Chapter 10
Motion Estimation and Optical Flow

Variational Methods for Computer Vision

Winter 2014/15

Prof. Daniel Cremers
Chair for Computer Vision and Pattern Recognition
Department of Computer Science
Technische Universität München

Overview

1. Motion Estimation
2. The Aperture Problem
3. The Optical Flow Constraint
4. Regularity Assumption
5. Lucas and Kanade
6. Horn and Schunck
7. Comparison
8. Limitations of Classical Approaches
9. Brox et al. 2004
10. Wedel et al. 2009
11. Motion Segmentation
12. Scene Flow Estimation

Motion Estimation

The Aperture Problem

The Optical Flow Constraint

Regularity Assumption

Lucas and Kanade

Horn and Schunck

Comparison

Limitations of Classical Approaches

Brox et al. 2004

Wedel et al. 2009

Motion Segmentation

Scene Flow Estimation

Motion Estimation

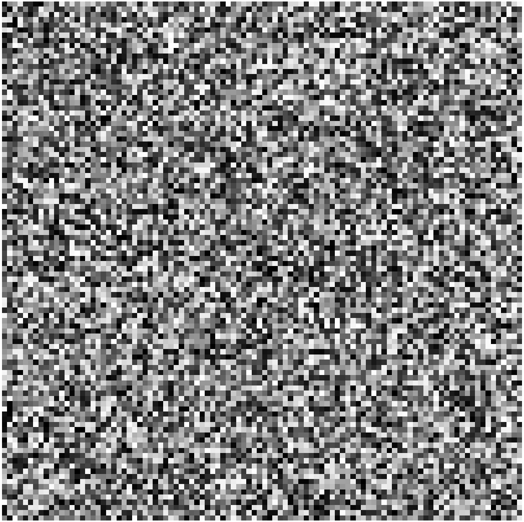
- The estimation of motion fields from image sequences is among the central problems in computer vision.
- With increasing amount of image sequence data – more and more video-capable cameras, higher frame rates, videos on the internet – image sequence analysis is becoming increasingly important.
- Compared to still images, video contains an enormous amount of information about the world surrounding us in the sense that structures can often be distinguished based on their temporal evolution.
- Some applications of motion estimation are already integrated in camera software – panorama generation from several images, video stabilization to remove camera shake, etc.
- Mathematically the problem of motion estimation from images is an ill-posed problem, which means that the question is not sufficiently specified to assure a unique solution.

The Correspondence Problem

Algorithmically, the key challenge in motion estimation is to solve the correspondence problem. Given two images, determine for each point in either image the corresponding partner in the other image. Many computer vision problems are inherently such correspondence problems:

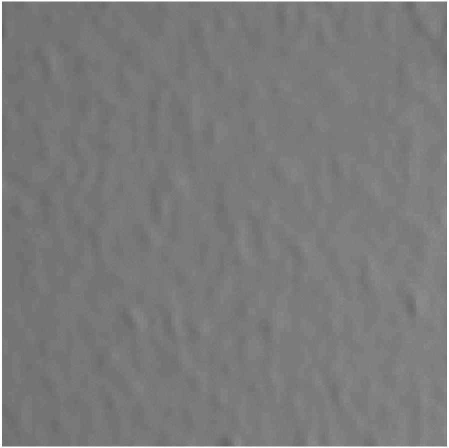
- Disparity estimation from stereo images: Determine a one-dimensional displacement for each pixel to determine the corresponding pixel in the other image. This displacement is inversely proportional to the depth of the respective point.
- Multimodal registration: Given two medical images of an organ acquired with different sensors – for example CT (Computer Tomography) and MRI (Magnetic Resonance Imaging), or CT and PET (Positron-Emission Tomography) – compute an optimal alignment of these images.
- Shape Matching: Given two object shapes (contours in 2D or surfaces in 3D), determine a correspondence between pairs of points from either shape.

Motion and Grouping



Moving regions of random brightness values

Motion and Grouping



Moving wallpaper regions

Motion and Grouping

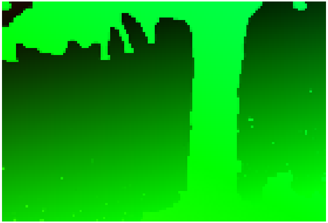
Automatic segmentation of the moving regions.

Cremers, Yuille, DAGM 2003

Motion and 3D Structure

Several images of a static scene filmed by a moving camera.
Foreground objects move faster than background objects.

Motion and 3D Structure



Schoenemann & Cremers,

Near Real-time Motion Segmentation, DAGM 2006.

Motion and Transparency

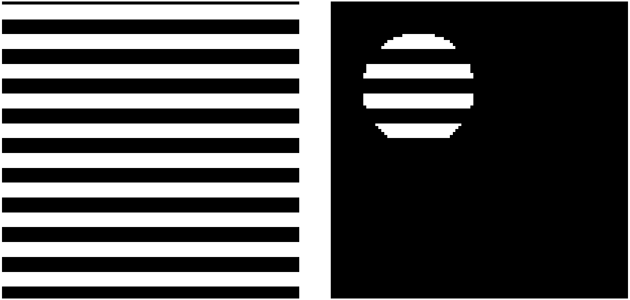
Image sequence showing semi-transparent superposition of two motions.

Author: Michael Black

Applications of Motion Estimation

- Grouping and Segmentation: Motion information allows to identify image regions as objects. This can also be done if semi-transparent structures overlap at a given location.
- Tracking: Using motion information, objects can be tracked in a video sequence.
- Depth estimation: Motion information allows to infer the distance of respective objects from the camera. In principle, one can recover the 3D geometry of the world from an image sequence.
- Time-to-Impact: In the context of driver assistance, motion information allows to make predictions when an obstacle will be hit. As a consequence, one can initiate evasion maneuvers or breaking.
- Video compression: Motion information allows to efficiently compress videos (MPEG encoding).

