Chapter 3
Advanced, non-linear and non-convex problems

Variational Methods for Computer Vision
WS 16/17

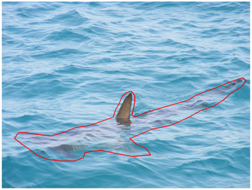
Michael Moeller
Visual Scene Analysis
Department of Computer Science
University of Siegen

Image Segmentation

Motivation

Remember: we have nice tools for image manipulation and editing now.

But we need a segmentation!



How do we segment images automatically?

Motivation

Numerous further applications, e.g.

- Image editing and composition
- Medical imaging
  - Locate and measure tumors
  - Measure tissue volumes
  - Diagnosis of anatomical structure
  - Automatic data analysis
- Object detection and recognition
  - Security and surveillance - e.g. faces, people, fingerprints
  - Self driving cars - traffic signs, brake lights, pedestrians
- Hyperspectral Imaging
  - Mineralogy
  - Agricultural applications
  - Chemical imaging

See Wikipedia `https://en.wikipedia.org/wiki/Image_segmentation` or search for *image segmentation* online to see more examples.

Example: Two region segmentation

Lets start with a simple example:

Find all coins in the image!

Thresholding

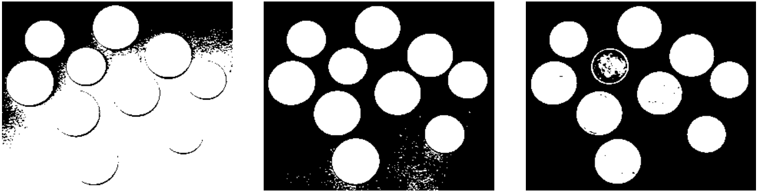
First idea for representing a segmentation: characteristic function

$$u(x) = \begin{cases} 1 & \text{if } x \text{ is foreground,} \\ 0 & \text{if } x \text{ is background.} \end{cases}$$

First idea for segmentation: Thresholding!

$$u(x) = \begin{cases} 1 & \text{if } f(x) > t, \\ 0 & \text{if } f(x) \leq t. \end{cases}$$

Coin results for different thresholds



$t = 0.25$ $t = 0.3$ $t = 0.5$

Thresholding

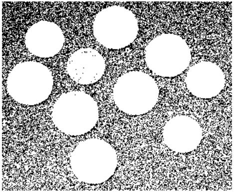
First idea for representing a segmentation: characteristic function

$$u(x) = \begin{cases} 1 & \text{if } x \text{ is foreground,} \\ 0 & \text{if } x \text{ is background.} \end{cases}$$

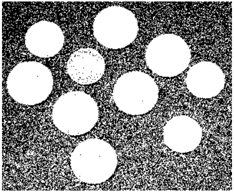
First idea for segmentation: Thresholding!

$$u(x) = \begin{cases} 1 & \text{if } f(x) > t, \\ 0 & \text{if } f(x) \leq t. \end{cases}$$

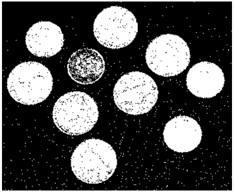
The results get far worse if the input image is noisy



$t = 0.25$



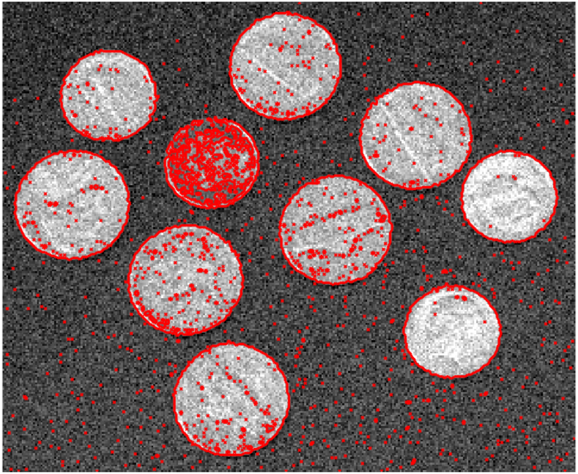
$t = 0.3$



$t = 0.5$

Thresholding

The resulting segmentation is highly scattered and irregular!



We need to introduce regularity of the segmentation curve!

Variational two region segmentation

Can we formulate the segmentation problem as a variational method?

We are still looking for a characteristic function:

$$u(x) = \begin{cases} 1 & \text{if } x \text{ is foreground,} \\ 0 & \text{if } x \text{ is background,} \end{cases}$$

i.e. $u : \Omega \to \{0, 1\}$ is our unknown.

Let us assume that we expect the foreground to have an expected intensity of $c_1 \in [0, 1]$ and the background to have an expected intensity of $c_2 \in [0, 1]$. A reasonable model is

$$H_f(u) = \int_\Omega (f(x) - c_1)^2 u(x)\, dx + \int_\Omega (f(x) - c_2)^2 (1 - u(x))\, dx$$

Board: Minimizing the above is the same as minmizing

$$H_f(u) = \int_\Omega \left(\frac{c_1 + c_2}{2} - f(x) \right) u(x)\, dx$$

Relation to thresholding

If we merely minimize

$$H_f(u) = \int_\Omega \left(\frac{c_1 + c_2}{2} - f(x) \right) u(x)\, dx,$$

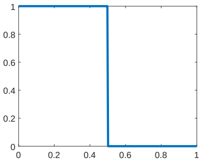
then we threshold f by $t := \frac{c_1+c_2}{2}$, i.e.

- If $t - f(x) < 0$ then $u(x)$ tries to be as large as possible, i.e. $u(x) = 1$.
- If $t - f(x) > 0$ then $u(x)$ tries to be as small as possible, i.e. $u(x) = 0$.

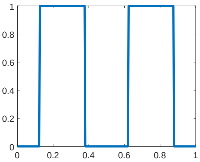
Next question what is a desirable regularity and how do we enforce it?

Regularization for image segmentation

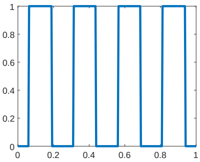
Let us consider the simple case of 1D segmentation. The following plots show different characteristic functions $u : [0,1] \to \{0,1\}$. What value would you assign to a *regularization function* for the different u?



Regularity 1



Regularity 4



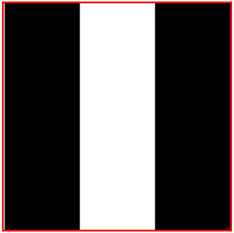
Regularity 8

Count the number of jumps!

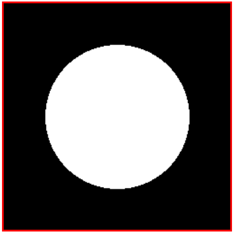
Regularization for image segmentation

In 2D there will be infinitely many jumps (for all non-trivial segmentations). However, the analogy of the 1D costs would be to penalize the length of the boundary of the segmentation!

Regularity l



Regularity $2l$



Regularity $2l$

How can we formulate the above regularizations in a variational setting?

