Graphical Models in Computer Vision

Andreas Geiger

Max Planck Institute for Intelligent Systems
Perceiving Systems

July 11, 2016

MAX-PLANCK-GESELLSCHAFT

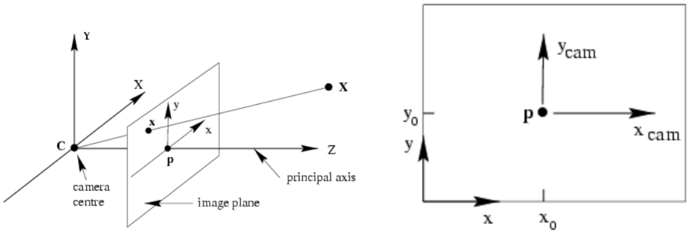
Syllabus

11.04.2016	Introduction
18.04.2016	Graphical Models 1
25.04.2016	Graphical Models 2 (Sand 6/7)
02.05.2016	Graphical Models 3
09.05.2016	Graphical Models 4
23.05.2016	Body Models 1
30.05.2016	Body Models 2
06.06.2016	Body Models 3
13.06.2016	Body Models 4
20.06.2016	Object Detection 1
27.06.2016	Object Detection 2
04.07.2016	Stereo
11.07.2016	Optical Flow
18.07.2016	Segmentation

Todays topic

- Recap
  - Pinhole Camera
  - Epipolar Geometry
  - Block Matching Stereo
  - Stereo Matching with Spatial Regularization
- Optical Flow
  - Motivation
  - Optical Flow Constraint Equation
  - Local Optical Flow Methods
  - Global Optical Flow Methods

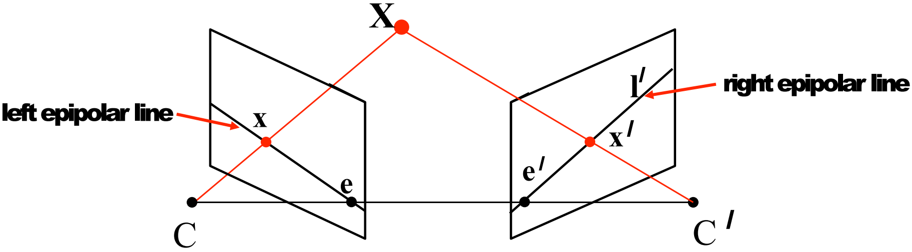
Pinhole Camera Model



- Principal point $\mathbf{p} = (c_x, c_y)^T$: Point where the principal axis intersects the image plane (origin of normalized coordinate system)

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} \mapsto \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} f\,X/Z + c_x \\ f\,Y/Z + c_y \end{pmatrix} \Rightarrow \underbrace{\begin{pmatrix} f\,X \\ f\,Y \\ Z \end{pmatrix}}_{\mathbf{x}\in\mathbb{R}^3} = \underbrace{\begin{pmatrix} f & 0 & c_x & 0 \\ 0 & f & c_y & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}}_{\mathbf{P}\in\mathbb{R}^{3\times 4}} \cdot \underbrace{\begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}}_{\mathbf{X}\in\mathbb{R}^4}$$

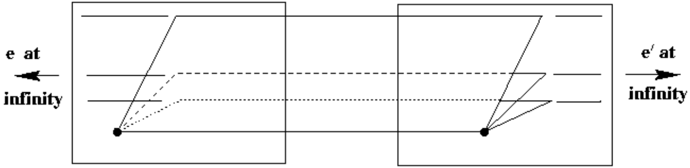
Epipolar Geometry

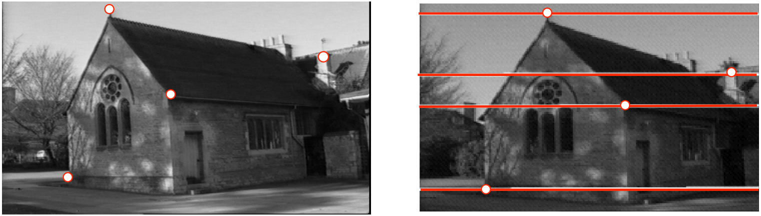


- $\overline{\mathbf{CC'}}$: Baseline (translation between cameras)
- $\mathbf{e}, \mathbf{e}'$: Epipole (intersection of image plane with baseline)
- $\mathbf{l}, \mathbf{l}'$: Epipolar line (intersection of image plane with epipolar plane)

Epipolar Geometry

What if both cameras face the same direction?





- Epipoles are at infinity, epipolar lines are parallel
- Correspondences along "scanlines" (simplifies computation)

Epipolar Geometry

How to recover a 3D point from two corresponding image points?

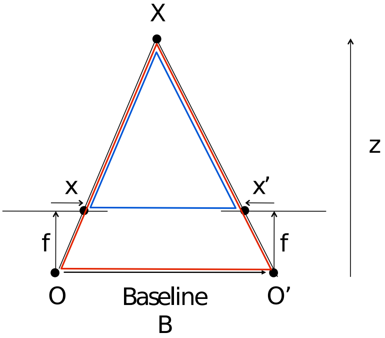
- Equal triangles (only when image planes are parallel)
- Using the definition $d = x - x'$:

