Chapter 5
Vectorial Total Variation
and Multilabel Problems

Daniel Cremers and Bastian Goldlücke
Computer Vision Group
Technical University of Munich

Thomas Pock
Institute for Computer Graphics and Vision
Graz University of Technology
Overview

1 Vectorial Total Variation
   Reminder: TV and binary segmentation
   Generalizations of the total variation
   Analytic properties
   Geometric properties

2 Multilabel segmentation
   The multilabel problem
   Regularization
   Domain relaxation
   Examples

3 Product Label Spaces
Overview

1 Vectorial Total Variation
   Reminder: TV and binary segmentation
   Generalizations of the total variation
   Analytic properties
   Geometric properties

2 Multilabel segmentation
   The multilabel problem
   Regularization
   Domain relaxation
   Examples

3 Product Label Spaces
Binary Segmentation

Region $\Omega_0$ (background)

Region $\Omega_1$ (flower)
Binary Segmentation

Find binary labeling $u : \Omega \rightarrow \{0, 1\}$ which minimizes

$$\int_{\Omega} |\nabla u|_2 \, dx + \int_{\Omega} c_1 \cdot u \, dx.$$  

- $|\nabla u|_2$ represents the length of the interface $\Sigma$.  
- $c_1 \cdot u$ represents the assignment cost.

$c_1(x) = \text{cost of assigning } \text{“1” to the point } x \in \Omega.$
Binary Segmentation

Find binary labeling $u : \Omega \rightarrow \{0, 1\}$ which minimizes

$$\int_{\Omega} |\nabla u|_2 \, dx + \int_{\Omega} c_1 \cdot u \, dx.$$

- length of interface $\Sigma$
- assignment cost

$c_1(x) = \text{cost of assigning “1” to the point } x \in \Omega.$

Can be minimized globally (Chan, Esedoglu and Nikolova 2006), as shown in Chapter 3 of the tutorial.
First goal in this chapter: introduce **total variation for vector-valued functions** which has a similar geometric interpretation, and can be used to define a regularizer for multi-label problems.
Reminder: scalar total variation

For a greyscale image $u : \Omega \rightarrow \mathbb{R}$ on a domain $\Omega \subset \mathbb{R}^m$, the scalar total variation (TV) is defined as

$$TV(u) = \int_{\Omega} |\nabla u|_2 \, dx = \sup_{\xi \in C^1_c(\Omega, \mathbb{E}(m))} \left\{ \int_{\Omega} u \, \text{div}(\xi) \, dx \right\},$$

where $C^1_c(\Omega, \mathbb{E}(m))$ is the unit ball in $\mathbb{R}^m$.

Requirements for the generalization to $u : \Omega \rightarrow \mathbb{R}^n$:

- Definitions coincide for $n = 1$
- Dual formulation available, so that it is defined for non-differentiable functions
- Convex, closed $\implies$ efficient minimization algorithms available
- Important invariances and other properties of scalar TV still satisfied
Sapiro’s general approach: image manifold

- The metric tensor of the image manifold \( u(\Omega) \) is given by
  \[
  (g_{\mu\nu}) = (Du)^T Du.
  \]

- The Eigenvector corresponding to the largest Eigenvalue \( \lambda_1 \) gives the direction of the vectorial edge.

- \( n = 1 \): Equal to direction of the gradient \( \nabla u \), which is always orthogonal to the level lines.

Leads to family of possible definitions for the vectorial TV in the case \( m = 2 \), which is of the form

\[
TV_{SR}(u) := \int_{\Sigma} \varphi(\lambda_1, \ldots, \lambda_n) \, ds,
\]

where \( \varphi \) is a suitable scalar-valued function (Sapiro and Ringach, 1996).
## Generalizations with dual formulation

<table>
<thead>
<tr>
<th>Variant</th>
<th>Primal</th>
<th>Dual</th>
</tr>
</thead>
<tbody>
<tr>
<td>$TV_S(u)$</td>
<td>$\sum_{i=1}^{n} \int_{\Omega}</td>
<td>\nabla u_i</td>
</tr>
<tr>
<td></td>
<td></td>
<td>with $K_S = C^1_c(\Omega, E(m) \times \cdots \times E(m))$</td>
</tr>
<tr>
<td>$TV_F(u)$</td>
<td>$\int_{\Omega} |Du(x)|_F , dx$</td>
<td>$\sup_{(\xi_1, \ldots, \xi_n) \in K_F} \left{ \sum_{i=1}^{n} \int_{\Omega} u_i \div(\xi_i) , dx \right}$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>with $K_F = C^1_c(\Omega, E(n \cdot m))$</td>
</tr>
<tr>
<td>$TV_J(u)$</td>
<td>$\int_{\Omega} \sqrt{\lambda_1} , dx$</td>
<td>$\sup_{(\xi, \eta) \in K_J} \left{ \sum_{i=1}^{n} \int_{\Omega} u_i \div(\eta_i \xi) , dx \right}$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>with $K_J = C^1_c(\Omega, E(m) \times E(n))$</td>
</tr>
</tbody>
</table>