Let us again start with the 1D case: If $u : [0,1] \to \{0,1\}$ has finitely many jumps as positions $\{x_1, \cdots, x_n\}$, then

$$|\nabla u| = \sum_{i=1}^{n} \delta_{x_i}, \quad \text{with delta distributions } \delta_{x_i}.$$

Counting the number of jumps

The so-called *delta distributions* are sometimes **heuristically** described as

$$\delta_{x_i} = \begin{cases} \infty & \text{if } x = x_i, \\ 0 & \text{otherwise,} \end{cases}$$

but their defining property is

$$\int_0^1 g(x)\, \delta_{x_i}(x)\, dx = g(x_i) \qquad \forall g \in C^\infty([0,1]).$$

Thus, for $u : [0,1] \to \{0,1\}$ with finitely many jumps as positions $\{x_1, \cdots, x_n\}$, we find

$$\int_0^1 |\nabla u(x)|\, dx = \sum_{i=1}^n \left(\int_0^1 \delta_{x_i}(x)\, dx \right) = \sum_{i=1}^n 1 = n,$$

as desired!

Formal framework for TV regularization

Interestingly, in the 1d case the total variation penalty

$$TV(u) = \int_0^1 |\nabla u(x)| \, dx$$

seems to be the right choice for counting the number of jumps!

Care has to be taken in our understanding of ∇u which we generalized from our usually understanding of differentiability to a differentiation of jump functions. Mathematically correct, we should consider

$$TV(u) = \sup_{p \in C_0^\infty(\Omega), ||p||_\infty \leq 1} \int_\Omega u(x) \, \text{div}(p)(x) \, dx,$$

but we will not detail these aspects here.

Natural question: Is the penalty $TV(u)$ the right choice even for $u : \Omega \subset \mathbb{R}^2 \to \{0, 1\}$?

Penalizing the perimeter

Co-area formula

For any function $u \in L^1(\Omega)$ with $TV(u) < \infty$ it holds that

$$TV(u) = \int_{-\infty}^{\infty} \text{Perimeter}\{x \mid u(x) > \theta\} \, d\theta$$

This is sufficient to see that $TV(u)$ is the right choice in 2d (or any other dimension), too! More of this on the exercise sheet!

Since for a function $u : \Omega \to \{0, 1\}$ it holds that $TV(u)$ is the perimeter of the set $\{x \mid u(x) = 1\}$ i.e. the perimeter of the object to be segmented we can now phrase our variational method as

$$\hat{u} = \arg \min_{u:\Omega \to \{0,1\}} \int_{\Omega} (t - f(x)) \, u(x) \, dx + \alpha TV(u).$$

While the cost function itself is convex, the constraint to functions $u : \Omega \to \{0, 1\}$ is highly nonconvex! How can we determine the minimizer?

Convex relaxation

Use convex relaxation and minimize over all $u : \Omega \to [0,1]$ instead!

(Sanity check: Why does it make sense that using $u : \Omega \to [0,1]$ is a convex relaxation of optimizing over all functions $u : \Omega \to \{0,1\}$?)

We use

$$\hat{u} = \arg \min_{u:\Omega \to [0,1]} \int_\Omega (t - f(x))\, u(x)\, dx + \alpha TV(u),$$

and discretize to arrive at

$$\hat{u} = \arg \min_{u, u_{i,j} \in [0,1]} \sum_{i,j} (t - f_{i,j})\, u_{i,j} + \alpha \sum_{i,j} \sqrt{(D_x u)_{i,j}^2 + (D_y u)_{i,j}^2}.$$

First question: How do we minimize the above functional with constraints?

Simplest answer: Smooth the non-differentiable part, use gradient descent and project the result of each iteration back to the constraint set!

Projected gradient descent

Task: Solve $\min_{u \in M} E(u)$ for a differentiable E, and a convex constraint set M.

Gradient projection with backtracking line search

Pick $\beta \in]0, 1[$. Iterate:

- Given an estimate u^k, compute $E(u^k)$ and $\nabla E(u^k)$.
- Initialize $\tau_k = \tau^0$.
- Find a good τ_k by:

$$u^{test} = \text{proj}_M(u^k - \tau_k \nabla E(u^k))$$

while $E\left(u^{test}\right) > E(u^k) - \langle \nabla E(u^k), u^k - u^{test} \rangle + \frac{1}{2\tau^k} \left\| u^k - u^{test} \right\|^2$

$$\tau_k \leftarrow \beta \tau_k$$

end

- Once τ^k meets the criterion in the while-loop, update

$$u^{k+1} = u^{test}.$$

Projections

Let $M = \{u \mid u_{i,j} \in [0,1]\}$. Then

$$\hat{u} := \text{proj}_M(v) := \underset{u \in M}{\text{argmin}} \|u - v\|_2^2$$

is given by

$$\hat{u}_{i,j} = \begin{cases} 1 & \text{if } v_{i,j} > 1 \\ v_{i,j} & \text{if } v_{i,j} \in [0,1] \\ 0 & \text{if } v_{i,j} < 0 \end{cases} = \min(\max(v_{i,j}, 0), 1)$$

The above projection is easy and the convergence criteria for the usual gradient descent and the projected gradient descent algorithm are identical. We know how to solve

$$\hat{u} = \arg \min_{u:\Omega \to [0,1]} \int_\Omega (t - f(x))\, u(x)\, dx + \alpha TV(u),$$

with the respective discretization

$$\hat{u} = \arg \min_{u, u_{i,j} \in [0,1]} \sum_{i,j} (t - f_{i,j})\, u_{i,j} + \alpha \sum_{i,j} \sqrt{(D_x u)_{i,j}^2 + (D_y u)_{i,j}^2}.$$

Relation: convex and nonconvex energy

The segmentation problem appears to be solved, BUT we relaxed $u : \Omega \to \{0, 1\}$ to $u : \Omega \to [0, 1]$ and are not guaranteed to get binary solutions as minimizers $\hat{u}$ anymore.

Remark: If $\hat{u}$ is binary then $\hat{u}$ is the global minimizer of the nonconvex energy! Why?

If $\hat{u}$ is not binary, we could pick $\mathbf{1}_{u>0.5}(x)$, where $\mathbf{1}_{u>\theta}$ denotes the threshold operator

$$\mathbf{1}_{u>\theta}(x) = \begin{cases} 1 & \text{if } u(x) > \theta, \\ 0 & \text{otherwise.} \end{cases}$$

The function $\mathbf{1}_{u>\theta}$ is binary and hence feasible, but will it be anywhere close to being a minimizer of the original nonconvex cost function?

Global optimality for 2-region segmentation

Remarkable result established by Chan, Esedoglu and Nicolova in *Algorithms for Finding Global Minimizers of Image Segmentation and Denoising Models*, 2006:

Exact convex relaxation for two region segmentation

Consider the minimization problems

$$\min_{u:\Omega\to\{0,1\}} \int_\Omega (t - f(x))\, u(x)\, dx + \alpha TV(u), \qquad \text{(NC)}$$

$$\min_{u:\Omega\to[0,1]} \int_\Omega (t - f(x))\, u(x)\, dx + \alpha TV(u). \qquad \text{(C)}$$

Let $\tilde{u}$ be a minimizer of (C). Then for any $\theta \in]0,1[$ the function $\mathbf{1}_{u>\theta}$ is a global minimizer of (NC)!