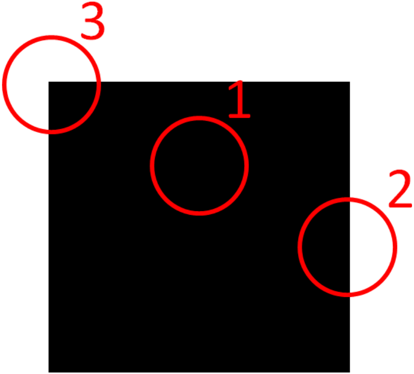
The Aperture Problem



In general, one cannot estimate motion in direction of constant brightness (for example along an image edge). This limitation is referred to as the aperture problem. For example: No matter how the horizontal stripe pattern behind the mask is displaced, we will only observe its vertical motion.

The Aperture Problem: Measurability

Consider three observers each watching a local patch of a moving black box.



Observer 1:	no motion can be observed
Observer 2:	horizontal motion only
Observer 3:	motion in both directions

The Brightness Constancy Assumption

Given an image sequence $I : \Omega \times [0, T] \to \mathbb{R}$, on the image plane $\Omega \subset \mathbb{R}^2$ and the time interval $[0, T]$, we wish to compute a motion field $v : \Omega \times [0, T] \to \mathbb{R}^2$, which assigns to each point $x \in \Omega$ at each time $t \in [0, T]$ a motion vector $v(x, t)$.

Let

$$x : [0, T] \to \Omega$$

denote the trajectory of an object point over time. The classical assumption in motion estimation state that the brightness of a moving point remains constant over time:

$$I(x(t), t) = \text{const.} \quad \forall t \in [0, T]$$

Assuming the brightness function to be differentiable, we can deduce that the total time derivative must vanish:

$$\frac{d}{dt} I(x(t), t) = \nabla I(x(t), t) \frac{dx(t)}{dt} + \frac{\partial I(x(t), t)}{\partial t} = 0 \quad \forall t \in [0, T]$$

The Optic Flow Constraint

The term $v(x, t) = \frac{dx}{dt}$ is nothing but the velocity of the moving point that we are looking for. Thus the assumptions of brightness constancy and differntiability lead to a relation between the desired velocity field $v(x, t)$ and the spatial and temporal image gradients:

$$\boxed{\nabla I^\top v + I_t = 0}$$

This equation is referred to as the differential brigthness constancy constraint or the optic flow constraint.

The optic flow constraint reflects the previously discussed aperture problem: It does not allow statements regarding motion along the level lines of constant intensity. More specifically, let $\tilde{v} = v + \eta$ be a modified motion field with η an arbitrary vector field normal to the image gradient ∇I. The $\tilde{v}$ also fulfills the optic flow constraint:

$$\nabla I^\top \tilde{v} + I_t = \nabla I^\top (v + \eta) + I_t = \nabla I^\top v + I_t = 0.$$

The OFC and the Aperture Problem

The aperture problem is reflected in the optic flow constraint because the constraint is invariant to changes in the motion field which are orthogonal to the local image gradient.

The central problem in motion estimation lies in the fact that the constraint coupling the velocity field $v(x, t)$ and the image gradients cannot be directly solved for v.

More specifically, the flow constraint provides the projection $v_\perp$ of the velocity vector v onto the image gradient ∇I. Dividing the OFC by $|\nabla I|$ leads to:

$$v_\perp \equiv \left(\frac{\nabla I}{|\nabla I|}\right)^\top v = -\frac{I_t}{|\nabla I|}$$

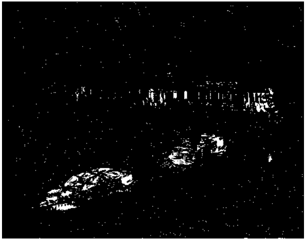
This component of the velocity normal to the level lines is called the normal flow. It is simply given by the negative ratio of temporal and spatial image gradient.

Example: Traffic Scene

$I_1(x)$

$I_2(x)$

$$I_t \approx |I_1(x) - I_2(x)|$$



$$|v_n| = \frac{|I_t|}{|\nabla I|} \approx \frac{|I_2 - I_1|}{\left|\nabla \frac{I_1 + I_2}{2}\right|}$$

Author: Daniel Cremers

Additional Assumptions: Spatial Regularity

The optic flow constraint is necessary but not sufficient to uniquely determine a motion field. It only specifieds the normal component of the velocity field.

In order to eliminate the additional degree of freedom, we therefore need to make additional assumptions.

Two pioneering approaches:

- *Lucas and Kanade 1981*: Assume that the velocity in an entire window around each point is constant. If the window is "sufficiently" large one obtains a unique solution. (over 8800 citations in Jan 2015).
- *Horn and Schunck 1981*: A variational approach to motion estimation based on the assumption of spatial smoothness of the the flow field $v(x, t)$. Extensions to temporal smoothness are straight-forward. (over 9800 citations in Jan 2015). This paper is often considered the first variational method in computer vision.

Lucas and Kanade

For each point $(x, y) \in \Omega \subset \mathbb{R}^2$ and time $t \in [0, T]$, *Lucas und Kanade (1981)* separately determine a motion vector $v(x, y, t)$ by assuming that the motion field is constant in a certain neighborhood $U_\sigma(x, y) \subset \Omega$ around this point.