Comparison: Goldlücke and Cremers, CVPR 2010
For this tutorial chapter, we choose the Frobenius TV as the vectorial total variation. The primal definition for differentiable $u$ is

$$
\int_{\Omega} \| Du(x) \|_F \, dx = \int_{\Omega} \sqrt{\sum_{i,j} \left( \frac{\partial u_i}{\partial x_j} \right)^2} \, dx
$$

$$
= \int_{\Omega} \sqrt{\lambda_1 + \ldots + \lambda_n} \, dx.
$$

The latter equality can be checked by substituting the SVD of $Du$. This corresponds to the dual definition

$$
\sup_{(\xi_1, \ldots, \xi_n) \in K_F} \left\{ \sum_{i=1}^{n} \int_{\Omega} u_i \text{div}(\xi_i) \, dx \right\}
$$

with $K_F = C^1_c(\Omega, \mathbb{E}(n \cdot m))$. 
The analytic properties can be verified directly from the definition, as in the scalar case.

**Proposition**

- $\text{TV}_F$ is a semi-norm, in particular it is convex.
- $\text{TV}_F$ is lower semi-continuous (closed).
**Geometric properties**

TV$_F$ has a geometric property similar to the scalar TV with regard to curve length. It allows to construct very general regularizers for multilabel segmentation problems.

**Theorem**

Let $S \subset \Omega$ and $\bar{S} := \Omega \setminus S$. Furthermore, let $a, b \in \mathbb{R}^k$. Then

$$TV_F(a \, 1_S + b \, 1_{\bar{S}}) = |a - b|_2 \text{Per}(S).$$

Note that this is a generalization of the scalar case, since

$$TV(1_S) = TV(1 \cdot 1_S + 0 \cdot 1_{\bar{S}}) = |1 - 0|_2 \text{Per}(S) = \text{Per}(S).$$

see Lellmann et al., ICCV 2009
Overview

1. Vectorial Total Variation
   Reminder: TV and binary segmentation
   Generalizations of the total variation
   Analytic properties
   Geometric properties

2. Multilabel segmentation
   The multilabel problem
   Regularization
   Domain relaxation
   Examples

3. Product Label Spaces
Interpretation as **segmentation problem**

\[ \Omega = \Omega_1 \cup ... \cup \Omega_N \]
The multilabel problem

Interpretation as labeling problem

\[ g : \Omega \rightarrow \{ \gamma_1, \ldots, \gamma_N \} \]

- Label \( \gamma_0 \) (background)
- Label \( \gamma_1 \) (leaves)
- Label \( \gamma_2 \) (flower 1)
- Label \( \gamma_3 \) (flower 2)
The regularization penalty is proportional to the label distance times the length of the interface.

In this example $d(\gamma_1, \gamma_2) \cdot L(\Sigma)$
The regularization penalty is proportional to the label distance times the length of the interface.

In this example $d(\gamma_1, \gamma_2) \cdot L(\Sigma)$

Euclidean representation of the label distance:

- Each label $\gamma$ is represented by a point $a_\gamma \in \mathbb{R}^k$.
- Label distance $d(\gamma, \mu) = |a_\gamma - a_\mu|_2$. 
The regularization penalty is proportional to the label distance times the length of the interface.

In this example $d(\gamma_1, \gamma_2) \cdot L(\Sigma)$

Euclidean representation of the label distance:
- Each label $\gamma$ is represented by a point $a_\gamma \in \mathbb{R}^k$.
- Label distance $d(\gamma, \mu) = |a_\gamma - a_\mu|_2$.