Some remarks

- Amazing result: **We can solve a highly nonconvex problem to global optimality** using convex relaxation!
- The result is specific to the energy and does not hold for all binary labeling problems.
- Similar problems have been studied in the discrete setting with the anisotropic total variation as a regularization.
- Global minimizers can be computed in the anisotropic spatially discrete setting using the min-cut/max-flow duality, see Ford and Fulkerson 1962, and Greig, Porteous, Seheult 1989.
- We will prove the theorem in the exercises. The two main ingredients are the co-area formula we already learned about, and the following formula.

Layer cake formula

Layer cake formula

Any measurable function $u : \mathbb{R}^n \to \mathbb{R}^+$ can be expressed via

$$u(x) = \int_0^\infty \mathbf{1}_{u>\theta}(x) \; d\theta.$$

Summary

The variational approach

$$\min_{u:\Omega\to[0,1]} \underbrace{\int_\Omega (t - f(x))\, u(x)\, dx + \alpha TV(u)}_{=:E(u)} \qquad \text{(C)}$$

performs a segmentation of the image f into two regions.

The alone data term acts as a thresholding. $t - f(x)$ can be seen as a measure of likelihood of x belonging to the background.

For functions $u : \Omega \to \{0, 1\}$ the regularization $TV(u)$ penalizes the boundary length of the segmentation.

The combination of the layer-cake formula and the co-area formula allows us to show that any thresholding $\mathbf{1}_{\tilde{u}>\theta}$ of a global minimizer $\tilde{u}$ of (C) is a global minimizer of the nonconvex problem $\min_{u:\Omega\to\{0,1\}} E(u)$.

ALERT: WRONG FORMULA IN PREVIOUS LECTURE!

Gradient projection with backtracking line search

Pick $\beta \in]0, 1[$. Iterate:

- Given an estimate u^k, compute $E(u^k)$ and $\nabla E(u^k)$.
- Initialize $\tau_k = \tau^0$.
- Find a good τ_k by:

$$u^{test} = \text{proj}_M(u^k - \tau_k \nabla E(u^k))$$

while $E\left(u^{test}\right) > E(u^k) - \langle \nabla E(u^k), u^k - u^{test} \rangle + \frac{1}{2\tau^k} \left\| u^k - u^{test} \right\|^2$

$\tau_k \leftarrow \beta \tau_k$

end

- Once τ^k meets the criterion in the while-loop, update

$$u^{k+1} = u^{test}.$$

Beyond intensity based segmentation

Particularly interesting for more practically relevant applications: $p(x) = t - f(x)$ can be seen as a measure of likelihood of x belonging to the background.

The data term

$$H_p(u) := \int_\Omega p(x)\, u(x)\, dx$$

leads to a desire of $u(x)$ to be 0 for large values of $p(x)$ and to be 1 for negative values of $p(x)$ with large magnitude. Points x at which $p(x)$ has a small magnitude will be dominated by the regularization.

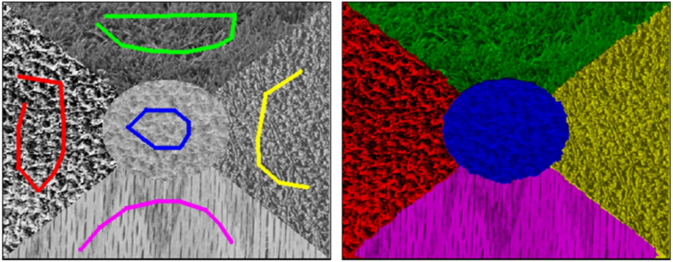
We do not have to choose $p(x) = t - f(x)$ for a threshold t, but may use any oracle p telling us the likelihood of a point belonging to the fore- or background.

Interesting choices for p

- Constant intensity (as seen before), e.g. Chan and Vese, *Active Contours Without Edges*, 2001.
- Histograms of parts to be segmented, e.g. Boykov and Jolly, *Interactive Graph Cuts*, 2001.
- Mixture of Gaussians of parts to be segmented, e.g. Rother, Kolmogorov, and Blake. *Grab-cut: interactive foreground segmentation using iterated graph cuts*, 2004.
- Spatially varying probabilities, e.g. Nieuwenhuis, and Cremers, *Spatially Varying Color Distributions for Interactive Multi-Label Segmentation*, 2012.
- Likelihoods based on geodesic distances, e.g. Protiere, Sapiro, *Interactive Image Segmentation via Adaptive Weighted Distances*, 2007.

Interesting choices for p

Of course the probability does not have to depend on color, it could also be a texture descriptor:



From: Protiere, Sapiro, *Interactive Image Segmentation via Adaptive Weighted Distances*, 2007.

Very powerful trend in computer vision: Stop designing features and image descriptors 'by hand'! Use a database of Millions of images and let a computer figure out the best descriptors itself $\rightarrow$ **Deep learning!**

Example: Zheng et al., *Conditional Random Fields as Recurrent Neural Networks*, 2015.

Example: Histogram based

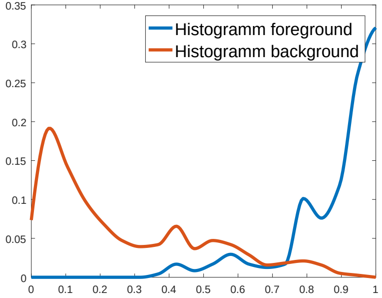
Common approach: Penalize

$$u(x)p_{fg}(x) + (1 - u(x))p_{bg}(x) = u(x)(p_{fg}(x) - p_{bg}(x)) + p_{bg}(x)$$

at each point x.

As an example, consider the histogram model. To determine the probabilities p_{fg} and p_{bg} one first needs histograms F and B of the expected fore- and background, e.g. by user scribbles. Then one determines

$$p_{fg} = -\log(p(f(x)|F), \qquad p_{bg} = -\log(p(f(x)|B).$$



Extension to multiple regions

So far we only considered dividing the image into foreground and background

What about multi-region segmentation?

$$\min_{\cup_i M_i = \Omega} \sum_{i=1}^{N} \left(\underbrace{\int_{M_i} p_i(x)\, dx}_{\text{costs to include point } x \text{ in region } M_i} + \alpha \underbrace{|\partial M_i|}_{\text{length of the boundary of } M_i} \right),$$

Natural extension of our previous variational method: Look for characteristic functions $u_i : \Omega \to \{0, 1\}$ of the different regions M_i!

Natural data term

$$\sum_i \int_\Omega u_i(x) p_i(x)\, dx = \sum_i \int_{M_i} p_i(x)\, dx,$$

where $p_i(x)$ is small in comparison to all other $p_j(x)$ if the object i is likely to be found at position x.

Multi region segmentation

Sanity check: What is

$$\min_{\substack{u_i:\Omega\rightarrow[0,1],\\ \sum_i u_i(x)=1\ \forall x}} \sum_i \int_\Omega u_i(x)p_i(x)\ dx?$$

How do we choose a regularization for the perimeters $\sum_i |\partial M_i|$ in terms of u_i?