The motion vector $v = (v_1, v_2)$ is determined in a least squares manner by minimizing the energy:

$$E(v) = \int\limits_{U_\sigma(x,y)} \left(\nabla I^\top v + I_t\right)^2 dx'\, dy' = \int\limits_{U_\sigma(x,y)} \left(I_x v_1 + I_y v_2 + I_t\right)^2 dx'\, dy'$$

The necessary condition for optimality is that the partial derivatives of this energy with respect to the two parameters v_1 and v_2 must vanish:

$$\frac{\partial E(v)}{\partial v_1} = \int\limits_{U_\sigma(x,y)} I_x \left(I_x v_1 + I_y v_2 + I_t\right) dx'\, dy' \overset{!}{=} 0$$

$$\frac{\partial E(v)}{\partial v_2} = \int\limits_{U_\sigma(x,y)} I_y \left(I_x v_1 + I_y v_2 + I_t\right) dx'\, dy' \overset{!}{=} 0$$

Lucas and Kanade

$$\frac{\partial E(v)}{\partial v_1} = \int_{U_\sigma(x,y)} I_x \Big(I_x v_1 + I_y v_2 + I_t \Big) \, dx' \, dy' \stackrel{!}{=} 0$$

$$\frac{\partial E(v)}{\partial v_2} = \int_{U_\sigma(x,y)} I_y \Big(I_x v_1 + I_y v_2 + I_t \Big) \, dx' \, dy' \stackrel{!}{=} 0$$

Since v_1 and v_2 are assumed constant over $U_\sigma(x, y)$ we can extract them from the integral and obtain a linear equation system of the form:

$$Mv = b \qquad \Rightarrow \; v = M^{-1} b$$

where

$$M = \int_{U_\sigma(x,y)} \nabla I \nabla I^\top \, dx' \, dy', \qquad \text{and} \;\; b = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} = -\int_{U_\sigma(x,y)} \nabla I \, I_t \, dx' \, dy'$$

For each point determine v by inversion of a 2×2-matrix.

Lucas and Kanade

In the original approach of Lucas and Kanade all points of the window $U_\sigma(x, y)$ are treated equally. In practice, it is preferable to give more weight to the central pixels. The corresponding cost function is then:

$$E(v) = \int_\Omega G_\sigma(x - x')\left(\nabla I^\top v + I_t\right)^2 dx'\,dy' = G_\sigma * \left(\nabla I^\top v + I_t\right)^2,$$

where the squared optic flow constraint is weighted by some function G_σ (for example a Gaussian kernel). The corresponding linear equation system is given by

$$M_\sigma v = b_\sigma \qquad \text{where}$$

$$M_\sigma = G_\sigma * \left(\nabla I \nabla I^\top\right) = G_\sigma * \begin{pmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{pmatrix}, \text{ and } b_\sigma = -G_\sigma * \left(\nabla I\, I_t\right).$$

The matrix M_σ is called structure tensor.

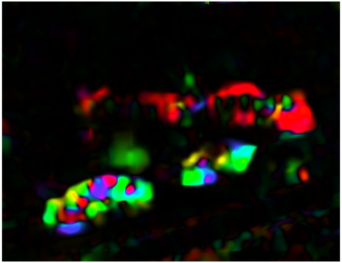
Lucas and Kanade: Solutions?

The goal of Lucas and Kanade was to dermine a unique velocity vector under the assumption of local constancy of the velocity. Depending on the local intensity structure there are three possible cases (see slide 13):

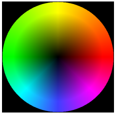
1. The brightness is entirely constant over U_σ, then the gradient ∇I is zero in the neighborhood, the matrix M is 0 and no velocity can be estimated. (Test: trace(M) $< \epsilon$?)
2. All image gradients in the neighborhood U_σ are colinear. Then $\text{rank}(M) = \text{rank}(\nabla I \nabla I^\top) = 1$. The matrix M has only one non-zero eigenvalue. It is not invertible, but one can determine the normal flow: $v_n = -I_t / |\nabla I|$. (Test: det(M) $< \epsilon$?)
3. The gradient ∇I in the window U_σ takes on multiple directions. Then we have $\text{rank}(M) = 2$ or. $\det(M) \neq 0$ and we can determine the velocity vector v by matrix inversion.

Lucas/Kanade: Example

One of two images



Color-coded flow field



Hue encodes direction, brightness encodes magnitude.

Author: Thomas Brox

Horn and Schunck

The approach of *Horn and Schunck (1981)* is considered the first variational approach in computer vision (cf. Snakes: 1988, Mumford-Shah: 1989). In addition to the optic flow constraint for each point, one assumes spatial smoothness of the velocity field $v(x)$:

$$E(v) = \int_{\Omega} \left(\nabla I^{\top} v + I_t \right)^2 dx\, dy \;+\; \lambda \int_{\Omega} |\nabla v(x)|^2 \, dx\, dy.$$

Increasing smoothness of the flow field can be imposed by increasing the weight $\lambda > 0$ of the regularizer. In contrast to standard notation, ∇v does not refer to the divergence of the flow field but to the gradients in each component:

$$|\nabla v(x)|^2 \equiv |\nabla v_1(x)|^2 + |\nabla v_2(x)|^2$$

In contrast to Lucas and Kanade, the approach of Horn and Schunck gives rise to a spatially dense flow field.

Euler-Lagrange Equations

Let $v = (v_1, v_2)$ be the flow field with components v_1 and v_2 in x- and y-direction. The minimizer of the Horn and Schunck functional

$$E(v) = \frac{1}{2} \int_\Omega \left(I_x v_1 + I_y v_2 + I_t\right)^2 + \lambda \left(|\nabla v_1(x)|^2 + |\nabla v_2(x)|^2\right) \, dx \, dy.$$

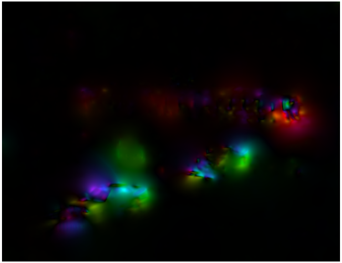
must fulfill the Euler-Lagrange equations:

$$\begin{cases} \dfrac{\partial E}{\partial v_1} = I_x \left(I_x v_1 + I_y v_2 + I_t\right) - \lambda \Delta v_1 = 0 \\ \dfrac{\partial E}{\partial v_2} = I_y \left(I_x v_1 + I_y v_2 + I_t\right) - \lambda \Delta v_2 = 0 \end{cases}$$

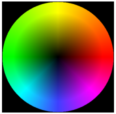
These equations are linear and can be solved with a Gauss-Seidel or Jacobi solver. The regularizer imposes smoothness of the computed flow field. It generates a fill in effect: Components of the velocity field which are not affected by the optic flow constraint are simply adopted from neighboring regions.

