Combinatorial Optimization in Computer Vision (IN2245)

Frank R. Schmidt
Csaba Domokos

Winter Semester 2015/2016

13. Graph Cut Approximation of Multilabel Problems 2

Local Optimization **3**
Multi-Label Problem 4
Metric Spaces 5
Metric Spaces and Convexity 6
Limitations of Mean Field Optimization 7

Binary Updates **8**
Long Range Moves 9
α-Expansion 10
Local Optimization via α-Expansion 11
α-Expansion Algorithm 12
$\alpha - \beta$ Swap 13
$\alpha - \beta$ Swap 14
Local Optimization via $\alpha - \beta$ Swap 15
$\alpha - \beta$-Swap Algorithm 16
Fusion Move 17

Fusion Move . . . 18
"Best of Two Worlds" . . . 19

Optical Flow **20**
Optical Flow . . . 21
Lucas-Kanade . . . 22
Horn-Schunck . . . 23
Optical Flow via Fusion Moves . . . 24
Literature . . . 25

Local Optimization 3 / 25

Multi-Label Problem

The multilabeling problem that we will address in the following is to find $x \in \mathcal{L}^n$ such that it minimizes the following energy

$$E(x) = \sum_{i=1}^{n} f_i(x_i) + \sum_{i=1}^{n} \sum_{j \in \mathcal{N}(i)} f_{ij} \cdot d(x_i, x_j)$$

In particular, we assume that for each label $\ell \in \mathcal{L}$ we have a data term $f_i(\ell)$ for each $i \in \{1, \ldots, n\}$. These data terms can be easily precomputed and are often motivated in a probabilistic fashion.

The pairwise term $f_{ij}\delta(x_i, x_j)$ depends on a precomputed measure f_{ij} that might depend on the image's gradient or some other information. In addition, we have a distance function $d(\ell_1, \ell_2)$ that measures the likelihood that the object corresponding to ℓ_1 is close to the object that corresponds to ℓ_2.

Metric Spaces

We already saw that we can find the global optimum if we use the *linear* or the *quadratic* model for $d(\cdot,\cdot)$. In fact, any convex model can be optimized using the **Ishikawa construction** of last lecture.

We will show that we can find an approximation of the multilabeling problem, if d is a metric,

$d\colon \mathcal{L} \times \mathcal{L} \to \mathbb{R}_0^+$ is called a **metric** if the following properties are satisfied

$$
\begin{aligned}
d(\ell_1,\ell_2) = 0 \Leftrightarrow \ell_1 = \ell_2 \quad & \text{for all } \ell_1,\ell_2 \in \mathcal{L} \quad & \text{(Positive Definite)}\\
d(\ell_1,\ell_2) = d(\ell_2,\ell_1) \quad & \text{for all } \ell_1,\ell_2 \in \mathcal{L} \quad & \text{(Symmetry)}\\
d(\ell_1,\ell_3) \leqslant d(\ell_1,\ell_2) + d(\ell_2,\ell_3) \quad & \text{for all } \ell_1,\ell_2,\ell_3 \in \mathcal{L} \quad & \text{(Triangle Inequality)}
\end{aligned}
$$

We call $(\mathcal{L}, d)$ a **metric space** if $d\colon \mathcal{L} \times \mathcal{L} \to \mathbb{R}$ is a metric.

Metric Spaces and Convexity

The *Potts model* is a metric as well as any L^p model for $0 < p \leqslant 1$.

For each $p > 1$ there is an L^p **metric** which is the p^{th} root of the L^p model. Since for the totally ordered set $\mathcal{L} \subset \mathbb{Z}$ there is no difference between the L^p metrics for all $p > 1$, we are usually using the L^p model instead.

For a general label space, the L^p model is only convex for $p \geqslant 1$.

For the binary label space, every L^p model coincides with the Potts model. In this case, the Potts model is convex and can be globally optimized with GraphCut.

Any good approximation scheme to minimize the Potts model should therefore find the global optimum of the binary Potts model.

Limitations of Mean Field Optimization

The mean field optimization looks at the objective function and optimizes it with respect to one variable while keeping the other variables fixed.