We already know that if

$$u_i(x) = \begin{cases} 1 & \text{if } x \in M_i, \\ 0 & \text{if } x \notin M_i, \end{cases}$$

then it holds that

$$TV(u_i) = \text{Perimeter}(M_i).$$

It makes sense to consider $\sum_i TV(u_i)$ as a regularization:

$$\min_{\substack{u_i:\Omega\rightarrow\{0,1\},\\ \sum_i u_i(x)=1\ \forall x}} \sum_i \int_\Omega u_i(x)p_i(x)\ dx + \alpha TV(u_i).$$

Multi region segmentation

Natural idea proposed in Lellmann et al., *Convex Multi-class Image Labeling by Simplex-Constrained Total Variation*, 2008: Use the convex relaxation of replacing $u_i : \Omega \to \{0, 1\}$ by $u_i : \Omega \to [0, 1]$:

$$\min_{\substack{u_i:\Omega\to[0,1],\\ \sum_i u_i(x)=1 \ \forall x}} \sum_{i=1}^{N} \int_\Omega u_i(x)p_i(x)\ dx + \alpha TV(u_i),$$

A smoothed version of this problem is something we can solve with our projected as soon as we know how to project onto the set

$$S := \{v \in \mathbb{R}^N \mid v_i \geq 0,\ \sum_{i=1}^{N} v_i = 1\}$$

The latter is possible in $\mathcal{O}(N \log(N))$ complexity (or even better in $\mathcal{O}(N)$ expected complexity), but will not be detailed here. See Duchi et al., *Efficient Projections onto the ℓ^1-Ball for Learning in High Dimensions*, 2008.

More on convex optimization (without smoothing the energy) next semester!

Multi region segmentation

For the segmentation approach

$$\min_{\substack{u_i:\Omega\to[0,1],\\ \sum_i u_i(x)=1\ \forall x}} \sum_{i=1}^{N} \int_\Omega u_i(x)p_i(x)\ dx + \alpha TV(u_i),$$

the thresholding theorem no longer holds.

Interesting alternative: Consider the functional $r : \mathbb{R}^{N\times 2} \to \mathbb{R}$

$$r(d) = \begin{cases} \|\nu\|_2 & \text{if } \exists i \neq j,\ \nu \in \mathbb{R}^2 \text{ such that } d = (e_i - e_j)\nu^T, \\ \infty & \text{otherwise.} \end{cases}$$

Then

$$\int_\Omega r^{**}(\nabla u(x))\ dx$$

is an alternative convex regularization that also penalizes the length of the boundary of the segmentation if all u_i are binary.

Remark: Even with this regularization the thresholding theorem does not hold...

Multi region segmentation

... but it has been proven by Chambolle, Cremers, and Pock, *A Convex Approach to Minimal Partitions*, 2012, that

$$\min_{\substack{u_i:\Omega\to[0,1],\\ \sum_i u_i(x)=1\ \forall x}} \sum_{i=1}^{N} \int_\Omega u_i(x)p_i(x)\ dx + \alpha \int_\Omega r^{**}(\nabla u(x))\ dx$$

is a larger convex underapproximation to the nonconvex functional

$$\min_{\cup_i M_i=\Omega} \sum_{i=1}^{N} \int_{M_i} p_i(x)\ dx + \alpha|\partial M_i|,$$

than

$$\min_{\substack{u_i:\Omega\to[0,1],\\ \sum_i u_i(x)=1\ \forall x}} \sum_{i=1}^{N} \int_\Omega u_i(x)p_i(x)\ dx + \alpha TV(u_i).$$

Punchline: Since computing the convex hull of the entire functional is not feasible, there are several different ways to find convex underapproximations, which can differ in the tightness of their approximation!

Summary: Multi region segmentation

The segmentation of the image domain Ω in N regions M_i can be phrased as solving

$$\min_{\cup_i M_i = \Omega} \sum_{i=1}^{N} \int_{M_i} p_i(x)\, dx + \alpha |\partial M_i|. \qquad \text{(Multi-NC)}$$

A possible convex approximation is

$$\min_{\substack{u_i:\Omega\to[0,1],\\ \sum_i u_i(x)=1\ \forall x}} \sum_{i=1}^{N} \int_{\Omega} u_i(x) p_i(x)\, dx + \alpha TV(u_i), \qquad \text{(Multi-C)}$$

where one finally determines $M_i = \{x \mid u_i(x) \geq u_j(x)\ \forall j\}$.

Solving a discrete version of (Multi-NC) with anisotropic TV as a regularization is NP-hard. Thus, we cannot expect to find a feasible convex formulation for which a threshold theorem holds.

Tighter convex approximations than (Multi-C) can be derived by different convexification strategies.

Stereo Matching

How can we sense depth?

Left eye / camera

How can we sense depth?

Right eye / camera

Stereo vision

Given two images of the same scene recorded with two cameras, can we compute the distance of the objects to the camera?

Simplifying assumptions: Cameras are level and image sensors are flat on the same plane.

Board: Draw example for a pinhole camera model and compute the relation of object displacement, depth and camera parameters.

We assume that a preprocessing step called *image rectification* makes the assumptions “Cameras are level and image sensors are flat on the same plane” valid. As seen on the board, knowing the object displacement, also called *disparity*, is sufficient to determine the depth. We will know focus on how to determine a disparity image from two given color images!

Data fidelity

Based on our previous considerations, we need to find a displacement $v : \Omega \to \mathbb{R}$ in such a way that the point/object at position (x, y) in image 1 is the same as the point/object at position $(x + v(x, y), y)$.

We need a measure for how likely the position (x, y) in image 1 and $(x + v(x, y), y)$ in image 2 are the same point.

Simplest criterion: **Photoconsistency**. The same point has the same color in both images. This gives rise to

$$H(v) = \int_\Omega |f_1(x, y) - f_2(x + v(x, y), y)|^2 \, dx \, dy$$

as a cost function.

Data fidelity

Many variants of stereo matching cost functions exist. Since the pointwise comparison

$$H(v) = \int_\Omega |f_1(x,y) - f_2(x + v(x,y), y)|^2 \, dx \, dy$$

is not robust to noise, people considered patch-wise comparisons

$$H(v) = \int_\Omega \int_{\mathcal{N}(x,y)} |f_1(s,t) - f_2(s + v(x,y), t)|^2 \, ds \, dt \, dx \, dy$$

or gradient-based variants to be more robust to illumination changes.

For more sophisticated data terms, see for instance the classical work Zabih, Woodfill, *Nonparametric Local Transforms for Computing Visual Correspondence*, '94, or Yoon, So, *Adaptive Support-Weight Approach for Correspondence Search*, '06.