Horn/Schunck: Examples

One of two images



Color-coded flow field



Color encodes direction and magnitude

Author: Thomas Brox

Lucas/Kanade vs. Horn/Schunck

Advantages of Lucas/Kanade:

- Fast and simple computation,
- often acceptable and robust results,

Advantages of Horn/Schunck:

- dense flow fields,
- more general: allows for non-translational motion such as rotation,
- strict convexity assures unique solution,
- global fill-in effect, smoothness can be regulated by the parameter λ.
- further extensions: discontinuous flow fields, segmentation,...

Limitations of Both Approaches

- Small motion assumption: The optic flow constraint only holds infinitesimally and thus only applies to small velocity. In general brightness constancy implies:

$$I_1(x) = I_2(x + v(x)).$$

  Linearization (for small v) leads to the optic flow constraint. For larger motions it is no longer valid.

- Brightness constancy: The assumption of brightness constancy is not always valid: Light reflexes on shiny materials, multimodal image registration (where modalities like CT and PET assign different brightness values to the same structure), lighting variations over time, automatic gain control in the camera, etc.

- The approach of Horn and Schunck tends to oversmooth flow fields. In particular, it does not allow for disontinuities in the flow field.

- The above approaches are formulated for two images. In general we have sequences with many images.

Further Advances

Since the pioneering works of Lucas/Kanade and Horn/Schunck a multitude of publications on optic flow estimation have appeared. A paper which integrates a number of advances is *Brox et al., ECCV 2004*:

- Discontinuity-preserving smoothness:

$$\int |\nabla v|^2 dx \;\rightarrow\; \int |\nabla v| dx$$

- Coarse-to-fine warping scheme to allow for larger motion:

$$|\nabla I^\top v + I_t|^2 \;\rightarrow\; |I_1(x) - I_2(x+v)|^2$$

- Robust non-quadratic data terms to allow for outliers:

$$|I_1(x) - I_2(x+v)|^2 \;\rightarrow\; |I_1(x) - I_2(x+v)|$$

- Gradient constancy to account for global brightness changes.

Further Advances

Over the years, the Horn and Schunck approach was modified to the form:

$$E(\boldsymbol{v}) = E_{data}(\boldsymbol{v}) + \alpha\, E_{smooth}(\boldsymbol{v}),$$

mit

$$E_{data}(\boldsymbol{v}) = \int \psi\Big(\underbrace{\big|I(\boldsymbol{x}+\boldsymbol{v}) - I(\boldsymbol{x})\big|^2}_{\text{brightness constancy}} + \gamma\, \underbrace{\big|\nabla I(\boldsymbol{x}+\boldsymbol{v}) - \nabla I(\boldsymbol{x})\big|^2}_{\text{gradient constancy}} \Big)\, d\boldsymbol{x},$$

and

$$E_{smooth}(\boldsymbol{v}) = \int \psi\left(|\nabla_3 u|^2 + |\nabla_3 w|^2\right)\, d\boldsymbol{x},$$

where

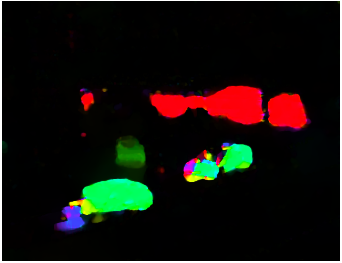
$$\boldsymbol{x} \equiv (x, y, t), \quad \boldsymbol{v} \equiv (u, w, 1), \quad \text{and } \nabla_3 \equiv (\partial_x, \partial_y, \partial_t),$$

and

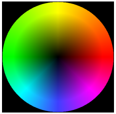
$$\psi(s^2) = \sqrt{s^2 + \epsilon}.$$

Discontinuity-preserving Flow Fields

One of two images



Color coded flow field

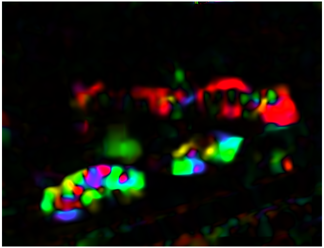


Color encodes motion direction and magnitude

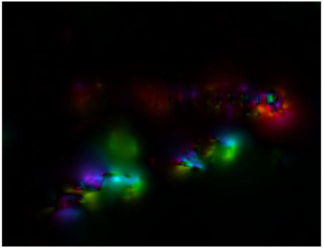
Author: Thomas Brox

Experimental Comparison

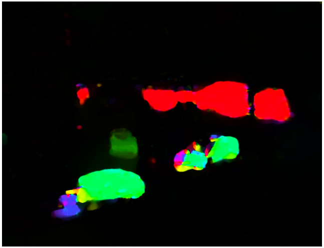
first of two images



Lucas & Kanade ’81



Horn & Schunck ’81



Brox et al. ’04

Author: Thomas Brox

Further Advances

Due to various benchmarks like the Middlebury optical flow benchmark the topic of optical flow estimation has gained renewed interest. Three further improvements contained in the paper *Wedel et al., "Adaptive Regularization...", ICCV 2009* are:

- Quadratic relaxation to decouple data term and regularizer:

$$\min_{v} \int \big| I(x+v) - I(x) \big| + |\nabla v| dx$$

$$\rightarrow \min_{v,u} \int \big| I(x+v) - I(x) \big| + |\nabla u| + \frac{1}{2\theta} |u-v|^2 \, dx$$

- Data-dependent regularization which favors flow edges to coincide with image edges:

$$\int |\nabla v| dx \;\rightarrow\; \int_{\Omega} \exp\left(-\alpha |\nabla I_{\sigma}|^{\alpha}\right) |\nabla v| \, d^2x$$

- Rigid body regularization to impose rigid body motion rather than smoothness.