This approach is in general problematic, because there may be a situation where we can only improve the energy by changing several variables at the same time.

Considering $f\colon \mathbb{Z}^2 \to \mathbb{R}$ with $f(x,y) = x^2 + 2(x-y)^2$, the mean field optimization could not improve upon the solution $(x,y) = (1,1)$.

$$f(0,1) = 2 \qquad f(1,1) = 1 \qquad f(2,1) = 6$$
$$f(1,0) = 3 \qquad f(1,1) = 1 \qquad f(1,2) = 3$$

Note that this function f is convex and any continuous optimization method could find the global optimum of $f\colon \mathbb{R}^2 \to \mathbb{R}$ by a gradient descent approach.

Long Range Moves

Since graph cuts will always compute the global energy of a submodular energy, one might be interested in formulating a binary submodular sub-problem that can be solved with graph cut.

Such an approach combines the main idea of mean field optimization, *i.e.*, local improvements with the insight that graph cut optimization can change the label of multiple variables at the same time.

We will discuss the following three different approaches

- α **Expansion** allows each variable to either keep its current label or to change it to the label $\alpha \in \mathcal{L}$. As a result, the region of α expands.
- $\alpha - \beta$ **Swap** only changes those pixels that are labeled $\ell \in \{\alpha, \beta\}$. Each of these variables can choose between α and β.
- **Fusion Move** starts with two different labelings $x, y \in \mathcal{L}^n$. Each variable chooses then for itself either the label from x or y. Both, α expansion and $\alpha - \beta$ swap can be seen as special cases of the fusion move.

α-Expansion

Instead of considering the energy

$$E(x) = \sum_{i=1}^{n} f_i(x_i) + \sum_{i=1}^{n} \sum_{j \in \mathcal{N}(i)} f_{ij} \cdot d(x_i, x_j) \qquad x \in \mathcal{L}^n$$

α **expansion** considers a different energy with respect to $y \in \mathbb{B}^n$.

Given current labeling $z \in \mathcal{L}^n$ and label $\alpha \in \mathcal{L}$, we like to minimize

$$E^{z,\alpha}(y) = \sum_{i=1}^{n} f_i^{z,\alpha}(y_i) + \sum_{i=1}^{n} \sum_{j \in \mathcal{N}(i)} f_{ij} \cdot d_{ij}^{z,\alpha}(y_i, y_j)$$

$$f_i^{z,\alpha}(0) = f_i(z_i) \qquad f_i^{z,\alpha}(1) = f_i(\alpha)$$
$$d_{ij}^{z,\alpha}(0,0) = d(z_i, z_j) \qquad d_{ij}^{z,\alpha}(0,1) = d(z_i, \alpha)$$
$$d_{ij}^{z,\alpha}(1,0) = d(\alpha, z_j) \qquad d_{ij}^{z,\alpha}(1,1) = d(\alpha, \alpha)$$

Local Optimization via α-Expansion

The mean field optimization finds the global optimal with respect to $|\mathcal{L}|$ different labelings per iteration. The α expansion optimization on the other hand seeks to find the global optimum with respect to $2^{|n|}$ different labels.

If $d\colon \mathcal{L} \times \mathcal{L} \to \mathbb{R}$ is a metric, the energy $E^{z,\alpha}$ is submodular for every $z \in \mathcal{L}^n$ and $\alpha \in \mathcal{L}$, because

$$\begin{aligned} d_{ij}^{z,\alpha}(0,0) + d_{ij}^{z,\alpha}(1,1) =& d(z_i, z_j) + d(\alpha, \alpha) = d(z_i, z_j) \\ \leqslant& d(z_i, \alpha) + d(\alpha, z_j) \\ =& d_{ij}^{z,\alpha}(0,1) + d_{ij}^{z,\alpha}(1,0) \end{aligned}$$

Since the Potts model is a metric one can use α expansion to find a local minimum of the original energy. If we use a model that is not a metric, we can use the roof duality to change some of the involved variables. In most cases this works better than the mean field optimization.