Data fidelity

For now, let us just stick to

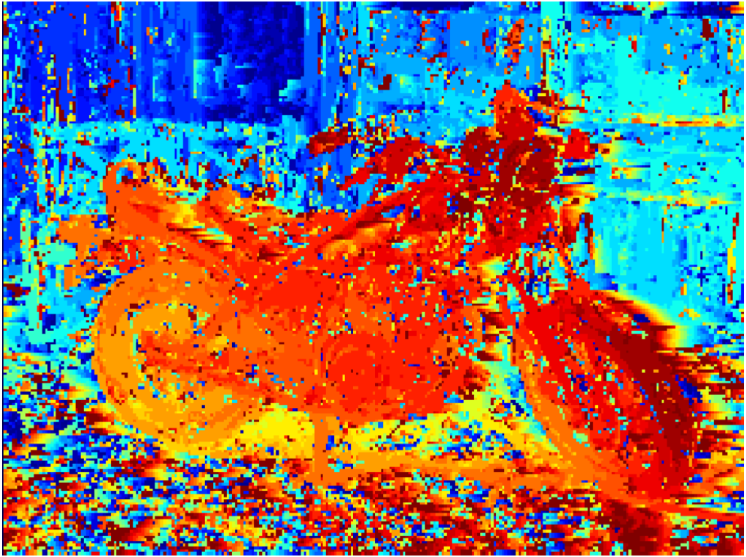
$$H(v) = \int_\Omega |f_1(x,y) - f_2(x + v(x,y), y)|^2 \, dx \, dy.$$

Some observations:

- The data term is highly nonconvex.
- The data term is pointwise, i.e. in the discrete setting we can solve
$$\min_{v \in \mathbb{R}} |f_1(x,y) - f_2(x+v,y)|^2$$
at each pixel (x,y) separately.
- In the discrete setting, we either need a model how to evaluate $f_2(x+v,y)$ for non-integers $x+v$ or we restrict the disparities v to be integers.
- For integers v, one can do a pixel-wise brute force search, to minimize the data term globally.

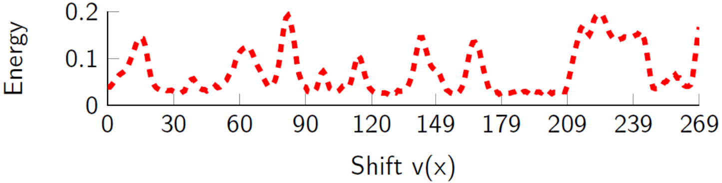
Pointwise minimization of photoconsistency

Pointwise minimization of photoconsistency



Pointwise minimization of photoconsistency

Reason for the strong noise: The data fidelity term finds several shifts with similar energy.



The correct disparity is difficult to determine pixel by pixel.

We need a regularization term to let neighboring pixel 'debate about the best disparity'!

Simple choice: Neighboring pixels are likely to have the same disparity (depth). There are discontinuities. $\rightarrow$ Use total variation (TV) regularization!

Variational approach to stereo matching

We would like to minimize

$$\underbrace{\int_\Omega |f_1(x,y) - f_2(x + v(x,y), y)|^2 \, dx \, dy}_{=\|F(v)\|_2^2} + \alpha \underbrace{\int_\Omega |\nabla v(x,y)| \, dx \, dy}_{=TV(v)}$$

for v, where F is a highly nonlinear operator.

What are possible strategies to solve

$$\min_v \|F(v)\|_2^2 + \alpha TV(v)?$$

Field of nonlinear optimization (with a special structure of the problem). Prominent strategies include regularized versions of Newton's method (approximating $\|F(v)\|^2$ with a second order Taylor expansion), the Gauss-Newton or Levenberg-Marquardt method (approximating $F(v)$ by a first order Taylor expansion), further quasi-Newton methods such as BFGS, and different versions of preconditioned or conjugate gradient (descent) methods.

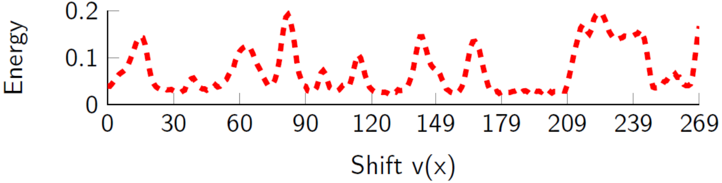
Variational approach to stereo matching

Let us limit ourselves to the plain gradient descent method for now. We use a smooth approximation TV_ϵ to TV and update

$$v^{k+1} = v^k - \tau^k \left(2(\nabla F(v^k))^T F(v^k) + \alpha \nabla TV_\epsilon(v^k)\right)$$

with τ^k small enough for the energy to decrease. Note that $\nabla F(v^k))$ denotes the Jakobian matrix of F at v^k. In our stereo matching example $\nabla F(v^k))$ would be a diagonal matrix (why?).

Problem: The Jakobian $\nabla F(v^k))$ often provides bad information about how to alter v - one quickly runs into bad local minima.



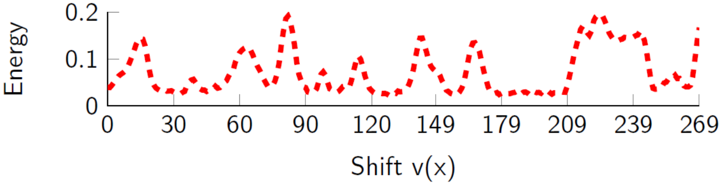
Coarse to fine strategy

Common approach: **Coarse-to-fine optimization**

1. Subsample the images f_1 and f_2 (starting with a large factor, e.g. of 10).
2. Run the gradient descent algorithm to convergence to obtain a guess v_k.
3. Use a simple interpolation scheme to upsample v_k by a small factor (e.g. 1.3).
4. Compute the subsampling of f_1 and f_2 to the same size as the upsampled version of v_k.
5. Go back to 2 and repeat until v_k is of the same size as the input images f_1 and f_2.

Further “tricks” have been considered in the literature, e.g. by median-filtering intermediate results.

Smooth energy approximations



Further ideas ways to overcome bad local minima: Smoothly approximate the 1D energies, e.g. by using parabolas and iteratively update the parabola fit, see for instance Kuschk, Cremers, *Fast and Accurate Large-scale Stereo Reconstruction using Variational Methods*, 2013.

A different strategy

Remember the previous topic we discussed: We saw that convex relaxation strategies are quite powerful in solving multi-region segmentation problems.

Interesting idea: We can approximate the solution to any problem of the form

$$\min_{v:\Omega\to[a,b]} \int_\Omega \rho(x, v(x))\, dx + \alpha TV(v)$$

by a multilabel segmentation problem! How?

Discretize the range of v!

For

$$\tilde{v} : \Omega \to \{a = t^0, \cdots, t^L = b\}$$

we may as well look for the indicator functions $\boldsymbol{u}_l : \Omega \to \{0, 1\}$ of the regions where $\tilde{v}(x) = t^l$, i.e. determine $\boldsymbol{u}_l$ such that

$$\boldsymbol{u}_l(x) = \begin{cases} 1 & \text{if } \tilde{v}(x) = t^l, \\ 0 & \text{otherwise.} \end{cases}$$

Reformulation of the data term

How can we express the data term

$$\int_\Omega \rho(x, \tilde{v}(x))\ dx$$

for $\tilde{v} : \Omega \to \{a = t^0, \cdots, t^L = b\}$ in terms of the $\boldsymbol{u}_l$?

If $\tilde{v}(x) = t^k$ then the costs at this x are

$$\begin{aligned}\rho(x, \tilde{v}(x)) &= \rho(x, t^k)\\ &= \boldsymbol{u}_k(x)\rho(x, t^k)\\ &= \sum_{l=0}^{L} \boldsymbol{u}_l(x)\rho(x, t^k)\\ &= \langle \boldsymbol{u}(x), \boldsymbol{\rho}(x)\rangle,\end{aligned}$$

where $\boldsymbol{\rho}(x) \in \mathbb{R}^{L+1}$ is the vector $\boldsymbol{\rho}(x) = (\rho(x, t^0), \cdots, \rho(x, t^L))^T$.