The Middlebury Benchmark

Optical flow evaluation results

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			Mequon (Hidden texture) GT im0 im1			Schefflera (Hidden texture) GT im0 im1			Wooden (Hidden texture) GT im0 im1			Grove (Synthetic) GT im0 im1			Urban (Synthetic) GT im0 im1			Yosemite (Synthetic) GT im0 im1			Teddy (Stereo) GT im0 im1		
		all	disc	untext	all	disc	untext	all	disc	untext	all	disc	untext	all	disc	untext	all	disc	untext	all	disc	untext	all	disc	untext
Adaptive [20]	4.4	0.09 1	0.26 1	0.06 1	0.23 5	0.78 4	[illegible]	0.54 8	[illegible]	0.21 3	0.18 1	0.91 3	0.10 1	0.88 3	1.25 3	0.73 5	0.50 3	1.28 3	0.31 3	0.14 10	0.16 12	0.22 10	0.65 3	1.37 3	0.79 4
Complementary OF [21]	5.7	0.11 5	0.28 3	0.10 9	0.18 1	0.63	0.12 1	0.31 3	0.75	0.18 1	0.19 2	0.97 5	0.12 3	0.97 10	1.31 6	1.00 11	1.78 20	1.73 7	0.87 14	0.11 4	0.12 2	0.22 10	0.68 4	1.48 4	0.95 8
Aniso. Huber-L1 [22]	5.8	0.10 3	0.28 3	0.08 3	0.31 11	0.88	0.28 12	[illegible]	[illegible]	0.29 12	0.20 4	0.92 4	0.13 5	0.84 2	1.20 2	0.70 2	0.39 1	1.23 1	0.28 1	0.17 15	0.15 9	0.27 16	0.64 2	1.36 2	0.79 4
DPOF [18]	6.1	0.13 12	0.35 12	0.09 4	0.25 6	0.79 5	[illegible]	[illegible]	[illegible]	0.21 3	0.19 2	0.62 1	0.15 11	0.74 1	1.09 1	0.49 1	0.66 7	1.80 10	0.63 8	0.19 17	0.17 14	0.35 20	0.50 1	1.08 1	0.55 1
TV-L1-improved [17]	7.2	0.09 1	0.26 1	0.07 2	0.20 3	0.71 3	0.16 2	0.53 7	1.18 9	0.22 5	0.21 7	1.24 11	0.11 2	0.90 4	1.31 6	0.72 3	1.51 14	1.93 11	0.84 11	0.18 16	0.17 14	0.31 17	0.73 8	1.62 9	0.87 7
CBF [12]	7.8	0.10 3	0.28 3	0.09 4	0.34 12	0.80 6	0.37 13	0.43 5	0.95 5	0.26 8	0.21 7	1.14 8	0.13 5	0.90 4	1.27 4	0.82 7	0.41 2	1.23 1	0.30 2	0.23 22	0.19 20	0.39 21	0.76 9	1.56 6	1.02 9
Brox et al. [5]	8.4	0.11 5	0.32 8	0.11 12	0.27 9	0.93 10	0.22 9	0.39 4	0.94 4	0.24 7	0.24 9	1.25 12	0.13 5	1.10 13	1.39 12	1.43 17	0.89 8	1.77 8	0.55 7	0.10 2	0.13 4	0.11 1	0.91 11	1.83 12	1.13 12
Rannacher [23]	8.5	0.11 5	0.31 6	0.09 4	0.25 6	0.84 7	0.21 8	0.57 12	1.27 15	0.26 8	0.24 9	1.32 14	0.13 5	0.91 7	1.33 8	0.72 3	1.49 13	1.95 13	0.78 9	0.15 12	0.14 7	0.26 13	0.69 6	1.58 8	0.86 6
F-TV-L1 [15]	8.8	0.14 13	0.35 12	0.14 15	0.34 12	0.98 12	0.26 11	0.59 14	1.19 10	0.26 8	0.27 13	1.36 15	0.16 12	0.90 4	1.30 5	0.76 6	0.54 4	1.62 6	0.36 4	0.13 6	0.15 9	0.20 9	0.68 4	1.56 6	0.66 2
Second-order prior [8]	9.0	0.11 5	0.31 6	0.09 4	0.26 8	0.93 10	0.20 7	0.57 12	1.25 14	0.26 8	0.20 4	1.04 6	0.12 3	0.94 8	1.34 9	0.83 9	0.61 6	1.93 11	0.47 6	0.20 18	0.16 12	0.34 19	0.77 10	1.64 10	1.07 10
Fusion [6]	9.4	0.11 5	0.34 10	0.10 9	0.19 2	0.69 2	0.16 2	0.29 2	0.66 2	0.23 6	0.20 4	1.19 10	0.14 9	1.07 11	1.42 13	1.22 13	1.35 10	1.49 5	0.86 13	0.20 18	0.20 21	0.26 13	1.07 14	2.07 16	1.39 16
Dynamic MRF [7]	11.1	0.12 11	0.34 10	0.11 12	0.22 4	0.89 9	0.16 2	0.44 6	1.13 7	0.20 2	0.24 9	1.29 13	0.14 9	1.11 14	1.52 17	1.13 12	1.54 15	2.37 20	0.93 15	0.13 6	0.12 2	0.31 17	1.27 18	2.33 20	1.66 17
SegOF [10]	11.7	0.15 14	0.36 14	0.10 9	0.57 15	1.16 15	0.59 19	0.68 15	1.24 12	0.64 14	0.32 15	0.86 2	0.26 15	1.18 17	1.50 16	1.47 18	1.63 18	2.09 14	0.96 16	0.08 1	0.13 4	0.12 2	0.70 7	1.50 5	0.69 3
Learning Flow [11]	13.3	0.11 5	0.32 8	0.09 4	0.29 10	0.99 13	0.23 10	0.55 9	1.24 12	0.29 12	0.36 16	1.56 17	0.25 14	1.25 19	1.64 21	1.41 16	1.55 17	2.32 19	0.85 12	0.14 10	0.18 18	0.24 12	1.09 15	2.09 18	1.27 13
Filter Flow [19]	14.3	0.17 16	0.39 16	0.13 14	0.43 14	1.09 14	0.38 14	0.75 16	1.34 16	0.78 19	0.70 19	1.54 16	0.68 19	1.13 16	1.38 11	1.51 19	0.57 5	1.32 4	0.44 5	0.22 20	0.23 23	0.26 13	0.96 12	1.66 11	1.12 11
GraphCuts [14]	14.5	0.16 15	0.38 15	0.14 15	0.59 18	1.36 19	0.46 15	0.56 10	1.07 6	0.64 14	0.26 12	1.14 8	0.17 13	0.96 9	1.35 10	0.84 10	2.25 23	1.79 9	1.22 21	0.22 20	0.17 14	0.43 22	1.22 17	2.05 15	1.78 19
Black & Anandan [4]	15.0	0.18 17	0.42 17	0.19 18	0.58 17	1.31 17	0.50 16	0.95 19	1.58 18	0.70 16	0.49 17	1.59 18	0.45 17	1.08 12	1.42 13	1.22 13	1.43 11	2.28 17	0.83 10	0.15 12	0.17 14	0.17 6	1.11 16	1.98 14	1.30 14
SPSA-learn [13]	15.7	0.18 17	0.45 18	0.17 17	0.57 15	1.32 18	0.51 17	0.84 17	1.50 17	0.72 17	0.52 18	1.64 19	0.49 18	1.12 15	1.42 13	1.39 15	1.75 19	2.14 15	1.06 20	0.13 6	0.13 4	0.19 7	1.32 19	2.08 17	1.73 18
GroupFlow [9]	15.9	0.21 19	0.51 19	0.21 19	0.79 21	1.69 21	0.72 21	0.86 18	1.64 19	0.74 18	0.30 14	1.07 7	0.26 15	1.29 22	1.81 22	0.82 7	1.94 21	2.30 18	1.36 22	0.11 4	0.14 7	0.19 7	1.06 13	1.96 13	1.35 15
2D-CLG [1]	17.4	0.28 21	0.62 22	0.21 19	0.67 20	1.21 16	0.70 20	1.12 21	1.80 21	0.99 22	1.07 22	2.06 21	1.12 22	1.23 18	1.52 17	1.62 22	1.54 15	2.15 16	0.96 16	0.10 2	0.11 1	0.16 4	1.38 20	2.26 19	1.83 20
Horn & Schunck [3]	18.6	0.22 20	0.55 20	0.22 21	0.61 19	1.53 20	0.52 18	1.01 20	1.73 20	0.80 20	0.78 20	2.02 20	0.77 20	1.26 20	1.58 19	1.55 20	1.43 11	2.59 22	1.00 18	0.16 14	0.18 18	0.15 3	1.51 21	2.50 21	1.88 21
TI-DOFE [24]	19.6	0.38 23	0.64 23	0.47 23	1.16 22	1.72 22	1.26 22	1.39 23	2.06 24	1.17 23	1.29 23	2.21 23	1.41 23	1.27 21	1.61 20	1.57 21	1.28 9	2.57 21	1.01 19	0.13 6	0.15 9	0.16 4	1.87 22	2.71 22	2.53 22
FOLKI [16]	22.6	0.29 22	0.73 24	0.33 22	1.52 23	1.96 24	1.80 23	1.23 22	2.04 23	0.95 21	0.99 21	2.20 22	1.08 21	1.53 23	1.85 23	2.07 23	2.14 22	3.23 24	1.60 23	0.26 23	0.21 22	0.68 23	2.67 23	3.27 23	4.32 23
Pyramid LK [2]	23.7	0.39 24	0.61 21	0.61 24	1.67 24	1.78 23	2.00 24	1.50 24	1.97 22	1.38 24	1.57 24	2.39 24	1.78 24	2.94 24	3.72 24	2.98 24	3.33 24	2.74 23	2.43 24	0.30 24	0.24 24	0.73 24	3.80 24	5.08 24	4.88 24

