

A Convex Clustering-based Regularizer for Image Segmentation

Benjamin Hell (TU Braunschweig), Marcus Magnor (TU Braunschweig)

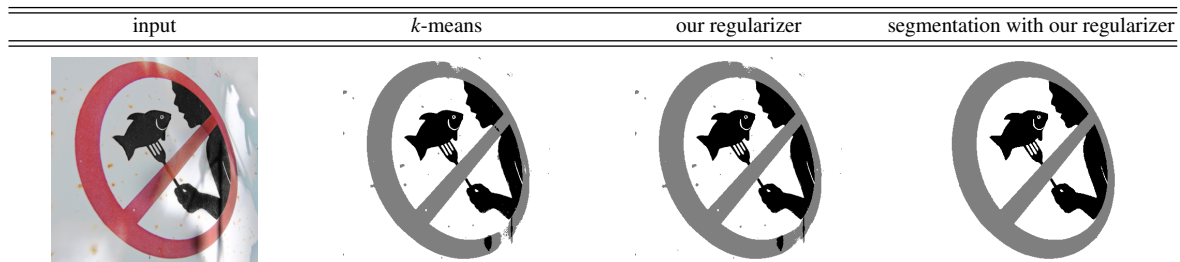


Figure 1: From left to right: Input image, k -means clustering result, result from our clustering-based regularizer and our clustering-based regularizer incorporated in a sophisticated image segmentation framework.

Abstract

In this paper we present a novel way of combining the process of k -means clustering with image segmentation by introducing a convex regularizer for segmentation-based optimization problems. Instead of separating the clustering process from the core image segmentation algorithm, this regularizer allows the direct incorporation of clustering information in many segmentation algorithms. Besides introducing the model of the regularizer, we present a numerical algorithm to efficiently solve the occurring optimization problem while maintaining complete compatibility with any other gradient descent based optimization method. As a side-product, this algorithm also introduces a new way to solve the rather elaborate relaxed k -means clustering problem, which has been established as a convex alternative to the non-convex k -means problem.

Categories and Subject Descriptors (according to ACM CCS): I.4.6 [IMAGE PROCESSING AND COMPUTER VISION]: Segmentation—Relaxation

1. Introduction

Clustering can be seen as some kind of segmentation process. Many popular clustering algorithms, like k -means clustering [XJ10] or mean-shift [FS10], have been successfully used for image segmentation tasks. More sophisticated and highly specialized image segmentation algorithms, however, rely on much more information of the image than just the distance of color information with respect to a certain metric like Euclidean distance. Nevertheless, this clustering information can be very useful for the segmentation process. It is therefore common to use clustering algorithms like k -means to obtain initial solutions for iterative numerical algorithms or pre-compute clusters, which serve as reference points throughout the whole segmentation process [HKB*15]. Unfortunately the k -means problem is non-convex, so the solution obtained by numerical algorithms quite frequently does not correspond to the global optimum of the k -means prob-

lem. To avoid this issue convex clustering problems like the relaxed k -means problem [ABC*14] or maximum likelihood models [LG07] have been investigated with great success.

Building upon the ideas of the convex relaxed k -means problem we introduce a novel convex regularizer, which follows a different philosophy than other approaches: Instead of separating the clustering process from the core segmentation procedure we combine both methods and fuse them into a single optimization problem. We therefore introduce a convex regularizer, which can be incorporated in many existing optimization pipelines. The optimization problem lying underneath this regularizer is essentially the relaxed k -means clustering problem, but mathematically modeled in a profoundly different way. As a side-product it also leads to a new method for solving the relaxed k -means clustering problem, which is a non-trivial task (cf. [ABC*14]).

A common practice to include clustering information in an optimization process is to pre-compute the clusters and use a deviation term from those pre-computed cluster centers. This keeps the cluster centers fixed throughout optimization, which might have a negative influence on the outcome of the optimization process, especially when a drastically different choice of cluster centers would lead to a lower overall objective function value, while barely influencing the clustering energy term value. Our approach implicitly allows adjustment of the cluster centers and hence does not have this disadvantage.