Except for special cases of the metric, the problem is highly non-convex.
Convex relaxation of the multilabel problem

We assign an indicator function \( u_\gamma : \Omega \to \{0, 1\} \) to each label \( \gamma \):

\[
\sum_\gamma u_\gamma \text{ must be one!}
\]
We assign an indicator function \( u_\gamma : \Omega \to \{0, 1\} \) to each label \( \gamma \):

\[
\begin{align*}
    u_0 &= 1, \text{ all others zero} \\
    u_1 &= 1, \text{ all others zero} \\
    u_2 &= 1, \text{ all others zero} \\
    u_3 &= 1, \text{ all others zero}
\end{align*}
\]

\[
\sum_\gamma u_\gamma \text{ must be one!}
\]

Let the columns of the matrix \( A \) consists of the vectors \( a_\gamma \). Then the problem we need to solve is

\[
\arg\min_{u_\gamma: \Omega \to \{0,1\}, \sum_\gamma u_\gamma = 1} \left[ J(Au) + \sum_\gamma \int_\Omega c_\gamma u_\gamma \, dx \right].
\]

Solution with optimality bound via domain relaxation possible.

Lellmann et al. ICCV 2009
Domain relaxation and optimality bound

\( \hat{u} = \text{argmin}_{u \in U} E(u) \)

\( \hat{w} = \text{argmin}_{u \in C} E(u) \)

\( \pi(\hat{w}) = \text{projection of } \hat{w} \text{ onto } U \)

\( E(\hat{w}) \leq E(\hat{u}) \leq E(\pi(\hat{w})) \)

Optimality estimate:

\( |E(\hat{u}) - E(\pi(\hat{w}))| \leq |E(\hat{w}) - E(\pi(\hat{w}))| \)

\( \text{known a posteriori} \)
Domain relaxation and optimality bound

\[ \hat{u} = \arg\min_{u \in U} E(u) \]

\[ \hat{w} = \arg\min_{u \in C} E(u) \]
Domain relaxation and optimality bound

\[ \hat{u} = \arg\min_{u \in U} E(u) \]

\[ \hat{w} = \arg\min_{u \in C} E(u) \]

\[ \pi(\hat{w}) = \text{projection of } \hat{w} \text{ onto } U \]
Domain relaxation and optimality bound

\[ \hat{u} = \arg\min_{u \in U} E(u) \]

\[ \hat{w} = \arg\min_{u \in C} E(u) \]

\[ \pi(\hat{w}) = \text{projection of } \hat{w} \text{ onto } U \]

\[ \implies E(\hat{w}) \leq E(\hat{u}) \leq E(\pi(\hat{w})) \]
Domain relaxation and optimality bound

\[ \hat{u} = \arg\min_{u \in U} E(u) \]

\[ \hat{w} = \arg\min_{u \in C} E(u) \]

\[ \pi(\hat{w}) = \text{projection of } \hat{w} \text{ onto } U \]

\[ \implies E(\hat{w}) \leq E(\hat{u}) \leq E(\pi(\hat{w})) \]

Optimality estimate:

\[ \left| E(\hat{u}) - E(\pi(\hat{w})) \right| \leq \left| E(\hat{w}) - E(\pi(\hat{w})) \right|. \]

how close am I? known a posteriori
Important special cases

Euclidean representation of the label distance:

- Each label $\gamma$ is *represented* by a point $a\gamma \in \mathbb{R}^k$.
- Label distance $d(\gamma, \mu) = |a\gamma - a\mu|_2$.

Assume here the labels are numbered, $\Gamma = \{1, \ldots, k\}$.

- The case of *ordered labels*, $a\gamma = \gamma \in \mathbb{R}$ arises for example in depth reconstruction. As we have seen Chapter 3, it can be solved globally with functional lifting (a continuous version of the Ishikawa construction).

- The *Potts distance* can be modeled by representing each label by a unit vector $a\gamma = e\gamma$, different labelings are all penalized equally. In this case, $A$ is the identity matrix. We have seen some alternative convex relaxations in the last chapter.
Another example: Optic Flow

Color input images $l_0, l_1 : \Omega \rightarrow \mathbb{R}^3$:

Label each pixel in $l_0$ with a flow vector in $\Gamma \subset \mathbb{R}^2$, choose representation $a_\gamma = \gamma$.