Move the mouse over the numbers in the table to see the corresponding images. Click to compare with the ground truth.

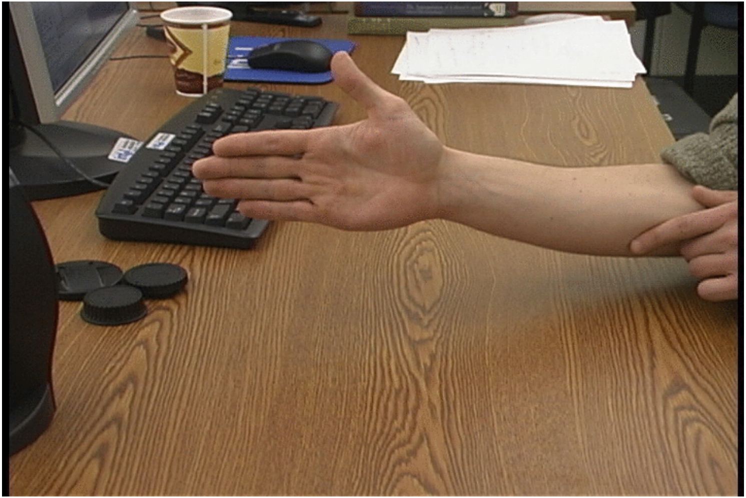
Source: Baker et al., "A Database and Evaluation Methodology for Optical Flow", IJCV 2011.