Hence, we can express the data term via

$$H(\boldsymbol{u}) = \int_\Omega \langle \boldsymbol{u}(x), \boldsymbol{\rho}(x)\rangle\ dx.$$

Reformulation of the data term

Sanity check: What happens if we minimize

$$H(\boldsymbol{u}) = \sum_{x \in \Omega} \langle \boldsymbol{u}(x), \rho(x) \rangle$$

for $\boldsymbol{u}$ subject to $\boldsymbol{u}(x) \in \{0, 1\}$ and $\sum_l \boldsymbol{u}_l(x) = 1$ for all x?

We will obtain $\boldsymbol{u}_l(x) = 1$ for some index l with $\rho(x, t^l) \leq \rho(x, t^k)$ for all k, and $\boldsymbol{u}_j(x) = 0$ for all $j \neq l$.

Since $\boldsymbol{u}_l(x)$ is the indicator function for the region where $v(x) = t^l$, we determined the disparity with minimal costs as expected/desired.

How about the regularization?

Finding a good regularization

One option: TV on $\boldsymbol{u}_l$ provides some kind of smoothness of the corresponding regions and can therefore also be used to provide smoothness of the final result.

Alternative: Derive a good convex formulation! Based on the idea of rephrasing the stereo matching problem as a multilabel segmentation problem we have the identification

$$v(x) = t^l \qquad \leftrightarrow \qquad \boldsymbol{u}(x) = e_l.$$

Knowing this identification, can we formulate a regularizer for $\boldsymbol{u}$ that behaves like $TV(v)$?

Define $\phi : \mathbb{R}^{L+1\times 2} \to \mathbb{R}$:

$$\phi(\boldsymbol{d}(x)) = \begin{cases} |t^i - t^j| \|v\|_2 & \text{if } \exists i,j,v \in \mathbb{R}^2 : \ \boldsymbol{d}(x) = (e_i - e_j)v^T \\ \infty & \text{otherwise.} \end{cases}$$

Finding a good regularization

We will then define

$$R(\boldsymbol{u}) = \int_{\Omega} \phi^{**}(\nabla \boldsymbol{u}(x))\ dx$$

to be our regularization.

The function

$$\phi(\boldsymbol{d}) = \begin{cases} |t^i - t^j| \|v\|_2 & \text{if } \exists i, j, v \in \mathbb{R}^2: \ \boldsymbol{d} = (e_i - e_j) v^T \\ \infty & \text{otherwise.} \end{cases}$$

can be written as $\phi(\boldsymbol{d}) = \min_{i,j} \phi_{i,j}(\boldsymbol{d})$, where

$$\phi_{i,j}(\boldsymbol{d}) = \begin{cases} |t^i - t^j| \|v\|_2 & \text{if } \exists v \in \mathbb{R}^2: \boldsymbol{d} = (e_i - e_j) v^T, \\ \infty & \text{otherwise.} \end{cases}$$

As we have seen before, it holds that

$$\phi(\boldsymbol{d}) = \min_{i,j} \phi_{i,j}(\boldsymbol{d}) \quad \Rightarrow \quad \phi^*(\boldsymbol{p}) = \max_{i,j} \phi^*_{i,j}(\boldsymbol{p}).$$

Finding a good regularization

Thus it sufficies to compute

$$\begin{aligned}
\phi^*_{i,j}(\boldsymbol{p}) &= \sup_{\boldsymbol{d}\in\mathbb{R}^{L+1\times 2}} \langle \boldsymbol{d}, \boldsymbol{p}\rangle_F - \phi_{i,j}(\boldsymbol{d}) \\
&= \sup_v \langle (e_i - e_j)v^T, \boldsymbol{p}\rangle_F - |t^i - t^j|\|v\|_2 \\
&= \sup_v \langle v, (e_i - e_j)^T \boldsymbol{p}\rangle - |t^i - t^j|\|v\|_2 \\
&= \sup_{\|v\|} \|(e_i - e_j)^T \boldsymbol{p}\|_2 \|v\|_2 - |t^i - t^j|\|v\|_2 \\
&= \begin{cases} \infty & \text{if } \|(e_i - e_j)^T \boldsymbol{p}\|_2 > |t^i - t^j| \\ 0 & \text{otherwise.} \end{cases}
\end{aligned}$$

Therefore

$$\phi^{**}(\boldsymbol{d}) = \sup_{\boldsymbol{p}\in K} \langle \boldsymbol{p}, \boldsymbol{d}\rangle_F$$

with

$$K = \{\boldsymbol{p} \in \mathbb{R}^{L+1\times 2} \mid \|\boldsymbol{p}_{i,:} - \boldsymbol{p}_{j,:}\|_2 \leq |t^i - t^j|, \;\; \forall i,j \in \{0, \cdots, L\}\}$$

Constraint set reduction

We can furthermore reduce the constraint set as follows:

$$K = \{\boldsymbol{p} \in \mathbb{R}^{L+1 \times 2} \mid \|\boldsymbol{p}_{i,:} - \boldsymbol{p}_{j,:}\|_2 \leq |t^i - t^j|, \ \forall i, j \in \{0, \cdots, L\}\}$$
$$= \{\boldsymbol{p} \in \mathbb{R}^{L+1 \times 2} \mid \|\boldsymbol{p}_{i,:} - \boldsymbol{p}_{i+1,:}\|_2 \leq |t^i - t^{i+1}|, \ \forall i \in \{0, \cdots, L\}\}.$$

Finally, one needs to know that terms of the form

$$\phi^{**}(\boldsymbol{d}) = \sup_{\boldsymbol{p} \in K} \langle \boldsymbol{p}, \boldsymbol{d} \rangle_F$$

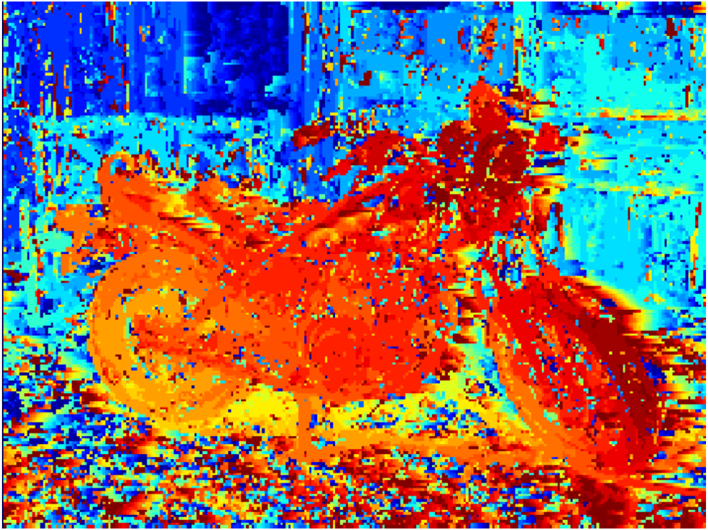
for a convex set K which one can easily project on, can be optimized efficiently.