$$\frac{Z - f}{B - (x - x')} = \frac{Z}{B}$$

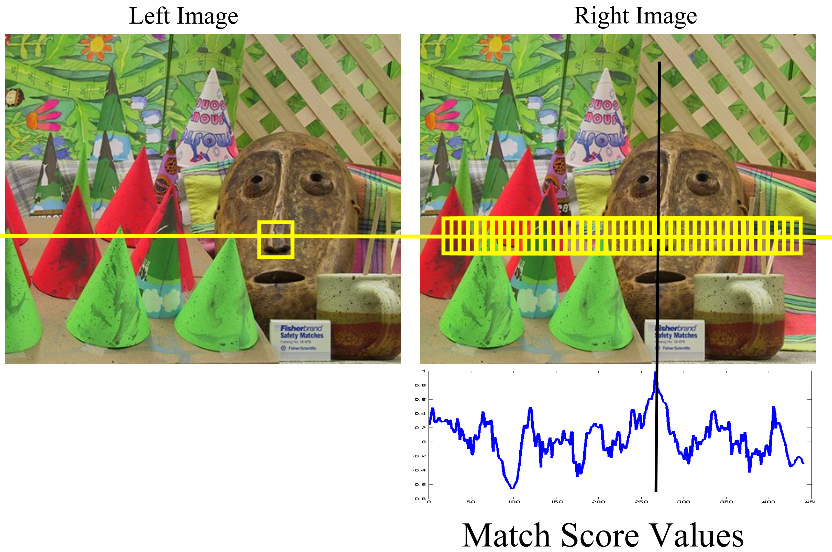
$$ZB - fB = ZB - Z(x - x')$$

$$Z = \frac{fB}{x - x'} = \frac{fB}{d}$$

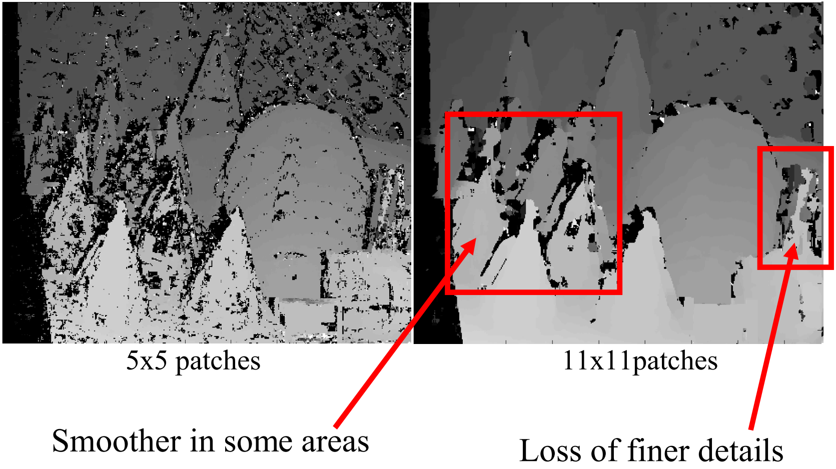
$$d = \frac{fB}{Z}$$



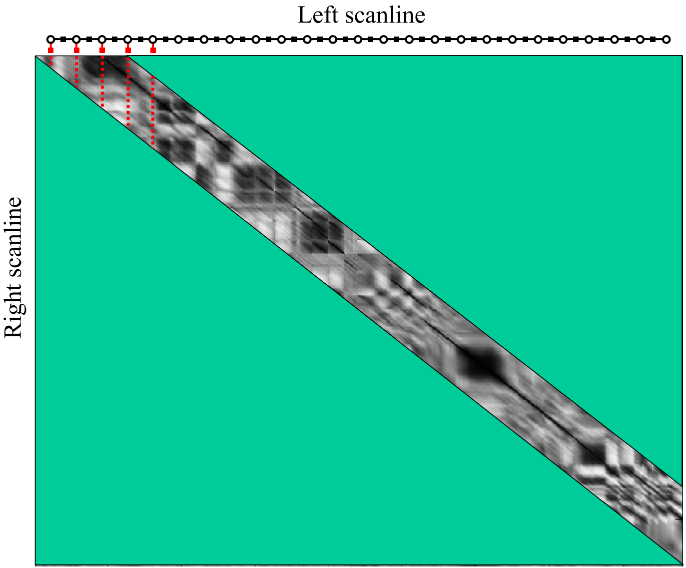
Block Matching



Block Matching



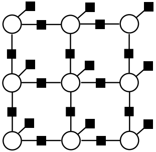
Spatial Regularization (1D)



Spatial Regularization (2D)

- What can we do to preserve inter-scanline consistency?
- Specify a loopy MRF on a grid instead of a chain MRF on individual scanlines and solve for the whole disparity map at once!

$$p(\mathbf{D}) \propto \exp\left\{ -\sum_{i} \psi_{data}(d_i) - \lambda \sum_{i \sim j} \psi_{smooth}(d_i, d_j) \right\}$$

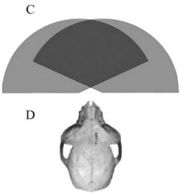


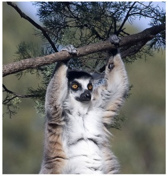
- Disparity image: $\mathbf{D}$
- $i \sim j$ indicates neighboring pixels on a 4-connected grid
- Unary terms: Matching cost $\psi_{data}(d)$
- Pairwise terms: Smoothness between adjacent pixels, e.g.:
  - Potts: $\psi_{smooth}(d, d') = [d \neq d']$
  - Truncated l_1: $\psi_{smooth}(d, d') = \min(|d - d'|, \tau)$
  - Truncated l_2: $\psi_{smooth}(d, d') = \min((d - d')^2, \tau)$
- Solve MRF approximately using max-product BP, graph cuts, etc.

Stereo vs. Optical Flow

Stereo

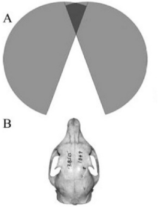
- 2 images at same time
- Only camera motion
- 1D estimation problem
- Monkeys



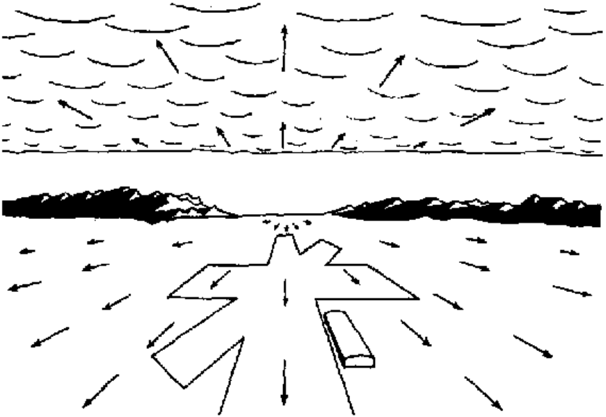


Optical Flow

- 2 images at 2 time steps
- Camera and object motion
- 2D estimation problem
- Squirrels

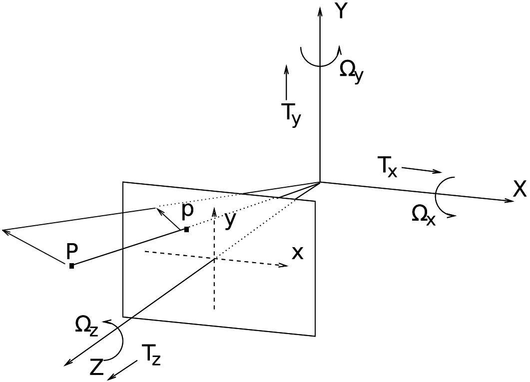


Optical Flow



[J. J. Gibson, 1950: The Ecological Approach to Visual Perception]

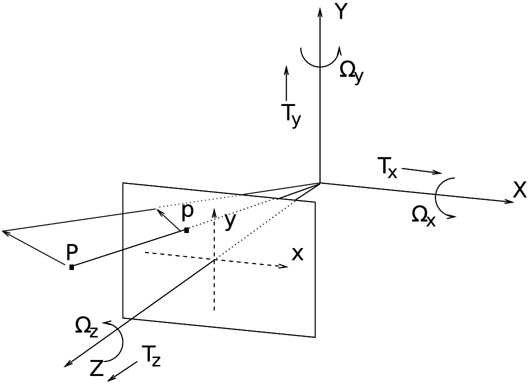
Optical Flow



Motion field:

- 2D motion field representing the **projection of the 3D motion** of points in the scene onto the image plane
- Can be the result of camera motion or object motion (or both)!

Optical Flow

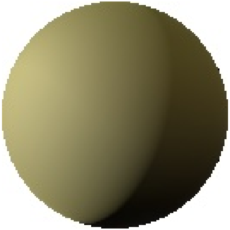


Optical flow:

- 2D velocity field describing the **apparent motion** in the image (*i.e.*, the displacement of pixels looking "similar")
- Optical flow $\neq$ motion field! Why?

Thought Experiment

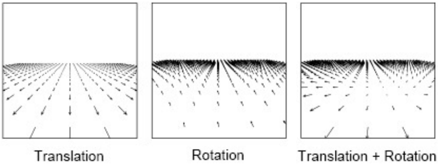
- Lambertian ball **rotating in 3D**
- What does the 2D **motion field** look like?
- What does the 2D **optical flow field** look like?



Thought Experiment

- Stationary specular ball **moving light source**
- What does the 2D **motion field** look like?
- What does the 2D **optical flow field** look like?

Optical Flow Field



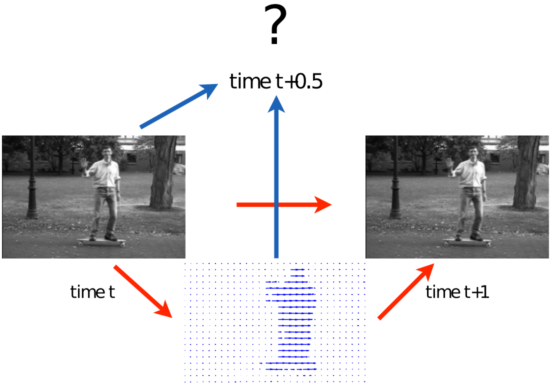
Optical flow fields tell us something (maybe ambiguous) about:

- The **3D structure** of the world
- The **motion of objects** in the viewing area
- The **motion of the observer** (if any)

In contrast to stereo:

- Calculated from images captured at 2 different time instances
- No epipolar geometry $\Rightarrow$ 2D estimation problem!