α-Expansion Algorithm

Summarizing all observations, the α expansion algorithm works as follows

1. Choose an initial labeling $x \in \mathcal{L}^n$ and set $z := x$.
2. For all $\alpha \in \mathcal{L}$ do
   (a) Find $y \in \operatorname{argmin} E^{z,\alpha}(\cdot)$.
   (b) Set $z_i := \alpha$ for all i such that $y_i = 1$.
3. If $E(z) < E(x)$ set $x := z$ and go to Step 2.
4. Return the local optimum $x \in \mathcal{L}^n$.

α expansion computes at least $|\mathcal{L}|$ different graph cuts. This may take a lot of time if the label space is big.

Since each α expansion step considers only a binary problem, the overall memory consumption is independent of $|\mathcal{L}|$.

$\alpha - \beta$ **Swap**

Instead of considering the energy

$$E(x) = \sum_{i=1}^{n} f_i(x_i) + \sum_{i=1}^{n} \sum_{j \in \mathcal{N}(i)} f_{ij} \cdot d(x_i, x_j) \qquad x \in \mathcal{L}^n$$

$\alpha - \beta$ **swap** considers a different energy with respect to $y \in \mathbb{B}^m$ where $m \leqslant n$.

Given $z \in \mathcal{L}^n$ and $\alpha, \beta \in \mathcal{L}$ let

$$\mathcal{X}_\alpha = \{1 \leqslant i \leqslant n | z_i = \alpha\} \qquad \mathcal{X}_\beta = \{1 \leqslant i \leqslant n | z_i = \beta\} \qquad \mathcal{X} = \mathcal{X}_\alpha + \mathcal{X}_\beta$$

Without loss of generality, we have

$$\mathcal{X}_\alpha = \{1, \ldots, m_1\} \qquad \mathcal{X}_\beta = \{m_1 + 1, \ldots, m_1 + m_2\} \qquad \mathcal{X} = \{1, \ldots, m\}$$

$\alpha - \beta$ **Swap**

The energy we like to minimize is

$$\hat{E}^{z,\alpha,\beta}(x) = \sum_{i=1}^{m} \hat{f}_i(x_i) + \sum_{i=1}^{m} \sum_{\substack{j \in \mathcal{N}(i), \\ j \leqslant m}} f_{ij} \cdot \hat{d}(x_i, x_j) \qquad x \in \mathbb{B}^m$$

with

$$\begin{aligned} \hat{f}_i(0) &= f_i(\alpha) + \sum_{j \in \mathcal{N}(i), j > m} [f_{ij} \cdot d(\alpha, z_j) + f_{ji} \cdot d(z_j, \alpha)] \\ \hat{f}_i(1) &= f_i(\beta) + \sum_{j \in \mathcal{N}(i), j > m} [f_{ij} \cdot d(\beta, z_j) + f_{ji} \cdot d(z_j, \beta)] \end{aligned}$$

$$\begin{aligned} \hat{d}(0,0) &= d(\alpha, \alpha) & \hat{d}(0,1) &= d(\alpha, \beta) \\ \hat{d}(1,0) &= d(\beta, \alpha) & \hat{d}(1,1) &= d(\beta, \beta) \end{aligned}$$

Local Optimization via $\alpha - \beta$ Swap

If $d\colon \mathcal{L} \times \mathcal{L} \to \mathbb{R}$ is a metric, the energy $\hat{E}^{z,\alpha,\beta}$ is submodular. Thus, we can always use $\alpha - \beta$ swap if we can use α expansion.

Moreover, we can use $\alpha - \beta$ swap if $d\colon \mathcal{L} \times \mathcal{L} \to \mathbb{R}$ is just a **premetric**, *i.e.*,

$$d(\ell_1, \ell_2) \geqslant 0 \qquad \text{for all } \ell_1, \ell_2 \in \mathcal{L}$$
$$d(\ell, \ell) = 0 \qquad \text{for all } \ell \in \mathcal{L}$$

Hence, we can use $\alpha - \beta$ swap in situation where α expansion can only be used with the help of the roof duality.

$\alpha - \beta$ swap like α expansion is more powerful than the naive mean field optimization. In particular, both methods will compute the global optimum if a binary submodular energy has to be minimized.