The Middlebury Benchmark

Algorithm	Data term				Prior term						Optimization					Misc.		
	L1 norm	Other robust penalty Fn	Gradient/other features	Illum. modeling/norm/color	L1/TV norm	Other robust penalty Fn	Spatial weighting	Anisotropic weighting	Higher-order prior	Rigidity prior	Cont.-gradient descent	Cont.-variational/extremal	Cont.-other	Discr.-fusion	Discr.-reparameterization	Learning	Visibility/occlusion	Color
Adaptive (Wedel et al. 2009)	X			X	X		X	X		X			X					
Complementary OF (Zimmer et al. 2009)	X		X	X		X	X	X				X						X
Aniso. Huber-L1 (Werlberger et al. 2009)	X			X		X	X	X					X					
DPOF (Lei and Yang 2009)				X	X		X				X				X		X	X
TV-L1-improved (Wedel et al. 2008)	X			X	X								X					
CBF (Trobin et al. 2008)	X				X				X					X				X
Brox et al. (Brox et al. 2004)	X		X		X							X						X
F-TV-L1 (Wedel et al. 2008)	X				X					X			X					
Second-order prior (Trobin et al. 2008)	X								X				X					
Fusion (Lempitsky et al. 2008)		X		X		X	X				X			X				X
Dynamic MRF (Glocker et al. 2008)	X				X										X			
Seg OF (Xu et al. 2008)	X				X		X					X					X	X
Learning Flow (Sun et al. 2008)		X	X			X		X				X				X		
Filter Flow (Seitz and Baker 2009)	X			X	X		X		X				X					X
Graph Cuts (Cooke 2008)	X				X										X			X
Black & Anandan (Black and Anandan 1996)		X				X					X							
SPSA-learn (Li and Huttenlocher 2008)		X				X					X					X		X
Horn & Schunck (Horn and Schunck 1981)												X						

Source: Baker et al., IJCV 2011

Motion Segmentation



Hand moving toward camera

Motion Segmentation

The estimated motion is typically not the final goal of image analysis. In a real-world video one may be interested in segmenting the differently moving objects.

Motion-based segmentation is a chicken-and-egg problem: To reliably estimate motion, we need a certain support-area (ideally the entire region which moves coherently). Yet, in order to partition the image plane into coherently moving region, we need to know the motion at each pixel.

In *Cremers, Soatto, IJCV 2005*, we tackled this chicken-and-egg problem, proposing a method to jointly estimate a segmentation and motion vectors v_i associated with each region Ω_i:

$$E(\Omega_1, \ldots, \Omega_n, v_1, \ldots, v_n) = \sum_{i=1}^{n} \int_{\Omega_i} |\nabla I^\top v_i + I_t|^2 \, dx \; + \; \tfrac{\nu}{2} |\partial \Omega_i|.$$

This can be seen as a variation of the Mumford-Shah model, where rather than estimating the average brightness of each region we estimate its average motion.

Piecewise Parametric Motion Segmentation

While the above model allows to separate differently translating regions, in many real world scenarios objects undergo rigid motion giving rise to rotational or zooming flow fields.

The above model can be extended to allow a parametric motion for each region:

$$v_i(x) = S(x)\, p_i = \begin{pmatrix} x & y & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & x & y & 1 \end{pmatrix} (a_i \ b_i \ c_i \ d_i \ e_i \ f_i)^\top,$$

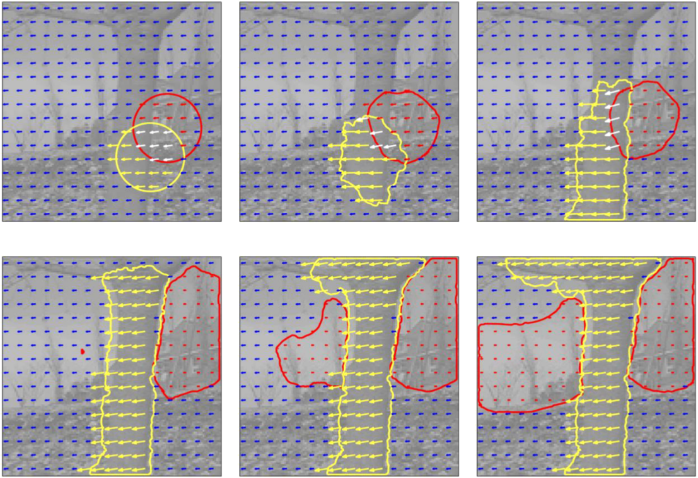
with a vector $p_i \in \mathbb{R}^6$ defining an affine motion for region Ω_i.

The variational approach

$$E(\Omega_1, \dots, \Omega_n, p_1, \dots, p_n) = \sum_{i=1}^{n} \int_{\Omega_i} |\nabla I^\top S(x)\, p_i + I_t|^2 \, dx + \tfrac{\nu}{2} |\partial \Omega_i|.$$

leads to a piecewise parametric motion field. In the two-region case, this can be solved at around 30 fps – see *Schoenemann, Cremers, DAGM 2006*.