Optimizing the convex multilabel-problem

$$\min_{\boldsymbol{u}} \int_\Omega \langle \boldsymbol{u}(x), \rho(x) \rangle \, dx + \alpha \int_\Omega \phi^{**}(\nabla \boldsymbol{u}(x)) \, dx$$

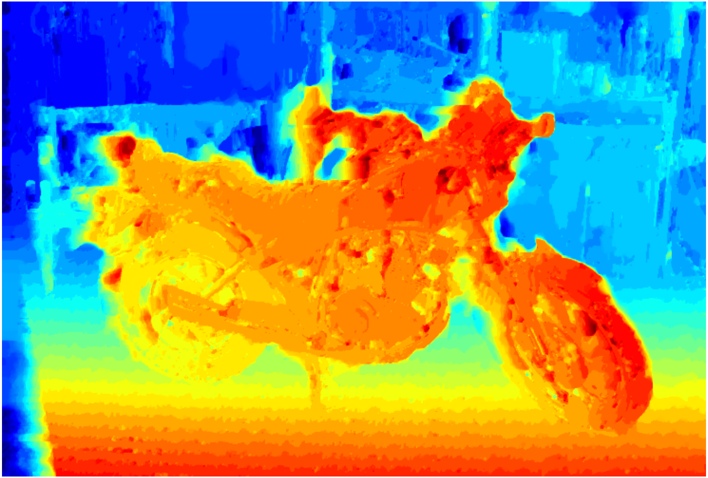
yields good stereo matching results independent of the initialization, inpedendent of the algorithm used for the minimization, and does not require a large number of heuristics!

Numerical results



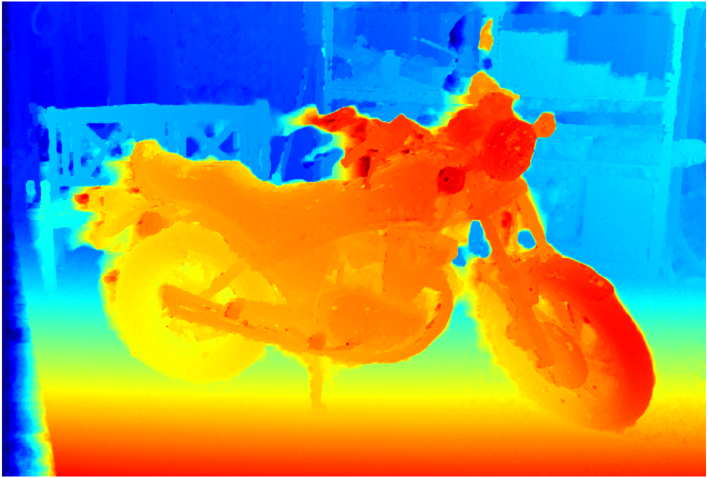
Pointwise minimum

Numerical results



Convex multilabel problem with $L + 1 = 32$

Numerical results



Convex multilabel problem with $L + 1 = 256$

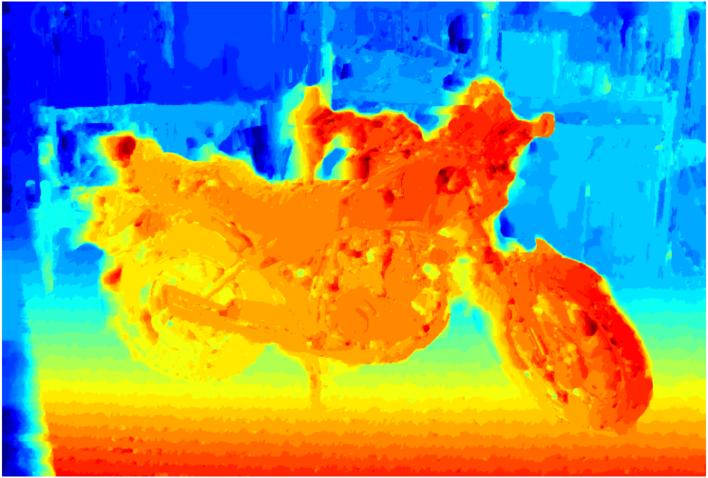
Summary and Outlook

- Reformulating stereo-matching as a convex multilabel problem has many nice properties.
- No assumptions that are specific to stereo matching were used - one can use the same construction for any problem of the form

$$\min_{u:\Omega\to\mathbb{R}} \int_\Omega \rho(x, u(x))\, dx + \alpha TV(x)$$

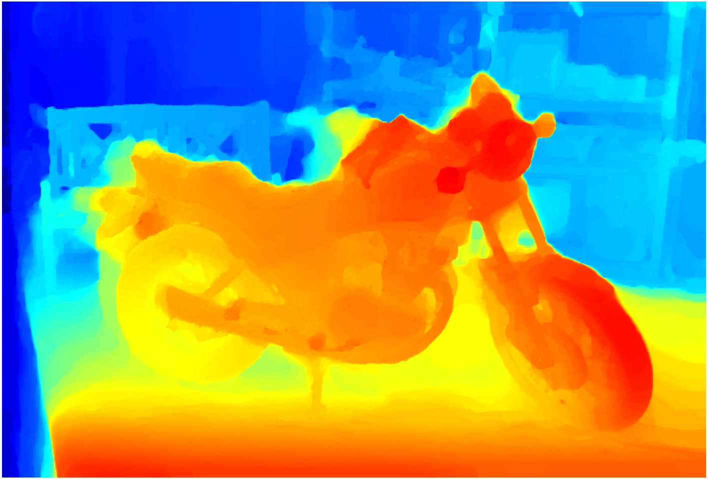
- Drawback: Computationally expensive and memory intense!
- Accuracy is only acceptable if sufficiently many labels are used.
- The latter is understandable considering that we started with the discretization of the range of $\boldsymbol{u}$ to $\{t^0, ..., t^L\}$.
- Our recent work *Sublabel–Accurate Relaxation of Nonconvex Energies*, CVPR 2016, showed how to make a similar construction without discretizing the range.

Numerical results



Convex multilabel problem with $L + 1 = 32$

Numerical results



Sublabel-accurate multilabeling with $L + 1 = 4$

Optical flow

Estimating the motion between frames of a video

Given two images f_1 and f_2, can we determine the movement of each point in f_1 to obtain f_2?

Applications:

- Changing the frame rate of a movie
- Motion segmentation
- Video super resolution ($\rightarrow$ Jonas)
- Object tracking
- Velocity estimation / prediction

If you put tracking and velocity estimation together with good hardware, you can do amazing things, see `https://www.youtube.com/watch?v=tIIJME8-au8`.

Estimating the motion between frames of a video

What is the abstract/mathematical problem?

Similar to stereo matching: We need to state what we mean by "determine the movement of each point in f_1 to obtain f_2". The only information we have is color, such that obtaining the true motion information will not necessarily be possible.

Modeling assumption similar to stereo vision: The same object has similar image properties in the two successive frames, e.g. the same color. The latter motivates to find

$$\min_{v:\Omega\to\mathbb{R}^2} \int_\Omega |f_1(x,y) - f_2(x + v_1(x,y), y + v_2(x,y))|^2 \, dx \, dy$$

Structurally the same as stereo matching, but our unknown has two components v_1 and v_2 at each point (x,y).