$\alpha - \beta$-Swap Algorithm

Summarizing, the $\alpha - \beta$ swap algorithm works as follows

1. Choose an initial labeling $x \in \mathcal{L}^n$ and set $z := x$.
2. For all $\alpha, \beta \in \mathcal{L}$ do
   (a) Find $y \in \operatorname{argmin} \hat{E}^{z,\alpha,\beta}(\cdot)$.
   (b) Set $z_i := \alpha$ for all i such that $y_i = 0$.
   (c) Set $z_i := \beta$ for all i such that $y_i = 1$.
3. If $E(z) < E(x)$ set $x := z$ and go to Step 2.
4. Return the local optimum $x \in \mathcal{L}^n$.

$\alpha - \beta$ swap computes at least $\mathcal{O}(|\mathcal{L}|^2)$ different graph cuts. This may take a lot of time, even for moderately large label spaces.

Since there are $|\mathcal{L}|$ different α expansion moves, but $\mathcal{O}(|\mathcal{L}|^2)$ different $\alpha - \beta$ swap moves, one usually uses the $\alpha - \beta$ swap moves only in situations where α expansion cannot be used.

Fusion Move

Comparing α expansion and $\alpha - \beta$ swap they can be seen as special instances of a more general idea.

Given a labeling $z \in \mathcal{L}^n$ and $\alpha, \beta \in \mathcal{L}$, let us define the followng three labelings $x^{(\alpha)}, z^{(\alpha)}, z^{(\beta)} \in \mathcal{L}^n$ as

$$x_i^{(\alpha)} = \alpha,$$

$$z_i^{(\alpha)} = \begin{cases} \alpha & \text{if } z_i \in \{\alpha, \beta\} \\ z_i & \text{otherwise} \end{cases} \quad \text{and} \quad z^{(\beta)} = \begin{cases} \beta & \text{if } z_i \in \{\alpha, \beta\} \\ z_i & \text{otherwise} \end{cases}$$

for all $1 \leqslant i \leqslant n$.

To obtain a better labeling, α expansion combines the information of z and $x^{(\alpha)}$, while $\alpha - \beta$ swap uses $z^{(\alpha)}, z^{(\beta)}$. To combine two labelings in order to obtain a better labeling is called **fusion move**.

Fusion Move

Instead of considering the energy

$$E(x) = \sum_{i=1}^{n} f_i(x_i) + \sum_{i=1}^{n} \sum_{j \in \mathcal{N}(i)} f_{ij} \cdot d(x_i, x_j) \qquad x \in \mathcal{L}^n$$

fusion move considers a different energy with respect to $y \in \mathbb{B}^n$.

Given two different labeling $z^{(0)}, z^{(1)} \in \mathcal{L}^n$, we like to minimize

$$E^{z^{(0)},z^{(1)}}(y) = \sum_{i=1}^{n} f_i^{z^{(0)},z^{(1)}}(y_i) + \sum_{i=1}^{n} \sum_{j \in \mathcal{N}(i)} f_{ij} \cdot d_{ij}^{z^{(0)},z^{(1)}}(y_i, y_j)$$

$$f_i^{z^{(0)},z^{(1)}}(0) = f_i(z_i^{(0)}) \qquad f_i^{z^{(0)},z^{(1)}}(1) = f_i(z_i^{(1)})$$

$$d_{ij}^{z^{(0)},z^{(1)}}(0,0) = d(z_i^{(0)}, z_j^{(0)}) \qquad d_{ij}^{z^{(0)},z^{(1)}}(0,1) = d(z_i^{(0)}, z_j^{(1)})$$

$$d_{ij}^{z^{(0)},z^{(1)}}(1,0) = d(z_i^{(1)}, z_j^{(0)}) \qquad d_{ij}^{z^{(0)},z^{(1)}}(1,1) = d(z_i^{(1)}, z_j^{(1)})$$

“Best of Two Worlds”

While fusion move is a generalization of both, α expansion and $\alpha - \beta$ swap, it is difficult to use it in order to design a different local optimization framework.

