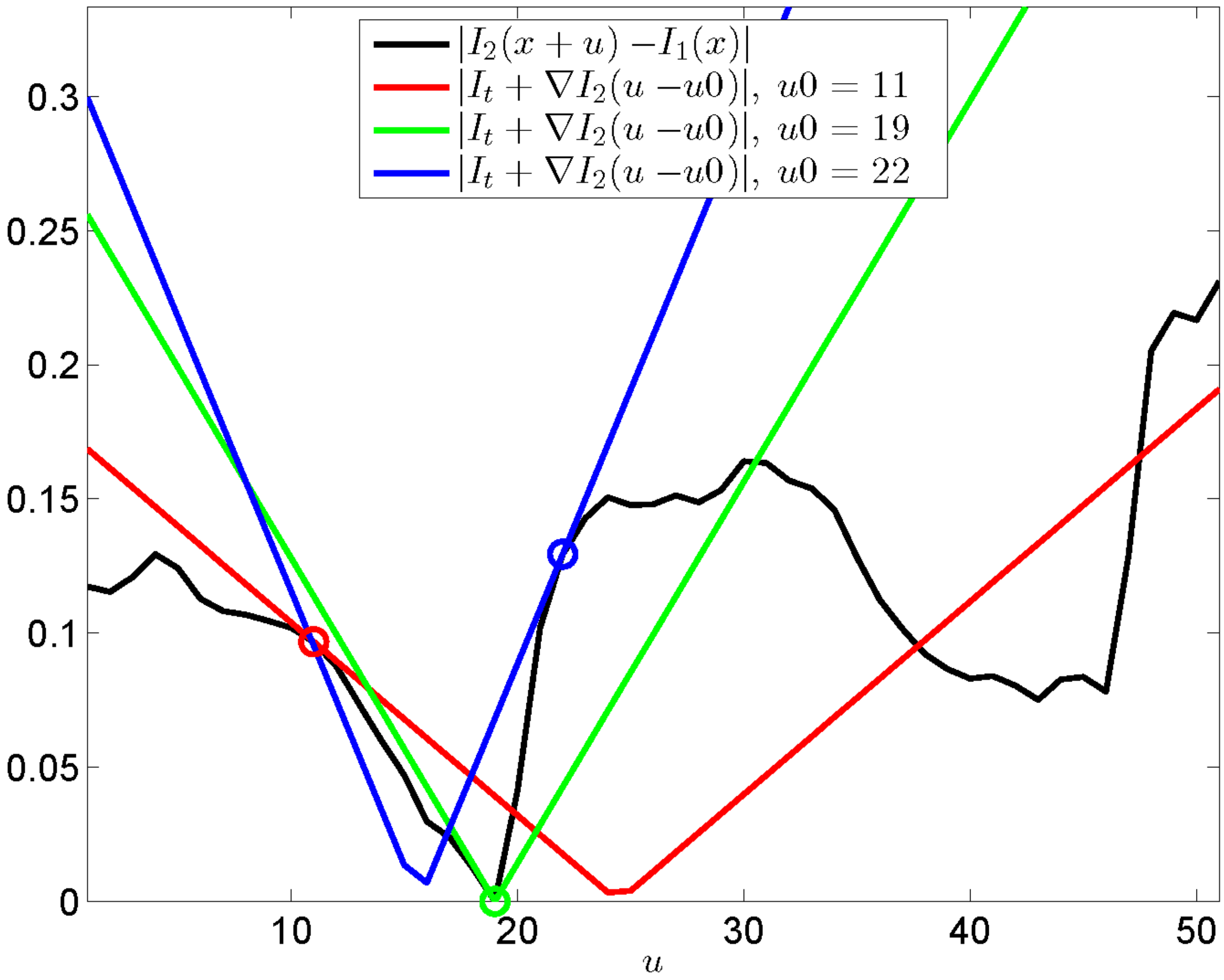
Linearization of the Image



The Linearized Data Term (p=2)

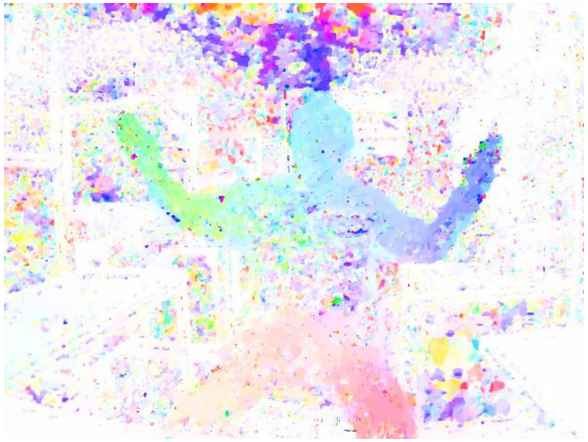


The Linearized Data Term (p=1)



Spatial coherence assumption

- Describes the a-priori assumption about flow fields
- Can be learned from statistics of natural flow fields