Applications: Video Interpolation / Frame Rate Adaption

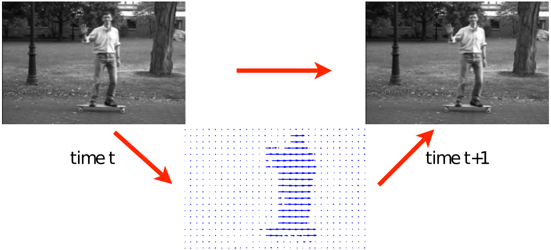


- If we know the image motion we can compute images at intermediate time steps

Applications: Video Interpolation / Frame Rate Adaption

- If we know the image motion we can compute images at intermediate time steps

Applications: Video Compression



- To compress an image sequence, we can predict new frames using the optical flow field and only store how to “fix” the prediction
- Flow fields are smooth, thus easier to compress/store than images!

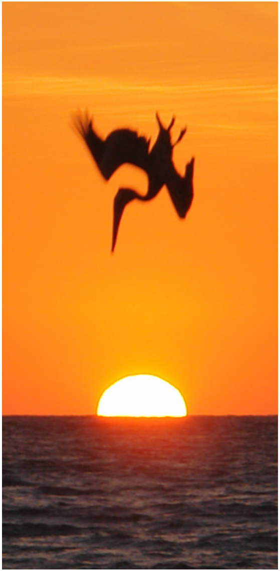
Applications: Autonomous Driving

The Northern Gannet

The Northern Gannet

The Northern Gannet

The Northern Gannet



Assumptions

Optical flow is an ill-posed problem. We will make 3 important assumptions and then exploit them in our algorithms:

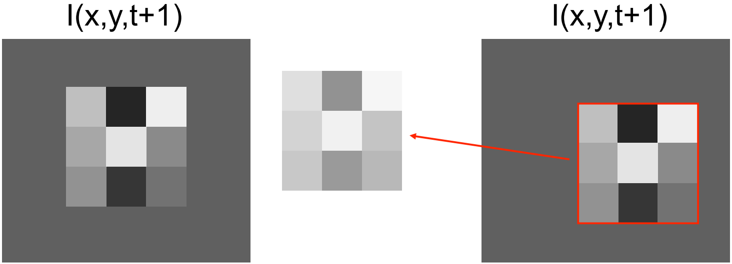
- **Brightness constancy:** Brightness will remain the same even though location might have changed:

$$I(x + u(x, y), y + v(x, y), t + 1) = I(x, y, t)$$

  - $I(x, y, t)$: intensity of image taken at time t at pixel (x, y)
  - $u(x, y)$: horizontal flow at (x, y)
  - $v(x, y)$: vertical flow at (x, y)

- **Spatial coherence:** Neighboring points in the scene typically belong to the same surface and hence have similar 2D/3D motions
- **Temporal coherence:** The image motion of a surface patch changes only gradually over time (not in this lecture)

A simple Optical Flow Algorithm



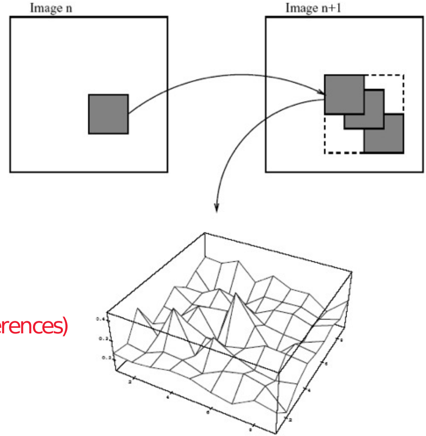
- Minimize brightness difference (SSD) with respect to flow (u, v):

$$E_{SSD}(u, v) = \sum_{(x,y) \in R} (I(x+u, y+v, t+1) - I(x, y, t))^2$$

- $I(x, y, t)$: Image intensity at pixel x, y and time t
- R: Small window region in the image
- u, v: Optical flow of window region R

A simple Optical Flow Algorithm

$$E_{SSD}(u, v) = \sum_{(x,y) \in R} (I(x+u, y+v, t+1) - I(x, y, t))^2$$



SSD surface
(sum of squared differences)

A simple Optical Flow Algorithm

$$E_{SSD}(u, v) = \sum_{(x,y)\in R} (I(x + u, y + v, t + 1) - I(x, y, t))^2$$

Simple optical flow algorithm:

- Discretize the space of possible motions
- For each pixel, try all possible motions (u, v) in a neighborhood region
- Select the one that minimizes SSD (WTA)

Problems:

- 2D search range (compared to 1D for stereo)
- Very inefficient
- In practice, motions are continuous (not discrete)

Can we instead optimize this non-convex objective function directly?

A simple Optical Flow Algorithm

$$E_{SSD}(u,v) = \sum_{(x,y)\in R} \left(I(x+u, y+v, t+1) - I(x,y,t)\right)^2$$

- Let us look at the general case:

$$I(x+\Delta_x, y+\Delta_y, t+\Delta_t)$$

- We can approximate this expression using Taylor series expansion:

$$\begin{aligned} & I(x+\Delta_x, y+\Delta_y, t+\Delta_t) \\ = \quad & I(x,y,t) \\ + \quad & \Delta_x \frac{\partial}{\partial x} I(x,y,t) + \Delta_y \frac{\partial}{\partial y} I(x,y,t) + \Delta_t \frac{\partial}{\partial t} I(x,y,t) \\ + \quad & \underbrace{\epsilon(\Delta_x^2, \Delta_y^2, \Delta_t^2)}_{\text{approximation error}} \end{aligned}$$

A simple Optical Flow Algorithm

$$E_{SSD}(u,v) = \sum_{(x,y)\in R} (I(x+u, y+v, t+1) - I(x,y,t))^2$$

- Thus we can (first-order) approximate $E_{SSD}(u,v)$ as follows:

$$\begin{aligned} E_{SSD}(u,v) &\approx \sum_{(x,y)\in R} \left(u\frac{\partial}{\partial x} I(x,y,t) + v\frac{\partial}{\partial y} I(x,y,t) + \frac{\partial}{\partial t} I(x,y,t) \right)^2 \\ &= \sum_{(x,y)\in R} (u \cdot I_x(x,y,t) + v \cdot I_y(x,y,t) + I_t(x,y,t))^2 \end{aligned}$$

- This approximated SSD objective is convex in the flow! (Why?)
- Can be easily solved in close form (Why?)
- But: holds only for small motions (due to linear approximation)

Optical Flow Constraint Equation

$$E_{SSD}(u,v) \approx \sum_{(x,y)\in R} (u \cdot I_x(x,y,t) + v \cdot I_y(x,y,t) + I_t(x,y,t))^2$$

- By minimizing this Taylor series approximation to the SSD, we are trying to enforce the so-called **optical flow constraint equation** (OFCE) at every pixel:

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

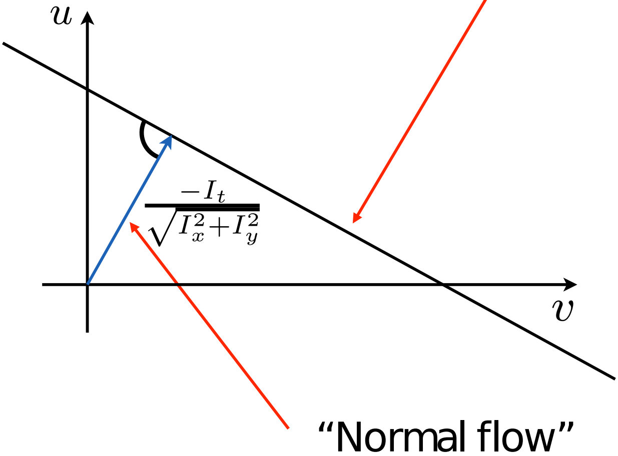
- This is also called the linearized brightness constancy constraint
- Let us simplify the notation a bit:

$$\nabla I^T \mathbf{u} = -I_t \quad \text{with} \quad \nabla I = \begin{pmatrix} I_x \\ I_y \end{pmatrix} \quad \text{and} \quad \mathbf{u} = \begin{pmatrix} u \\ v \end{pmatrix}$$

- How does this constraint look like?