Paper structure: Section 2 gives an overview of the well known k -means clustering optimization problem and a corresponding (not so well known) relaxed counterpart, which has the property of being convex. Section 3 bridges the gap between discrete and continuous considerations for the theory of this paper. Continuing this line of thought Section 4 introduces a mathematical model for representing segmentations in a layer structure and introduces our novel clustering-based segmentation model afterwards. To solve the problem we propose a numeric gradient descent scheme, which is presented in Section 5. Section 6 shows results using our model and algorithm and Section 7 concludes the paper.

1.1. Contributions

This paper contains the following contributions:

- We show a way of combining the process of image segmentation with clustering algorithms in form of a novel k -means clustering-based segmentation problem, which has a magnitude of applications. For example, the presented approach can be used as a regularizer in several image segmentation problems.
- We introduce a continuous version of the relaxed k -means clustering problem.
- We introduce a numerical algorithm to solve our novel clustering-based regularization problem, which can be incorporated in many other optimization frameworks.
- The introduced clustering-based segmentation model and the corresponding optimization algorithm yield a new way of solving the relaxed k -means clustering problem.

1.2. Notation and Abbreviations

The following notation will be used in this paper.

\square	a box marks essential results and definitions
I	feature vector in \mathbb{R}^m (in image case: intensity)
$\langle \cdot, \cdot \rangle$	inner product on a Hilbert space
a	$(1/m, \dots, 1/m)^T \in \mathbb{R}^m$ (for conversion to grayscale)
$\ \cdot\ _p$	p -norm, $\ \cdot\ $ is 2-norm
R_i	i -th segmentation region or cluster
$ R_i $	volume of region R_i
d	distance function
$\Omega \subset \mathbb{R}^n$	basic domain
v	segmentation variable representing layer structure
V	convex restriction set for the segmentation variable v
k	number of segments or clusters

2. Discrete Problems

In this section we discuss discrete clustering algorithms that serve as an inspiration for this work, specifically the well known discrete k -means clustering problem as well as its relaxed form.

2.1. The Discrete k -Means Clustering Problem

One of the most popular clustering algorithms is k -means clustering [KMN*02]. For a given set N of datapoints $\{x_i \in \mathbb{R}^m | i = 1, \dots, N\}$, the k -means clustering tries to find k centers c_i ($i = 1, \dots, k$), such that the sum over corresponding distances $d(x_i, c_j)$, where each x_i ($i = 1, \dots, N$) is assigned to one specific c_j ($j = 1, \dots, k$), is minimal. So k -means clustering is the process of solving the following minimization problem:

$$\operatorname{argmin}_{\{c_j \in \mathbb{R}^m | j=1, \dots, k\}} \sum_{i=1}^N \min_{j \in \{1, \dots, k\}} d(x_i, c_j) \quad (1)$$

Note that d can be any sort of distance measure, but for classic k -means clustering d corresponds to squared Euclidean distance. Due to the min function occurring in (1) the minimization problem is non-convex. Indeed most k -means clustering algorithms suffer from the fact, that they do not necessarily converge to a global optimum, which might lead to undesired results (cf. Section 6.1.2). For this and many other reasons, it is therefore of high interest to find convex relaxations of Problem (1), which should yield approximate solutions to the initial problem while maintaining convexity of the objective function.

2.2. Relaxed Discrete k -Means Clustering Problem

As it turns out Problem (1) can be relaxed into a convex problem, by introducing an indicator function $z(i, j)$ ($i, j = 1, \dots, N$) which is intended to be greater than zero if the points x_i and x_j belong to the same cluster and zero otherwise. Imposing further constraints on the indicator function $z(\cdot, \cdot)$ we obtain the following linear optimization problem:

$$\begin{aligned} \operatorname{argmin}_z \quad & \sum_{i,j=1}^N d(x_i, x_j) z(i, j) \\ \text{s.t.} \quad & \sum_{j=1}^N z(i, j) = 1 \quad \forall i \in \{1, \dots, N\} \\ & \sum_{i=1}^N z(i, i) = k \quad (\text{k: number of clusters}) \\ & z(i, j) \leq z(i, i) \quad \forall i, j \in \{1, \dots, N\} \\ & z(i, j) \in [0, 1] \quad \forall i, j \in \{1, \dots, N\} \end{aligned} \quad (2)$$

For further details see [ABC*14]. The structure of Problem (2) will become more evident in Section 3, when we consider its continuous counterpart and an implicit representation of the optimal solution to this problem.