Goal: Win

`https://www.youtube.com/watch?v=ZmiBI4tPk_o`

Estimating the motion between frames of a video

Similar to stereo matching, any minimization of

$$\min_{v:\Omega\to\mathbb{R}^2} \int_\Omega |f_1(x,y) - f_2(x+v_1(x,y), y+v_2(x,y))|^2 \, dx \, dy$$

leads to very noisy results. Additionally, brute force search strategies are significantly more expensive!

Let us use $\boldsymbol{x} = (x,y) \in \mathbb{R}^2$, i.e.

$$\min_{v:\Omega\to\mathbb{R}^2} \int_\Omega |f_1(\boldsymbol{x}) - f_2(\boldsymbol{x}+v(\boldsymbol{x}))|^2 \, d\boldsymbol{x}$$

The first methods of Lucase and Kanade, and Horn and Schunck (both 1981) relied on linearizations also known as first-order Taylor approximations, i.e. for a grayscale image $f_2 : \Omega \to \mathbb{R}$ use

$$f_2(\boldsymbol{x}+v(\boldsymbol{x})) \approx f_2(\boldsymbol{x}+\tilde{v}(\boldsymbol{x})) + \langle \nabla f_2(\boldsymbol{x}+\tilde{v}(\boldsymbol{x})), v(\boldsymbol{x}) - \tilde{v}(\boldsymbol{x})\rangle$$

where $\tilde{v}(\boldsymbol{x})$ is an initial guess for v, e.g. $\tilde{v}(\boldsymbol{x}) = 0$.

Minimizing the data term with linearizations

Consider the linear equation

$$f_1(\boldsymbol{x}) = f_2(\boldsymbol{x} + \tilde{v}(\boldsymbol{x})) + \langle \nabla f_2(\boldsymbol{x} + \tilde{v}(\boldsymbol{x})), v(\boldsymbol{x}) - \tilde{v}(\boldsymbol{x})\rangle$$

Bad news for grayscale images: One equation, two unknowns.

Approach by Lucas and Kanade: Use patches, e.g. minimize

$$\left\| \begin{pmatrix} (f_1 - f_2)(i,j) \\ (f_1 - f_2)(i-1,j) \\ (f_1 - f_2)(i,j-1) \\ (f_1 - f_2)(i+1,j) \\ (f_1 - f_2)(i,j+1) \end{pmatrix} - \begin{pmatrix} \partial_x f_2(i,j) & \partial_y f_2(i,j) \\ \partial_x f_2(i-1,j) & \partial_y f_2(i-1,j) \\ \partial_x f_2(i,j-1) & \partial_y f_2(i,j-1) \\ \partial_x f_2(i+1,j) & \partial_y f_2(i+1,j) \\ \partial_x f_2(i,j+1) & \partial_y f_2(i,j+1) \end{pmatrix} \begin{pmatrix} v_1(i,j) \\ v_2(i,j) \end{pmatrix} \right\|^2$$

Approach by Horn and Schunck: Use regularization, e.g. minimize

$$\|f_1 - f_2 - \langle \nabla f_2, v\rangle\|_2^2 + \alpha\|\nabla v_1\|^2 + \alpha\|\nabla v_2\|^2$$

where in both of the above approaches we used $\tilde{v} = 0$.

State-of-the-art variational methods

More sophisticated variational methods

$$\int_\Omega |f_1(\boldsymbol{x}) - f_2(\boldsymbol{x} + v(\boldsymbol{x}))| \, d\boldsymbol{x} \tag{1}$$

$$+ \beta \int_\Omega |\nabla f_1(\boldsymbol{x}) - \nabla f_2(\boldsymbol{x} + v(\boldsymbol{x}))| \, d\boldsymbol{x} \tag{2}$$

$$+ \alpha \int_\Omega \|g(f_2)\nabla v(\boldsymbol{x})\|_p \, d\boldsymbol{x} \tag{3}$$

- In (1): No square $\rightarrow$ more robust to outliers
- In (2): Gradient-constancy $\rightarrow$ invariant to illumination changes like $f_1 \leftarrow f_1 + c$.
- In (3):
  - No square $\rightarrow$ allows jumps (=TV)
  - Reweight regularization based on values of f_2, e.g.

$$g(f_2)(\boldsymbol{x}) = \exp(-\kappa|\nabla f_2(\boldsymbol{x})|)$$

$\rightarrow$ Jumps in v likely coincide with edges in f_2.

References

Example references for the different "tricks":

- For (1): Zach, Pock, Bischof, *A Duality Based Approach for Realtime TV-L1 Optical Flow*, DAGM 2007.
- For (2): Zimmer, Bruhn, Weickert, *Optic flow in harmony*, IJCV 2011.
- For (3): Wedel et al., *Structure- and Motion-adaptive Regularization for High Accuracy Optic Flow*, ICCV 2009.

Despite all improvements in the variational formulation, the (nonconvex) minimization remains extremely challenging!

Strategies:

- Coarse-to-fine: a) Subsample images, b) solve for the flow using linearizations, c) upsample flow to next resolution
- Quadratic decoupling (similar to exercises)
- More generally: Sophisticated filterings of nearest neighbor fields (= brute force patch matching)
- Initializations from sparse feature matches

Example: EpicFlow

From Revaud et al., *EpicFlow: Edge-Preserving Interpolation of Correspondences for Optical Flow*, CVPR 2015.

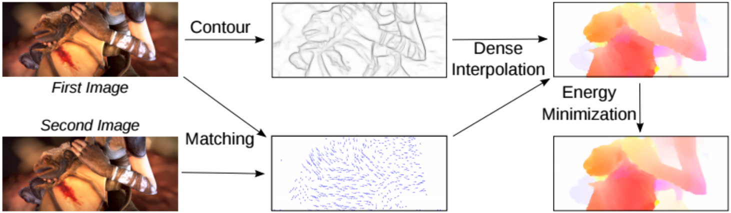
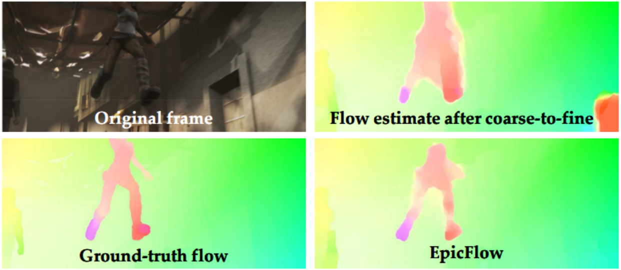


Figure 3. Overview of EpicFlow. Given two images, we compute matches using DeepMatching [34] and the edges of the first image using SED [15]. We combine these two cues to densely interpolate matches and obtain a dense correspondence field. This is used as initialization of a one-level energy minimization framework.



General remarks

Optical flow remains an extremely active area of research:
`http://sintel.is.tue.mpg.de/results`

Current strategies are powerful, but still somewhat disappointing from an optimization point of view: The initialization scheme is more important than the objective.

The structure of the problem is common in many areas of research, in particular in 3D vision, correspondence problems, simultaneous localization and matching etc.

The latter creates the urgent need to understand the underlying nonconvex optimization problems in more detail, and develop “more global” techniques without too many heuristics!

Almost anything can be posed as an optimization problem! You can do cool things if you master the art!
`https://www.youtube.com/watch?v=ohmajJTcpNk`

I’ll offer a course on the theory of convex optimization next semester. Nonconvex optimization is often “street-fighting convex optimization”!