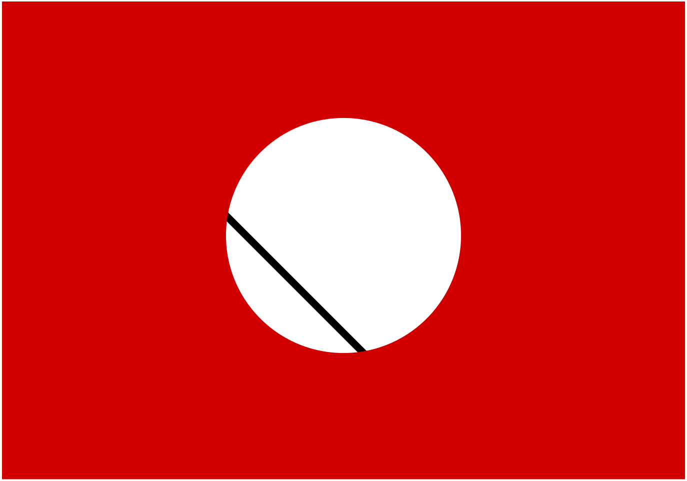
Optical Flow Constraint Equation

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



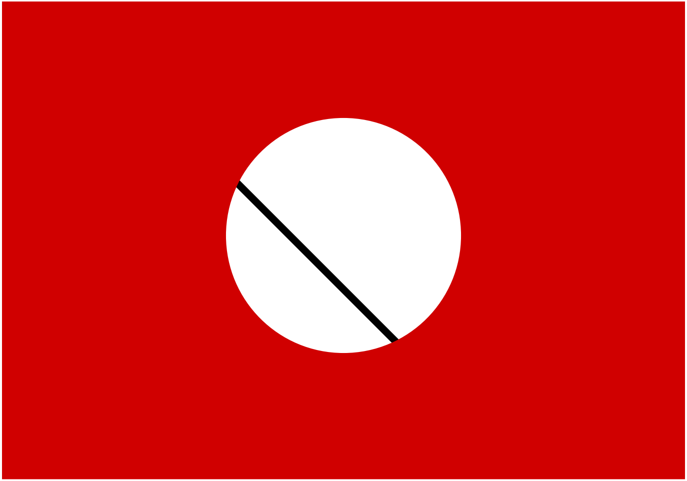
Aperture Problem

In which direction does the line move?



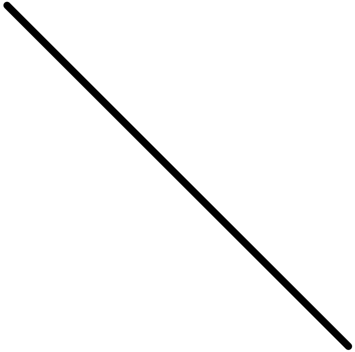
Aperture Problem

In which direction does the line move?



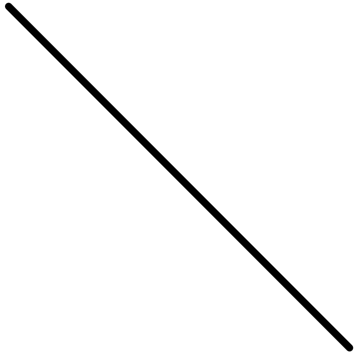
Aperture Problem

Now the full picture ...



Aperture Problem

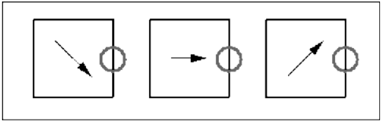
Now the full picture ...

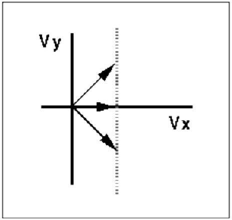


Aperture Problem

$$u \cdot I_x + v \cdot I_y = -I_t$$
$$u \cdot I_x + 0 \cdot I_y = -I_t$$
$$u \cdot I_x = -I_t$$
$$u = -\frac{I_t}{I_x}$$

- 1 equation, 2 unknowns
- v could be anything!
- We can only measure the normal velocity


a



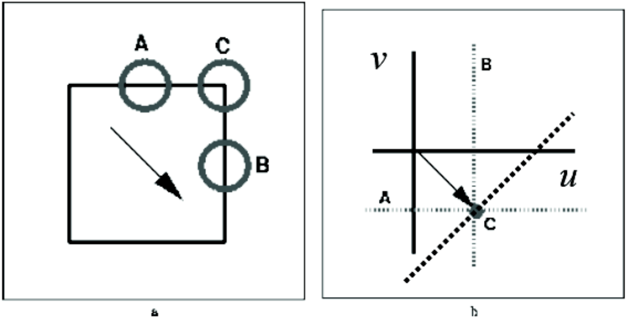
b

Aperture Problem

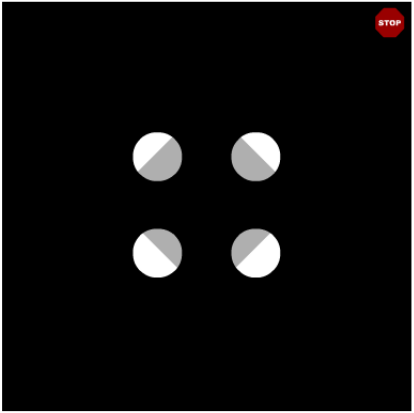
- Barber Pole: What is the motion field? What is the optic flow field?

Aperture Problem

- The optical flow field depends on the geometry of the aperture and the orientation and frequency of the line pattern!
- In practice, we combine several constraints to get an estimate of the full velocity vector (not only the normal component):



Aperture Problem



a one edge

b whole square

c a corner

d four edges

Source: McDermott, J., Weiss, Y., Adelson, E.H. (2001). Beyond junctions: Nonlocal form contraints on motion interpretation. Perception, 30: 905-923.

The motion of an edge seen through an aperture, as in (a) is inherently ambiguous. In this case the edge is physically moving upwards, as can be seen in (b), but the edge motion alone is consistent with many other possible motions, and in this case the edge in (a) generally appears to move diagonally. This ambiguity is known as the aperture problem. The upshot is that local motion measurements, such as those made by neurons in the early stages of the visual system, often do not specify the direction of motion of objects in the world.

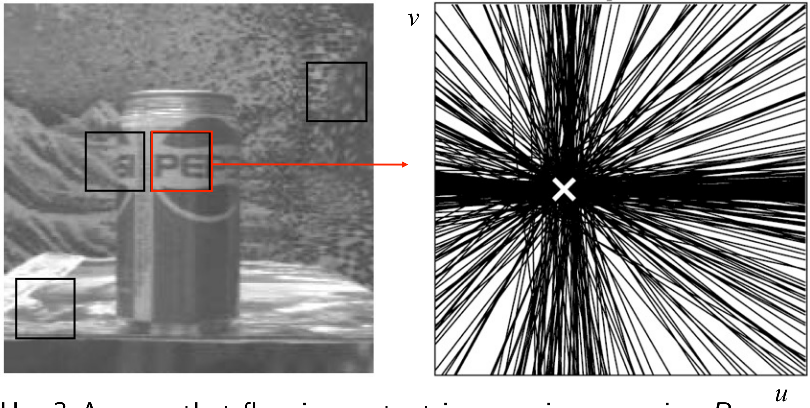
Two solutions are generally proposed to the aperture problem. The first involves finding 2D features, such as a corner of the square (c), whose motions are unambiguous. The second involves combining motion measurements made at different locations in space. As is evident in (d), when information from all four edges is available, the correct vertical motion can be extracted.