3. Towards a Continuous Structure

Instead of clustering discrete points we want to shift our focus onto continuous problems. So far we have only considered discrete versions of clustering problems. Our ultimate goal is to use the relaxed k -means problem in a continuous setting to obtain a novel regularizer that works well in a continuous image segmentation setup. We consider a domain $\Omega \subset \mathbb{R}^n$, which will later on just be a rectangle representing an image (case $n = 2$). Nevertheless, all future considerations work for arbitrary space dimension n . Each point $x \in \Omega$ is assigned a corresponding feature vector $I(x) \in \mathbb{R}^m$. If this function $I : \Omega \rightarrow \mathbb{R}^m$ represents a color image, each point x is assigned a color vector, i.e. $I : \mathbb{R}^2 \supset \Omega \rightarrow \mathbb{R}^3$.

3.1. Continuous Extension of the Relaxed k -Means Problem

We extend Problem (2) to the continuous case, using the notation mentioned at the beginning of this section, and obtain

$$\begin{aligned} \operatorname{argmin}_z \int_{\Omega} \int_{\Omega} d(I(x), I(y)) z(x, y) dx dy \\ \text{s.t.} \int_{\Omega} z(x, y) dx = 1 \quad \forall x \in \Omega \subset \mathbb{R}^n \\ \int_{\Omega} z(x, x) dx = k \quad (k: \text{number of clusters}) \\ z(x, y) \leq z(x, x) \quad \forall x, y \in \Omega \\ z(x, y) \in [0, 1] \quad \forall x, y \in \Omega \end{aligned} \quad (3)$$

It can be easily verified, that the optimal solution $\hat{z}(x, y)$ to this problem is

$$\hat{z}(x, y) = \begin{cases} \frac{1}{|R_i|} & x, y \in R_i \subset \Omega \quad (i = 1, \dots, k) \\ 0 & \end{cases} \quad (4)$$

with R_i being the i -th cluster and $|R_i|$ its corresponding n -dimensional volume. In the case of $\Omega \subset \mathbb{R}^2$ representing an image, $|R_i|$ corresponds to the image area covered by cluster i ($i = 1, \dots, k$). Note that representation (4) does not give an explicit solution to the problem because it relies on knowledge about the clusters R_i ($i = 1, \dots, k$).

4. From Clustering to Segmentation

Clustering can be considered as some kind of segmentation, where each cluster of datapoints represents one segment. To include the clustering process in a continuous image segmentation framework additional work has to be done. So far we have considered discrete clustering problems in Section 2. As we want to maintain convexity in all our considerations, the relaxed Problem (2) seems to be a good starting point. In Section 3 we extended the relaxed discrete Problem (2) to a continuous domain $\Omega \subset \mathbb{R}^n$. Unfortunately, the

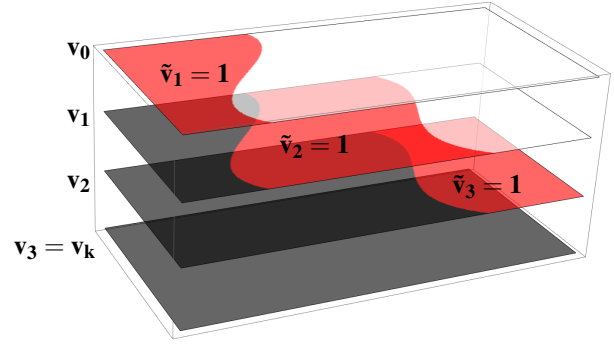


Figure 2: Layer Structure Example: This figure shows an example of 4 layers $v_0, \dots, v_3 \in \tilde{V}$. White color corresponds to the value 1, black to the value 0. The red regions are the individual segments R_i ($i = 1, \dots, 3$) given by the layer structure and are characterized by $\tilde{v}_i = v_{i-1} - v_i = 1$.

representation of clusters via the function $z(x, y)$ in the continuous relaxed k -means Problem (3) is not well suited for a segmentation environment. The reason for this is its implicit representation of individual clusters. So, the final and most elaborate step we need to take is to find a mathematical foundation that is compatible with image segmentation and is based on an explicit representation of the individual regions R_i involved in the segmentation process.