α expansion can be seen as a version of **coordinate descent** with respect to the label space. Instead of changing one variable it changes one label.

$\alpha - \beta$ swap changes two labels simultaneously, but this comes at the cost of optimizing only a small subset of all variables.

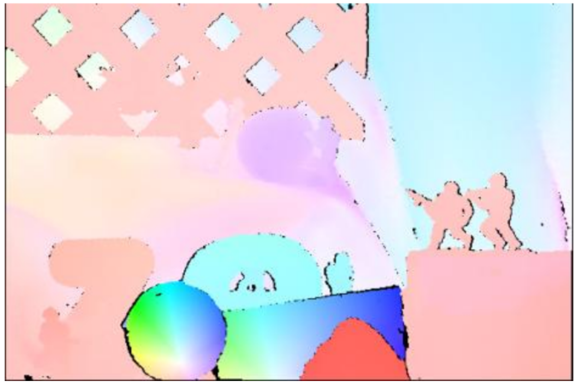
So far, no other “generic steps” have been introduced that could be applied to any multilabeling problem.

A common approach to use the fusion move is to pre-compute several labelings with different approaches and then to combine them with a fusion move. Thus, fusion can be seen as finding the “best of two worlds”.

Optical Flow

Image $I_0 \colon \Omega \to \mathbb{R}^3$

Image $I_1 \colon \Omega \to \mathbb{R}^3$



Flow $v \colon \Omega \to \mathbb{R}^2$

Given two images I_0 and I_1 of a video, we would like to detect the movements between these two images.

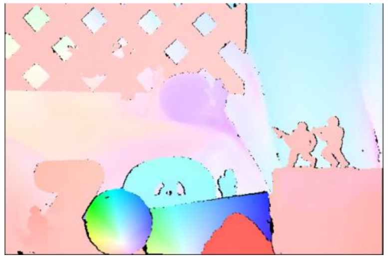
In other words, we are interested in a mapping $v \colon \Omega \to \mathbb{R}^2$ such that $I_1(x) \approx I_2(x + v(x))$. The vector field v is called the **optical flow**.

If we quantize $\mathbb{R}^2$, we obtain a finite label space and the optical flow v can be understood as a multilabeling of Ω.

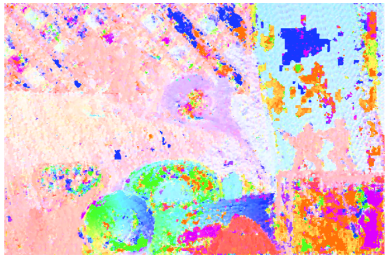
Lucas-Kanade

Image $I_0 : \Omega \to \mathbb{R}^3$

Image $I_1 : \Omega \to \mathbb{R}^3$



Optical Flow



Estimated Flow

If we assume that the color remains constant during the video, we have

$$0 = \frac{d}{dt} I(t, x + v(t,x)) = \frac{\partial}{\partial t} I(t, x + v(t,x)) + \nabla_x I(t, x + v(t,x)) \frac{\partial}{\partial t} v(t,x)$$

Reformulating this in a discrete setting means

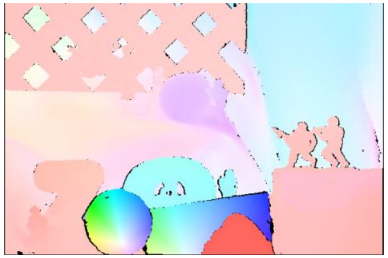
$$[I_1(x) - I_0(x)] + \left\langle \nabla I_{\frac{1}{2}}(x), v(x) \right\rangle = 0$$

These linear constraints lead to an over-determined system of linear equations and Lucas and Kanade proposed to solve it using least squares.