However, both of these strategies for solving the aperture problem run into further problems in real world scenes with multiple objects, as can be seen in the next demo.

- http://web.mit.edu/persci/demos/Motion&Form/mini.html

A simple Optical Flow Algorithm

- How can we use this in our optical flow algorithm?
- Combine multiple constraints!



- How? Assume that flow is constant in some image region R

$$E_{SSD}(u,v) \approx \sum_{(x,y)\in R} \left(u \cdot I_x(x,y,t) + v \cdot I_y(x,y,t) + I_t(x,y,t)\right)^2$$

A simple Optical Flow Algorithm

$$E_{SSD}(u,v) \approx \sum_{(x,y)\in R} \left(u \cdot I_x(x,y,t) + v \cdot I_y(x,y,t) + I_t(x,y,t)\right)^2$$

- How to minimize $E_{SSD}(u,v)$ wrt. flow (u,v)?
- Business as usual: Differentiate and set to zero:

$$\frac{\partial}{\partial u} E_{SSD}(u,v) \approx 2 \sum_{(x,y)\in R} \left(u \cdot I_x + v \cdot I_y + I_t\right) I_x = 0$$

$$\frac{\partial}{\partial v} E_{SSD}(u,v) \approx 2 \sum_{(x,y)\in R} \left(u \cdot I_x + v \cdot I_y + I_t\right) I_y = 0$$

- By rearranging the terms, we can write this as:

$$\underbrace{\begin{pmatrix} \sum_R I_x^2 & \sum_R I_x I_y \\ \sum_R I_x I_y & \sum_R I_y^2 \end{pmatrix}}_{\text{“structure tensor”}} \begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} -\sum_R I_t I_x \\ -\sum_R I_t I_y \end{pmatrix}$$

A simple Optical Flow Algorithm

$$\underbrace{\begin{pmatrix} \sum_R I_x^2 & \sum_R I_x I_y \\ \sum_R I_x I_y & \sum_R I_y^2 \end{pmatrix}}_{\text{“structure tensor”}} \begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} -\sum_R I_t I_x \\ -\sum_R I_t I_y \end{pmatrix}$$

- The structure tensor is a s.p.d. matrix (*i.e.*, we can invert it!)
- Rewriting this using the abbreviations from before:

$$\left(\sum_R \nabla I \nabla I^T\right) \mathbf{u} = -\sum_R I_t \nabla I$$

- We simply need to invert the structure tensor to obtain the flow:

$$\mathbf{u} = -\underbrace{\left(\sum_R \nabla I \nabla I^T\right)^{-1}}_{\in \mathbb{R}^{2\times 2}} \underbrace{\left(\sum_R I_t \nabla I\right)}_{\in \mathbb{R}^{2\times 1}}$$

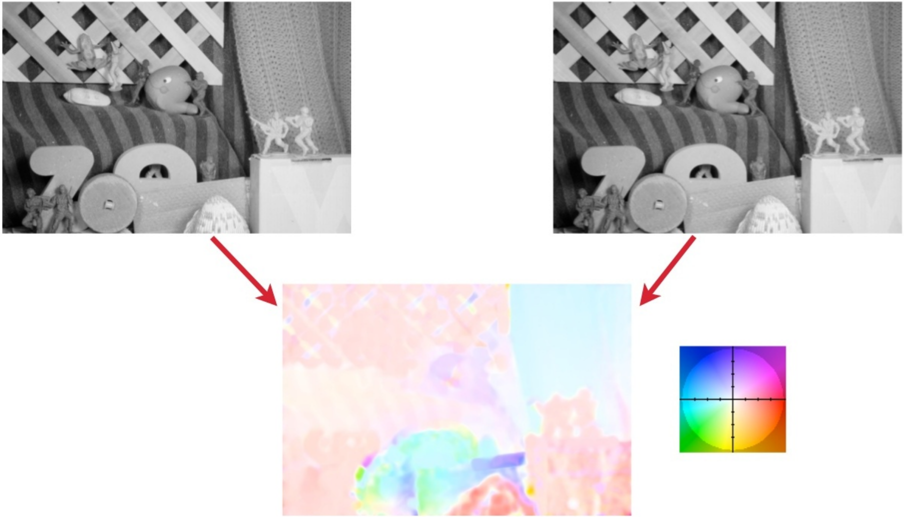
Lucas Kanade Optical Flow

Analytic solution to optical flow in image region R:

$$\mathbf{u} = -\left(\sum_R \nabla I \nabla I^T\right)^{-1} \left(\sum_R I_t \nabla I\right)$$

- This is a classic flow technique:
  B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. IJCAI, pp. 674-679, 1981.
- How to obtain dense motion fields?
- Compute a flow vector for each pixel using LK
- However, LK only works for small motions
  - Iterative estimation using warping
  - Coarse-to-fine estimation

Iterative Estimation



Calculate dense LK optical flow on input images

Iterative Estimation

Frame 1

Iterative Estimation

Frame 2

Iterative Estimation

Frame 2 (warped)

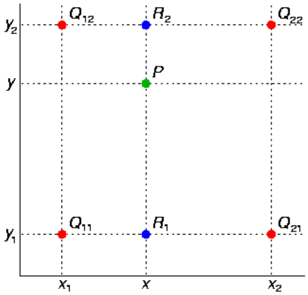
Iterative Estimation

Bilinear Interpolation for Image Warping:

$$f(R_1) \approx \frac{x_2 - x}{x_2 - x_1} f(Q_{11}) + \frac{x - x_1}{x_2 - x_1} f(Q_{21})$$

$$f(R_2) \approx \frac{x_2 - x}{x_2 - x_1} f(Q_{12}) + \frac{x - x_1}{x_2 - x_1} f(Q_{22})$$

$$f(P) \approx \frac{y_2 - y}{y_2 - y_1} f(R_1) + \frac{y - y_1}{y_2 - y_1} f(R_2)$$



Other interpolation strategies:

- Bicubic
- Lanczos

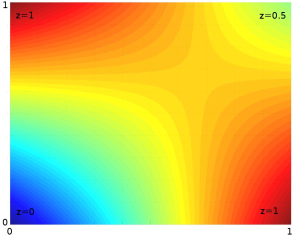
Iterative Estimation

Bilinear Interpolation for Image Warping:

$$f(R_1) \approx \frac{x_2 - x}{x_2 - x_1} f(Q_{11}) + \frac{x - x_1}{x_2 - x_1} f(Q_{21})$$

$$f(R_2) \approx \frac{x_2 - x}{x_2 - x_1} f(Q_{12}) + \frac{x - x_1}{x_2 - x_1} f(Q_{22})$$