4.1. Layer Structure

To represent multiple regions mathematically we use a concatenation of binary functions, each one splitting the image area into one more segment. This is a convenient way for handling segmentation information and has a variety of benefits. The representation follows the ideas presented in [CPKC11] and [HKB*15], which are based on [PCBC09]. The set of binary variables is defined by:

$$\begin{aligned} \tilde{V} := \{v = (v_0, \dots, v_k) : \Omega \rightarrow \{0, 1\}^{k+1} \mid \\ 1 \geq v_1(x) \geq \dots \geq v_{k-1}(x) \geq 0, x \in \Omega\} \end{aligned} \quad (5)$$

with $v_0(\cdot) \equiv 1$ and $v_k(\cdot) \equiv 0$. Each region R_i is being represented by the difference $\tilde{v}_i := v_{i-1} - v_i$ ($i = 1, \dots, k$):

$$R_i = \{x \in \Omega : \underbrace{v_{i-1}(x) - v_i(x)}_{\tilde{v}_i(x)} = 1\} \quad (i = 1, \dots, k) \quad (6)$$

Figure 2 shows an example of a specific segmentation into different regions via some variable $v \in \tilde{V}$.

As we are interested in a convex model we have to allow the variables v_i ($i = 1, \dots, k$) to take values in the range $[0, 1]$ instead of just assuming binary values. This yields the following convex restriction set:

$$\begin{aligned} V := \{v = (v_0, \dots, v_k) : \Omega \rightarrow [0, 1]^{k+1} \mid \\ 1 \geq v_1(x) \geq \dots \geq v_{k-1}(x) \geq 0, x \in \Omega\} \end{aligned} \quad (7)$$

Note that due to the definition of V we have

$$\sum_{i=1}^k \tilde{v}_i = \sum_{i=1}^k v_{i-1} - v_i = v_0 - v_k = 1$$

4.2. The Novel Clustering-based Regularizer

Combining the layer structure v_i ($i = 1, \dots, k$) and the basic ideas behind the relaxed k -means Problem (3), we obtain the following optimization problem

Clustering-based Segmentation Problem	
$\operatorname{argmin}_{v \in V} \int_{\Omega} \int_{\Omega} d(I(x), I(y)) w(x) \overbrace{\langle \tilde{v}(x), \tilde{v}(y) \rangle}^{= \sum_{i=1}^k \tilde{v}_i(x) \cdot \tilde{v}_i(y)} dx dy \quad (8)$	
$\text{s.t. } \int_{\Omega} w(x) \tilde{v}_i(x) dx = 1 \quad (i = 1, \dots, k) \quad (\text{c1})$	
$\int_{\Omega} w(x) \langle a, I(x) \rangle \tilde{v}_i(x) dx \leq \int_{\Omega} w(x) \langle a, I(x) \rangle \tilde{v}_{i+1}(x) dx \quad (\text{c2})$	
$w(x) \in [0, 1] \quad \forall x \in \Omega \quad (\text{c3})$	

with $a := (1/m, \dots, 1/m)^T \in \mathbb{R}^m$ such that $\langle a, I(x) \rangle$ is a "grayscale" value. As mentioned before, d can be any kind of distance measure. For our purpose we only consider functions of the form $d(I(x), I(y)) = \psi(\|I(x) - I(y)\|)$, with ψ being strictly monotonically increasing on \mathbb{R}^+ and hence $d(\cdot, \cdot)$ being minimal for $I(x) = I(y)$ (which is a very important property for our model). Reasonable choices for d are squared Euclidean distance or negative Parzen density (cf. Section 6).

If we compare this problem with Problem (3), we see that $z(x, y)$ corresponds to $w(x) \langle \tilde{v}(x), \tilde{v}(y) \rangle$ and that the constraints have taken a different form. The constraints (c1) and (c3) are motivated by Theorem 1. The only reason for the existence of (c2) is that the layer structure introduced in Section 4.1 needs a specific order of \tilde{v}_i ($i = 1, \dots, k$) to guarantee a unique optimum \hat{v} to Problem (8). Without (c2), regions R_i and R_j ($i, j \in \{1, \dots, k\}$) could be interchanged without influencing the objective function value and the result would still fulfill all the other constraints.