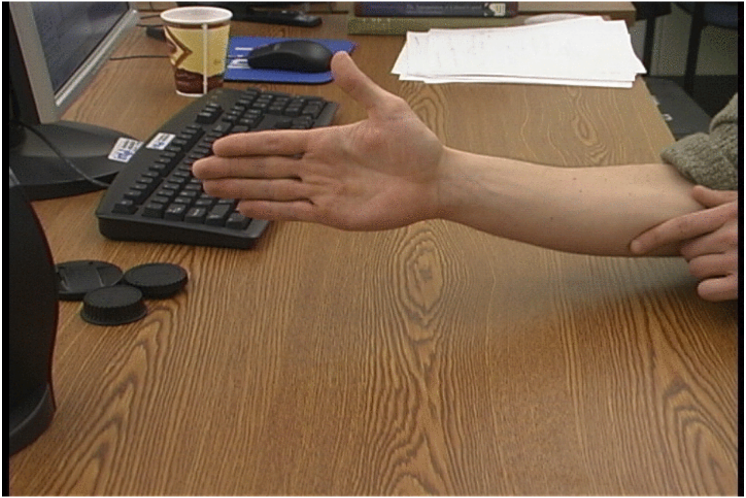
Piecewise Parametric Motion Segmentation



Motion-based segmentation into depth layers.

Cremers, Soatto, IJCV 2005.

Piecewise Parametric Motion Segmentation



Hand moving toward camera

Cremers, Soatto, IJCV 2005.

Piecewise Parametric Motion Segmentation

Piecewise parametric motion segmentation with level sets

Cremers, Soatto, IJCV 2005.

Motion in 3D: Scene Flow

The optical flow gives the motion in the image plane, sometimes referred to as the apparent motion. Based on stereo video one can jointly estimate depth maps and a dense 3D motion field called scene flow.

Let $I(x, y, t)^\ell$ and $I(x, y, t)^r$ be the left and right images at pixel (x, y) and time t and d the stereo disparity for that pixel. *Wedel et al. '08* make several constancy assumptions:

$$\begin{aligned} I(x, y, t)^\ell &= I(x + u, y + v, t + 1)^\ell \\ I(x + d, y, t)^r &= I(x + d + d' + u, y + v, t + 1)^r, \end{aligned}$$

where d' denotes the change in disparity (motion in z-direction). Enforcing consistency of the left and right images at time $t + 1$ leads to:

$$I(x + u, y + v, t + 1)^\ell - I(x + d + d' + u, y + v, t + 1)^r = 0.$$

Variational Scene Flow Estimation

Wedel et al. '08 combine these assumptions with a smoothness prior for variational scene flow estimation:

$$E(u, v, d') = E_{\text{data}} + E_{\text{smooth}},$$

with

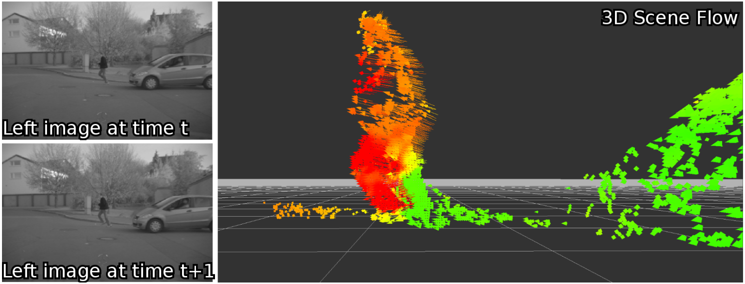
$$E_{\text{data}} = \int_\Omega \left| I(x+u, y+v, t+1)^\ell - I(x, y, t)^\ell \right| dxdy$$
$$+ \int_\Omega c(x,y) \left| I(x_d + d' + u, y+v, t+1)^r - I(x_d, y, t)^r \right| dxdy$$
$$+ \int_\Omega c(x,y) \left| I(x_d + d' + u, y+v, t+1)^r - I(x+u, y+v, t+1)^\ell \right| dxdy,$$

and

$$E_{\text{smooth}} = \int_\Omega \sqrt{\lambda|\nabla u|^2 + \lambda|\nabla v|^2 + \gamma|\nabla d'|^2}\, dxdy,$$

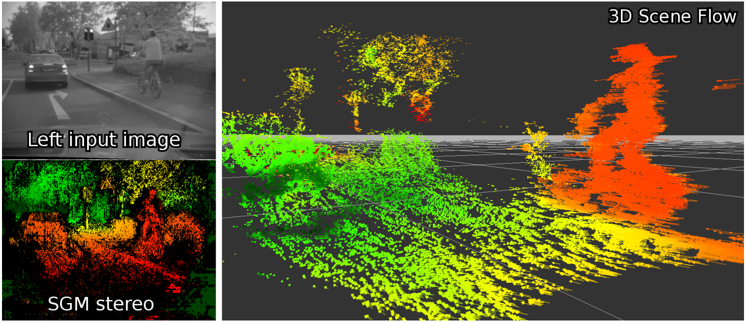
where $c: \Omega \to \{0, 1\}$, $c(x, y) = 0$ if there is no disparity at (x, y).

Motion in 3D: Scene Flow



Wedel, Brox, Vaudrey, Rabe, Franke, Cremers, "Stereoscopic Scene Flow Computation for 3D Motion Understanding", Int. J. of Computer Vision 2011.

Motion in 3D: Scene Flow



Wedel, Brox, Vaudrey, Rabe, Franke, Cremers, "Stereoscopic Scene Flow Computation for 3D Motion Understanding", Int. J. of Computer Vision 2011.

Motion in 3D: Scene Flow

Wedel, Brox, Vaudrey, Rabe, Franke, Cremers, "Stereoscopic Scene Flow Computation for 3D Motion Understanding", Int. J. of Computer Vision 2011.

Motion in 3D: Scene Flow