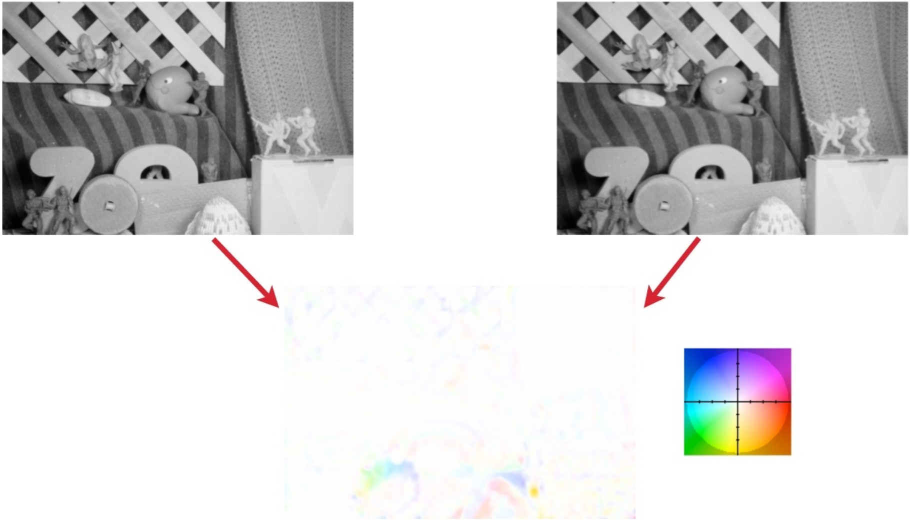
$$f(P) \approx \frac{y_2 - y}{y_2 - y_1} f(R_1) + \frac{y - y_1}{y_2 - y_1} f(R_2)$$



Other interpolation strategies:

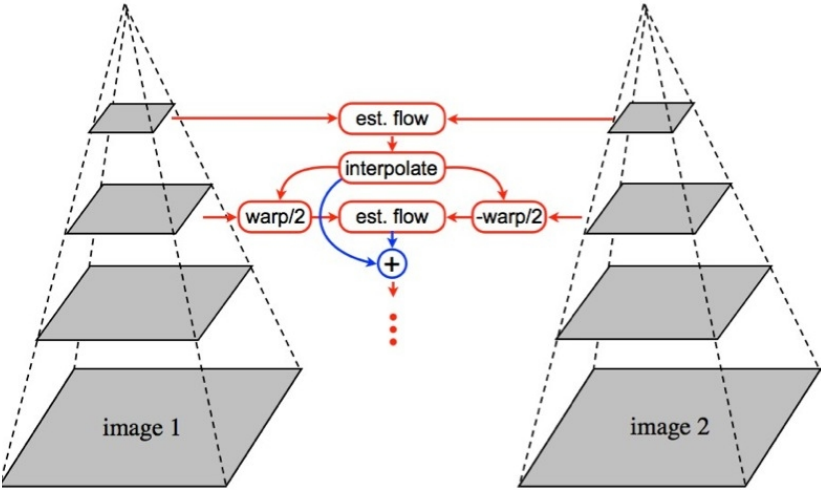
- Bicubic
- Lanczos

Iterative Estimation



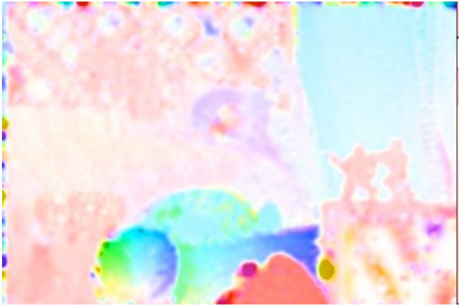
Estimate incremental flow on warped images, warp again, ...

Coarse-to-Fine Estimation

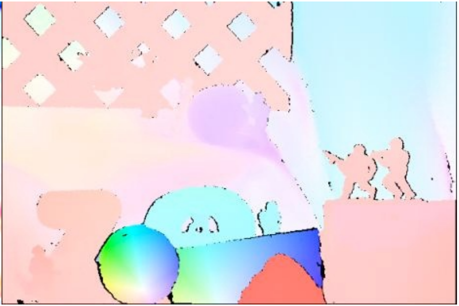


Build Gaussian pyramid, initialize with flow from coarser level

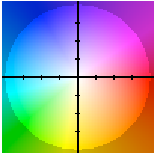
Results: Dense Iterative Lucas-Kanade Optical Flow (Coarse-to-Fine)



Lucas-Kanade



Ground Truth



Horn-Schunck Optical Flow

Problems of Lucas-Kanade:

- Similar to block matching (WTA) in stereo, LK is a local method which considers each pixel independently
- Spatial coherence is formulated in a simple way where the flow within an image window is assumed to be constant (or affine for affine LK)
- For small windows, ambiguities lead to wrong matches
- Large windows almost certainly violate the constant flow assumption
- As in stereo, there is no right choice for the window size!

What can we do?

- In stereo we introduced prior knowledge which supplied global regularity on the disparity map
- In optical flow we can do exactly the same!
- $\Rightarrow$ Horn-Schunck optical flow algorithm

Horn-Schunck Optical Flow

B. K. P. Horn and B. G. Schunck. Determining optical flow. Artificial Intelligence, 17(1-3):185-203, 1981.

- Consider the image I as a function of continuous variables x, y, t
- Consider $u(x, y)$ and $v(x, y)$ as continuous flow fields
- Goal: Minimizing the following energy functional

$$
\begin{aligned}
E(u,v) \;=\; & \iint \underbrace{\left(I(x+u(x,y), y+v(x,y), t+1) - I(x,y,t)\right)^2}_{\text{quadratic penalty for brightness change}} \\
& + \; \lambda \cdot \underbrace{\left(\|\nabla u(x,y)\|^2 + \|\nabla v(x,y)\|^2\right)}_{\text{quadratic penalty for flow change}} \, dx\, dy
\end{aligned}
$$

with regularization parameter λ and $\nabla = \left(\frac{\partial}{\partial x}, \frac{\partial}{\partial y}\right)$.

Horn-Schunck Optical Flow

$$
\begin{aligned}
E(u,v) &= \iint \left(I(x+u(x,y), y+v(x,y), t+1) - I(x,y,t)\right)^2 \\
&+ \lambda \cdot \left(\|\nabla u(x,y)\|^2 + \|\nabla v(x,y)\|^2\right) dx\,dy
\end{aligned}
$$

- Minimizing this directly is a hard problem because the energy is highly non-convex and has many local optima
- We've had this problem before and linearized the brightness constancy assumption
- We will do the same here:

$$
\begin{aligned}
E(u,v) &= \iint \left(I_x(x,y,t)u(x,y) + I_y(x,y,t)v(x,y) + I_t(x,y,t)\right)^2 \\
&+ \lambda \cdot \left(\|\nabla u(x,y)\|^2 + \|\nabla v(x,y)\|^2\right) dx\,dy
\end{aligned}
$$

Horn-Schunck Optical Flow

$$\begin{aligned} E(u,v) &= \iint \left(I_x(x,y,t)u(x,y) + I_y(x,y,t)v(x,y) + I_t(x,y,t)\right)^2 \\ &+ \lambda \cdot \left(\|\nabla u(x,y)\|^2 + \|\nabla v(x,y)\|^2 \right) dx\, dy \end{aligned}$$

- This energy imposes a quadratic penalty on the optical flow constraint and the gradient of the flow field
- It is convex and thus has a unique optimum
- The flow can be estimated by discretizing it spatially and performing gradient descent on the discretized objective:

$$\begin{aligned} E(\mathbf{U},\mathbf{V}) &= \sum_{x,y} \left(I_x(x,y)\, u_{x,y} + I_y(x,y)\, v_{x,y} + I_t(x,y)\right)^2 \\ &+ \lambda \cdot \left((u_{x,y} - u_{x+1,y})^2 + (u_{x,y} - u_{x,y+1})^2 + \right. \\ &\qquad \left. (v_{x,y} - v_{x+1,y})^2 + (v_{x,y} - v_{x,y+1})^2 \right) \end{aligned}$$