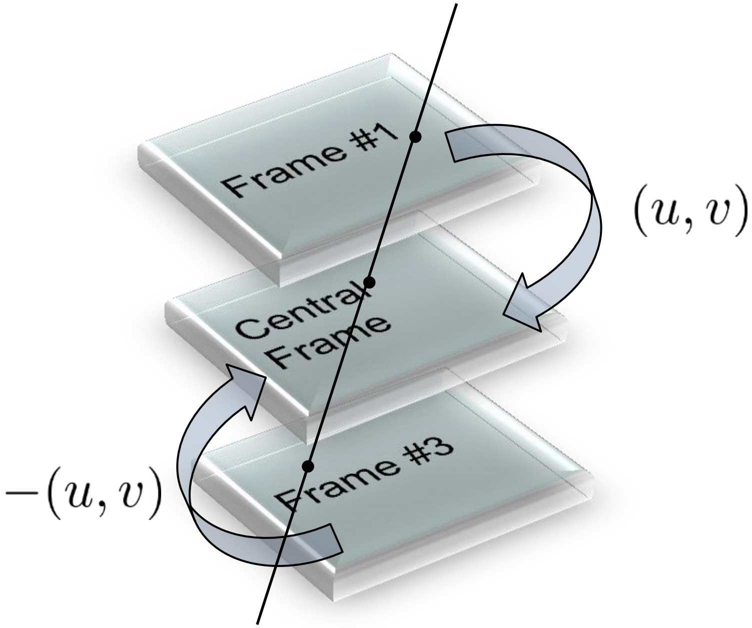


without spatial coherence

with spatial coherence

Temporal persistency

- Idea: The motion of objects does not suddenly change
- for example, one can assume linear motion model between three frames
- Can be generalized to higher-order models
- Unfortunately, it does not improve too much in practice



General outlook

- We will now discuss three methods
  - Lukas – Kanade method
  - Horn – Schunck method
  - TV-L1 method
- All methods can only be used to estimate small motion
- We will first assume that we only have small motion

- Large motion can be computed using a coarse-to-fine warping framework
- Finally we will discuss Matlab implementations of all three methods

The Lucas Kanade (LK) method

An Iterative Image Registration Technique with an Application to Stereo Vision

Bruce D. Lucas
Takeo Kanade

Computer Science Department
Carnegie-Mellon University
Pittsburgh, Pennsylvania 15213

Abstract

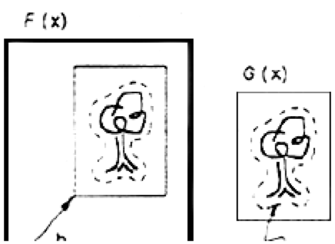
Image registration finds a variety of applications in computer vision. Unfortunately, traditional image registration techniques tend to be costly. We present a new image registration technique that makes use of the spatial intensity gradient of the images to find a good match using a type of Newton-Raphson iteration. Our technique is faster because it examines far fewer potential matches between the images than existing techniques. Furthermore, this registration technique can be generalized to handle rotation, scaling and shearing. We show show our technique can be adapted for use in a stereo vision system.

1. Introduction

Image registration finds a variety of applications in

2. The registration problem

The translational image registration problem can be characterized as follows: We are given functions $F(x)$ and $G(x)$ which give the respective pixel values at each location x in two images, where x is a vector. We wish to find the disparity vector h which minimizes some measure of the difference between $F(x + h)$ and $G(x)$, for x in some region of interest R. (See figure 1).



A local approach

- Impossible to compute a dense flow field by only using the optical flow constraint
- Basic idea: Introduce additional constraints
  - Flow field should be locally smooth
  - Assume that a certain neighborhood has the same motion
  - E.g. 5x5 pixels would give us 25 equations instead of one !

$$0 = I_t(\mathbf{p_i}) + \nabla I(\mathbf{p_i}) \cdot [u \;\; v]$$

$$\underbrace{\begin{bmatrix} I_x(\mathbf{p}_1) & I_y(\mathbf{p}_1) \\ I_x(\mathbf{p}_2) & I_y(\mathbf{p}_2) \\ \vdots & \vdots \\ I_x(\mathbf{p}_{25}) & I_y(\mathbf{p}_{25}) \end{bmatrix}}_{\substack{A \\ 25\times2}} \underbrace{\begin{bmatrix} u \\ v \end{bmatrix}}_{\substack{d \\ 2\times1}} = - \underbrace{\begin{bmatrix} I_t(\mathbf{p}_1) \\ I_t(\mathbf{p}_2) \\ \vdots \\ I_t(\mathbf{p}_{25}) \end{bmatrix}}_{\substack{b \\ 25\times1}}$$

Lucas-Kanade optical flow

- We now have more equations than unknowns

$$\underset{25\times2}{A}\ \underset{2\times1}{d} = \underset{25\times1}{b} \longrightarrow \text{minimize } \|Ad - b\|^2$$

- Solution: solve least squares problem
  - minimum least squares solution given by solution of:

$$(A^T A)\ d = A^T b$$

$$\underset{2\times2}{\underbrace{\begin{bmatrix} \sum I_x I_x & \sum I_x I_y \\ \sum I_x I_y & \sum I_y I_y \end{bmatrix}}_{A^T A}} \underset{2\times1}{\begin{bmatrix} u \\ v \end{bmatrix}} = - \underset{2\times1}{\underbrace{\begin{bmatrix} \sum I_x I_t \\ \sum I_y I_t \end{bmatrix}}_{A^T b}}$$

  - The summations are over all pixels in the K x K window

Conditions for solvability

- Optimal (u, v) satisfies Lucas-Kanade equation

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

$$A^T A \qquad\qquad\qquad\qquad A^T b$$

- When is this system solvable?
  - $A^T A$ is recognized to be the structure tensor
  - Invertible, if both eigenvalues are sufficiently larger than zero

The Lucas Kanade (LK) method

- Advantages
  - Only one parameter (window size)
  - Very fast to compute (easily realtime)
  - Can be done dense or sparse
- Disadvantages
  - Each patch is independent, no global consensus
  - The local window assumes a constant motion
  - Sometimes bad results