Horn-Schunck

Image $I_0 : \Omega \to \mathbb{R}^3$

Image $I_1 : \Omega \to \mathbb{R}^3$



Optical Flow

Estimated Flow

Instead of solving

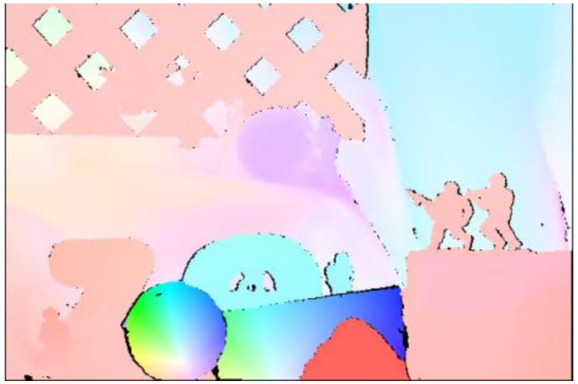
$$0 = \frac{\partial}{\partial t} I + \left\langle \nabla I_t, \frac{\partial}{\partial t} v \right\rangle$$

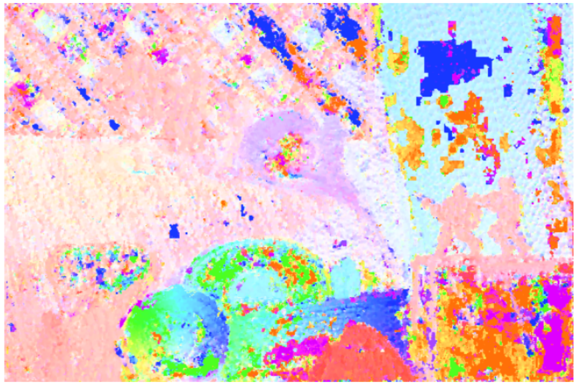
the method of Horn und Schunck tries to minimize the following energy

$$E(v) = \int \left(\frac{\partial}{\partial t} I + \left\langle \nabla I_t, \frac{\partial}{\partial t} v \right\rangle \right)^2 + \lambda \left\| \nabla v \right\|^2$$

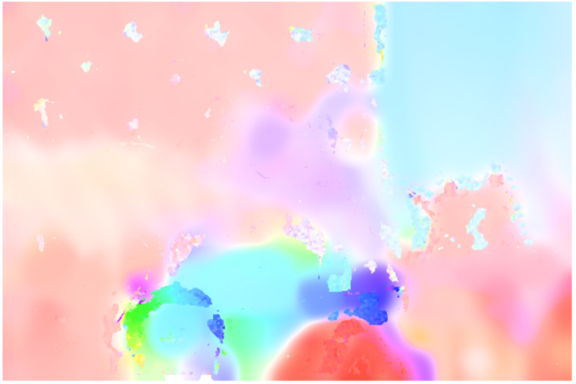
This energy is usually minimized by a variational framework.

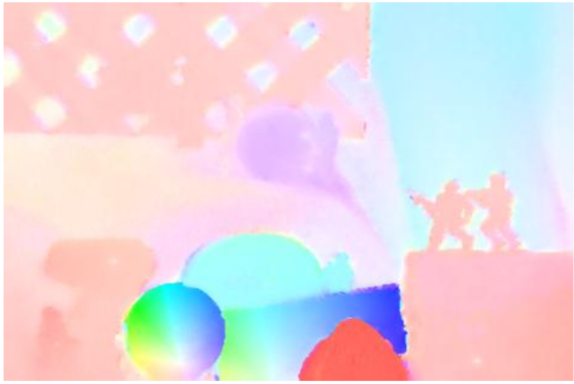
Optical Flow via Fusion Moves


Ground Truth


Lucas-Kanade


Horn-Schunck


1st Fusion


Final Fusion

After each fusion a local (variational) optimization is been performed and we can restart the fusion process.

Literature

Graph Cut Approximation

- Boykov, Veksler, Zabih, *Markov Random Fields with Efficient Approximations*, 1998, IEEE CVPR, 648–655.

Optical Flow

- Lucas, Kanade, *An iterative image registration technique with an application to stereo vision*, 1981, Image Understanding, 121–130.
- Horn, Schunck, *Determining Optical Flow*, 1981, Artificial Intelligence 17, 185–203.
- Lempitsky, Roth, Rother, *FusionFlow: Discrete-Continuous Optimization for Optical Flow Estimation*, 2008, IEEE CVPR, 1–8.