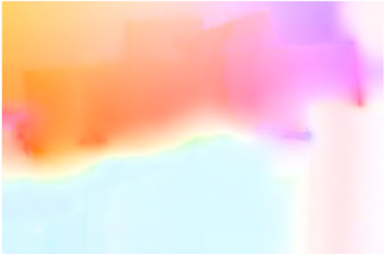
Horn-Schunck Optical Flow

Discretized Objective:

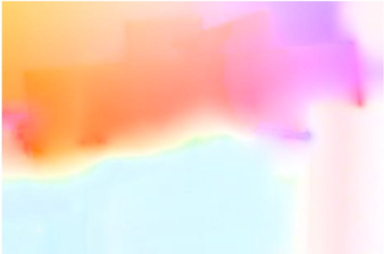
$$
\begin{aligned}
E(\mathbf{U},\mathbf{V}) \;\; &= \;\; \sum_{x,y} \left(I_x(x,y)\, u_{x,y} + I_y(x,y)\, v_{x,y} + I_t(x,y) \right)^2 \\
&+ \;\; \lambda \cdot \Big((u_{x,y} - u_{x+1,y})^2 + (u_{x,y} - u_{x,y+1})^2 + \\
&\qquad (v_{x,y} - v_{x+1,y})^2 + (v_{x,y} - v_{x,y+1})^2 \Big)
\end{aligned}
$$

- Differentiate wrt. $\mathbf{U}, \mathbf{V}$ and set the gradient to 0
- Results in a huge (but sparse) linear system
- Can be solved using standard techniques (*e.g.*, Gauss-Seidel, SOR)
- What would happen for $\lambda = 0$?
- However: Linearization works only for small motions!
  - Coarse-to-fine estimation
  - Warping

Result of [Horn-Schunck, 1981]



Result of [Horn-Schunck, 1981]



Horn-Schunck Optical Flow

- Our HS results are quite a bit better than the results using LK
- However, the flow is very smooth, *i.e.*, to overcome ambituities we need to set λ to a high value which oversmooths flow discontinuities
- Why?
- We use a quadratic penalty for penalizing changes in the flow
- This does not allow for discontinuities in the flow field
- It penalizes large changes too much!

Probabilistic Interpretation

Remember from introduction to graphical models:

$$p(x) = \frac{1}{Z} \exp\{-E(x)\}$$

Gibbs Energy:

$$\begin{aligned} E(\mathbf{U}, \mathbf{V}) &= \sum_{x,y} (I_x(x,y)\, u_{x,y} + I_y(x,y)\, v_{x,y} + I_t(x,y))^2 \\ &+ \lambda \cdot \big((u_{x,y} - u_{x+1,y})^2 + (u_{x,y} - u_{x,y+1})^2 + \\ &\qquad (v_{x,y} - v_{x+1,y})^2 + (v_{x,y} - v_{x,y+1})^2\big) \end{aligned}$$

Gibbs Distribution (Prior+Likelihood):

$$\begin{aligned} p(\mathbf{U}, \mathbf{V}) &\propto \prod_{x,y} \exp\left\{-(I_x(x,y)\, u_{x,y} + I_y(x,y)\, v_{x,y} + I_t(x,y))^2\right\} \\ &\times \exp\left\{-\lambda(u_{x,y} - u_{x+1,y})^2\right\} \times \exp\left\{-\lambda(u_{x,y} - u_{x,y+1})^2\right\} \\ &\times \exp\left\{-\lambda(v_{x,y} - v_{x+1,y})^2\right\} \times \exp\left\{-\lambda(v_{x,y} - v_{x,y+1})^2\right\} \end{aligned}$$

Robust Regularization

- The classic Horn-Schunck formulation uses quadratic penalties ...
  - ... for enforcing the brightness constancy assumption
  - ... for enforcing smooth flow fields

$$\begin{aligned} p(\mathbf{U},\mathbf{V}) \quad &\propto \quad \prod_{x,y} \exp\left\{-(I_x(x,y)\,u_{x,y} + I_y(x,y)\,v_{x,y} + I_t(x,y))^2\right\} \\ &\times \quad \exp\left\{-\lambda(u_{x,y} - u_{x+1,y})^2\right\} \times \exp\left\{-\lambda(u_{x,y} - u_{x,y+1})^2\right\} \\ &\times \quad \exp\left\{-\lambda(v_{x,y} - v_{x+1,y})^2\right\} \times \exp\left\{-\lambda(v_{x,y} - v_{x,y+1})^2\right\} \end{aligned}$$

- Both assumptions are invalid in practice (occlusions/discontinuities)!
- Formulation with robust data term and smoothness penalties:

$$\begin{aligned} p(\mathbf{U},\mathbf{V}) \quad &\propto \quad \prod_{x,y} \exp\left\{-\rho_D\left(I_x(x,y)\,u_{x,y} + I_y(x,y)\,v_{x,y} + I_t(x,y)\right)\right\} \\ &\times \quad \exp\left\{-\lambda\,\rho_S(u_{x,y} - u_{x+1,y})\right\} \times \exp\left\{-\lambda\,\rho_S(u_{x,y} - u_{x,y+1})\right\} \\ &\times \quad \exp\left\{-\lambda\,\rho_S(v_{x,y} - v_{x+1,y})\right\} \times \exp\left\{-\lambda\,\rho_S(v_{x,y} - v_{x,y+1})\right\} \end{aligned}$$

Robust Regularization

$$\begin{aligned} p(\mathbf{U},\mathbf{V}) \quad &\propto \quad \prod_{x,y} \exp\left\{-\rho_D\left(I_x(x,y)\,u_{x,y} + I_y(x,y)\,v_{x,y} + I_t(x,y)\right)\right\} \\ &\times \quad \exp\left\{-\lambda\,\rho_S(u_{x,y} - u_{x+1,y})\right\} \times \exp\left\{-\lambda\,\rho_S(u_{x,y} - u_{x,y+1})\right\} \\ &\times \quad \exp\left\{-\lambda\,\rho_S(v_{x,y} - v_{x+1,y})\right\} \times \exp\left\{-\lambda\,\rho_S(v_{x,y} - v_{x,y+1})\right\} \end{aligned}$$

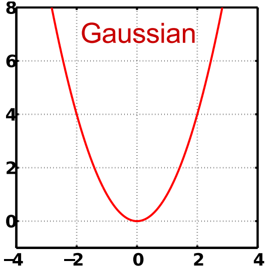
- But how to choose $\rho_D(\cdot)$ and $\rho_S(\cdot)$?
- We want a prior that allows for discontinuities in the optical flow field (and occlusions in the data term)
- Thus we need something more heavy-tailed than a Gaussian distribution, such as a Student-t distribution (Lorentzian penalty):

$$p(x) \propto \left(1 + \frac{x^2}{2\sigma^2}\right)^{-\alpha} \quad \Rightarrow \quad \rho(x) = ?-\log\left(p(x)\right) = \alpha \log\left(1 + \frac{x^2}{2\sigma^2}\right)$$

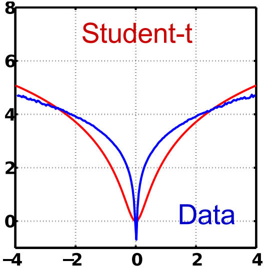
Robust Regularization

- Student-t distribution / Lorentzian penalty:

$$p(x) \propto \left(1 + \frac{x^2}{2\sigma^2}\right)^{-\alpha} \quad \Rightarrow \quad \rho(x) = -\log\left(p(x)\right) = \alpha \log\left(1 + \frac{x^2}{2\sigma^2}\right)$$



negative
log-density
(i.e. energy)



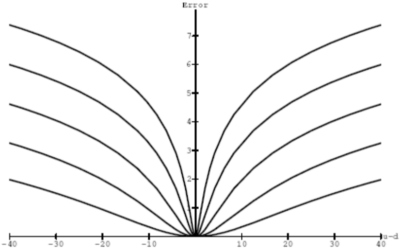
- Has been proposed in [Black-Anandan, 1996]
- How to estimate the parameters? Learn from data!