Cost function compares pointwise pixel colors in the images:

$$c_\gamma(x) = \|l_0(x) - l_1(x + \gamma)\|_2$$
Overview

1 Vectorial Total Variation
   Reminder: TV and binary segmentation
   Generalizations of the total variation
   Analytic properties
   Geometric properties

2 Multilabel segmentation
   The multilabel problem
   Regularization
   Domain relaxation
   Examples

3 Product Label Spaces
The optic flow label space is a product space.

Each red dot requires one indicator function - too many. Can we exploit the special structure of the label space?
Reduction idea for product label spaces

\[ \Lambda_1 \]

\[ \begin{array}{cccc}
0 & 0 & 0 & 0 \\
\vdots & \vdots & \vdots & \vdots \\
0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 \\
\end{array} \]

\[ \Lambda_2 \]

\[ \begin{array}{cccc}
0 & \ldots & 1 & \ldots & 0 \\
\end{array} \]
Reduction idea for product label spaces

Indicator functions are products $u_\gamma = u^{1}_{\lambda_1} \cdot u^{2}_{\lambda_2}$. 
Product function and convex envelope

Product function $m(x_1, x_2) = x_1x_2$
Product function and convex envelope

Product function \( m(x_1, x_2) = x_1 x_2 \)

Convex envelope \( \text{co}(m) \)

Exchanging multiplication with its convex hull does not change the location of the binary minimizer!
Final problem is convex of the form

\[
\argmin_{u_i: \Omega \rightarrow [0,1], \sum_i u_i=1} \left[ \underbrace{J(Au)}_{\text{convex, closed}} + \underbrace{F(u)}_{\text{convex, closed}} \right],
\]

neither regularizer nor data term are differentiable.

Can be solved exactly and efficiently with the algorithm from Chapter 2.
## Runtime and memory requirements

<table>
<thead>
<tr>
<th># of Pixels $P = P_x \times P_y$</th>
<th># Labels $N_1 \times N_2$</th>
<th>Memory [Mb]</th>
<th>Runtime [s]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Previous</td>
<td>Proposed</td>
</tr>
<tr>
<td>320 $\times$ 240</td>
<td>8 $\times$ 8</td>
<td>112</td>
<td>28</td>
</tr>
<tr>
<td>320 $\times$ 240</td>
<td>16 $\times$ 16</td>
<td>450</td>
<td>56</td>
</tr>
<tr>
<td>320 $\times$ 240</td>
<td>32 $\times$ 32</td>
<td>1800</td>
<td>112</td>
</tr>
<tr>
<td>320 $\times$ 240</td>
<td>64 $\times$ 64</td>
<td>7200</td>
<td>225</td>
</tr>
<tr>
<td>640 $\times$ 480</td>
<td>8 $\times$ 8</td>
<td>448</td>
<td>112</td>
</tr>
<tr>
<td>640 $\times$ 480</td>
<td>16 $\times$ 16</td>
<td>1800</td>
<td>224</td>
</tr>
<tr>
<td>640 $\times$ 480</td>
<td>32 $\times$ 32</td>
<td>7200</td>
<td>448</td>
</tr>
<tr>
<td>640 $\times$ 480</td>
<td>64 $\times$ 64</td>
<td>28800</td>
<td>900</td>
</tr>
</tbody>
</table>
50 × 50 flow vectors, image resolution 640 × 480, within 5% of global optimum
Depth and Occlusion

\[ I_L : \Omega \rightarrow \mathbb{R}^3 \]
\[ I_R : \Omega \rightarrow \mathbb{R}^3 \]

estimate depth map and binary occlusion map - two dimensional label space!

\[
c_{\gamma}(x, y) = \begin{cases} 
  c_{occ} & \text{if } \gamma_2 = 1, \\
  \|I_L(x, y) - I_R(x - \gamma_1, y)\|_2 & \text{otherwise.}
\end{cases}
\]
Image color segmentation

Stronger smoothing in darker regions - possible with flexible label distance.
• **Vectorial total variation** extends the definition of TV from scalar-to vector-valued functions.

• A common and useful generalization is the **Frobenius-TV**. In the primal formulation, you integrate over the Frobenius norm of the derivative matrix (Jacobian).

• Frobenius-TV is closed and convex, so it can be minimized efficiently. Furthermore, it has a similar geometric property that the scalar TV with regards to jump functions.

• Vectorial TV can be used to construct functionals for **multilabel problems** with convex relaxations available.

• In the case of **product label spaces**, the memory and runtime requirements can be drastically reduced.

**See also: our talk on Thursday.**
Vectorial Total Variation


- Exhaustive introduction to variational methods and convex optimization in infinite dimensional spaces, as well as the theory of BV functions.
- Mathematically very advanced, requires solid knowledge of functional analysis.


- Classification and comparison of several extensions of TV to vector valued function.
- Evaluation of the cases with a dual formulation available.
VTV and Multilabel Problems


- Introduction of a certain convex relaxation for multilabel problems
- VTV to define regularizers with Euclidean representations for the label distance.


- Reduction technique for label spaces with product structure.
- Makes the algorithm feasible for very large problems like optic flow with thousands of labels.