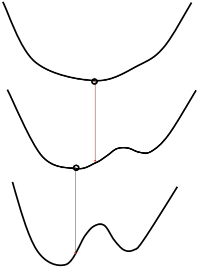
Robust Regularization

- How to obtain optical flow ground truth data?
- There is no sensor which can measure optical flow directly
- Synthesize optical flow fields using
  - A set of natural geometries (*e.g.*, Brown range database)
  - A set of natural camera motions

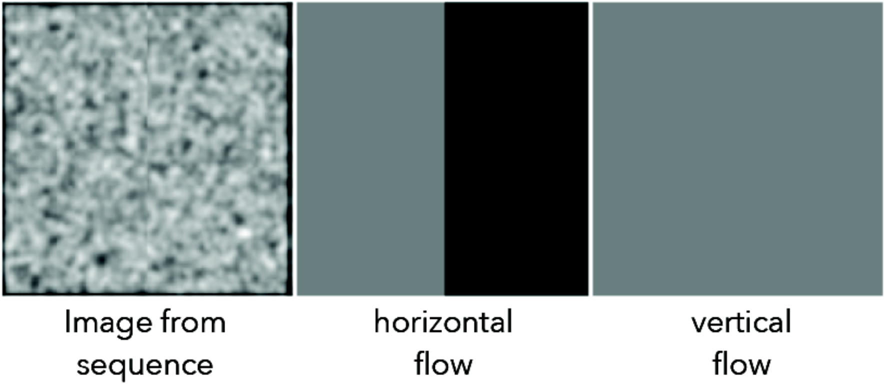
Robust Regularization

- How to optimize the energy function with robust penalties?
- Non-convex data and smoothness terms due to $\rho(\cdot)$!
- This is hard. Several tricks applied in practice:
  - Linearization, warping, coarse-to-fine estimation
  - Graduated non-convexity: Start with a quadratic optimization problem, then gradually increase the difficulty / non-convexity of the problem:





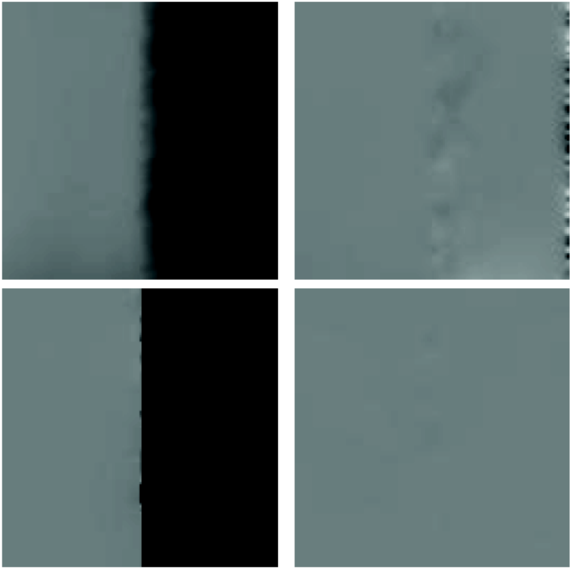
Robust Regularization



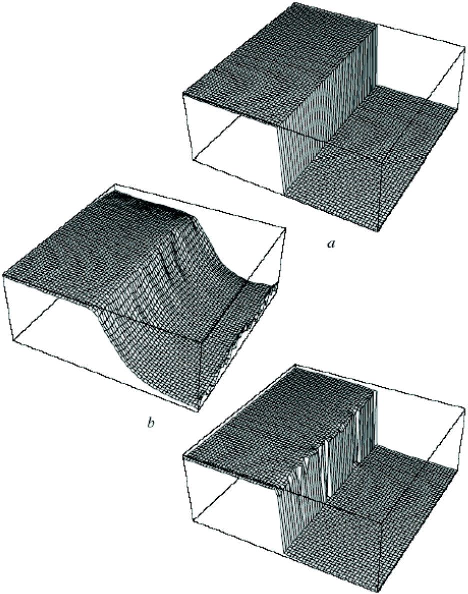
Robust Regularization

Estimated flow with Gaussian spatial term:

Estimated flow with robust (Student-t) spatial term:



Robust Regularization

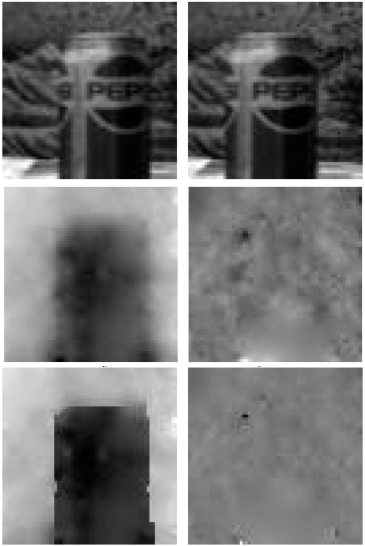


True horizontal flow

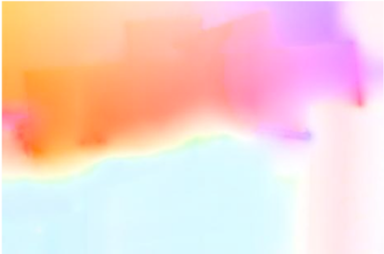
Estimated with Gaussian spatial term

Estimated flow with robust (Student-t) spatial term

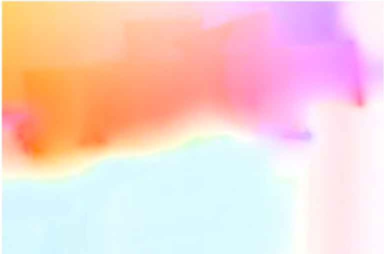
Robust Regularization



Result of [Horn-Schunck, 1981]



Result of [Horn-Schunck, 1981]

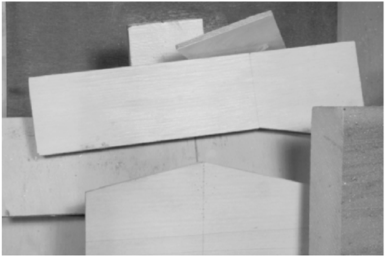


Results of [Black-Anandan, 1996]

Results of [Black-Anandan, 1996]

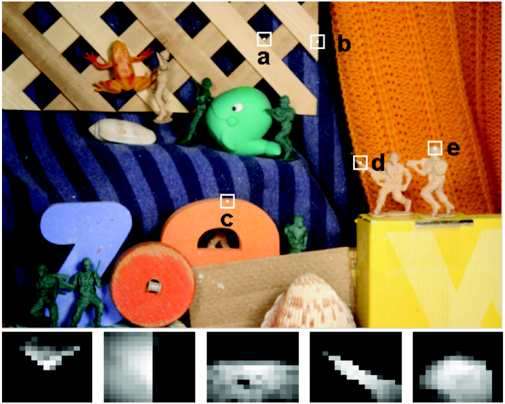
Results of [Sun et al., 2010]

Results of [Sun et al., 2010]



Non-local Regularization

- Pairwise regularization is often too limited
- Idea: use larger neighborhoods for regularization
- Problem: Not all pixels in the neighborhood are equally meaningful
  $\Rightarrow$ introduce adaptive weights based on image color or intensity:



Exercise for this Week

You will use your digital camera and the robust optical flow algorithm discussed in this lecture to create your own bullet time sequence!