In general, there are many ways to model a clustering-based segmentation problem with similar properties to Problem (8), most of all by changing the definition of $z(x, y)$. Our particular modeling of $z(x, y)$ has been inspired by the following considerations:

At the optimal point of Problem (8) we want to achieve

$$\tilde{v}_i(x) = \begin{cases} 1 & x \in R_i \\ 0 & \text{otherwise} \end{cases} \quad (i = 1, \dots, k)$$

which leads to

$$\langle \tilde{v}(x), \tilde{v}(y) \rangle = \begin{cases} 1 & x, y \in R_i \quad (i = 1, \dots, k) \\ 0 & \text{otherwise} \end{cases}$$

As we do know the general form of the optimal solution to the relaxed k -means problem (cf. (4)), we introduce an additional weighting function w , which is intended to take the value $w(x) = 1/|R_i|$ for $x \in R_i$ at the optimal solution $\hat{z}(x, y)$.

In addition to that, our definition has the following important property, which states that Problem (8) is in accordance with the relaxed k -means Problem (3).

Theorem 1. *Let $z(x, y) := w(x) \langle \tilde{v}(x), \tilde{v}(y) \rangle$. Then each optimal solution (\hat{v}, \hat{w}) to Problem (8) fulfills*

$$\begin{aligned} \int_{\Omega} z(x, x) dx &= \int_{\Omega} w(x) dx = k \\ z(x, y) &\leq z(x, x) \quad \forall x, y \in \Omega \\ z(x, y) &\in [0, 1] \quad \forall x, y \in \Omega \end{aligned} \quad (9)$$

Proof. Due to $v \in V$ we have $\sum_{i=1}^k \tilde{v}_i = 1$. So the first assertion follows from

$$\int_{\Omega} z(x, x) dx = \int_{\Omega} w(x) dx \stackrel{(\text{c1})}{=} \int_{\Omega} w(x) \sum_{i=1}^k \tilde{v}_i dx = k \quad (10)$$

The second one is fulfilled if $\langle \tilde{v}(x), \tilde{v}(y) \rangle \leq \|\tilde{v}(x)\|^2$. This is in general not the case but it can be shown that each optimal solution \hat{v} will exhibit the property of $\tilde{v}_i \in \{0, 1\}$ ($i = 1, \dots, k$), which in return means that the inequality is fulfilled for optimal solutions, which is all we need.

The third assertion follows from

$$z(x, y) = \overbrace{w(x)}^{\geq 0} \overbrace{\langle \tilde{v}(x), \tilde{v}(y) \rangle}^{\geq 0} \geq 0 \quad (11)$$

$$\begin{aligned} z(x, y) &= w(x) \langle \tilde{v}(x), \tilde{v}(y) \rangle \stackrel{(\text{c3})}{\leq} \langle \tilde{v}(x), \tilde{v}(y) \rangle \leq \|\tilde{v}(x)\| \cdot \|\tilde{v}(y)\| \\ &\leq \|\tilde{v}(x)\|_1 \cdot \|\tilde{v}(y)\|_1 = 1 \end{aligned}$$

where the fact that $0 \leq \tilde{v}_i(x) \leq 1$ ($i = 1, \dots, k$) due to $v \in V$ plays a major role. \square

Essentially Theorem 1 expresses that each optimal solution to our clustering-based regularization Problem (8) (which of course fulfills constraints (c1), (c2), (c3)) automatically fulfills all the constraints imposed by the continuous relaxed k -means clustering Problem (3).

Note that Problem (8) is not convex in the variable v , but it is convex in z . Our algorithm for solving the problem, which is presented in Section 5, is essentially looking for an optimal solution \hat{z} by solving the problem in v and w .

5. Solving the Problem

To solve Problem (8), representing the model for our novel regularizer, we propose a projected gradient descent method, which alternates between taking a gradient descent step in the primal variable v , projecting the result on the restriction

set V and adjusting the weighting function w in order to fulfill the constraints.

5.1. Finding an Initial Solution

The first step is to find an initial solution $v^0 \in V$. This can be done in a various number of ways, as long as the initial solution lies in the set V and fulfills the ordering constraint (c2). We propose the following simple algorithm (Algorithm 1), which builds its clusters based on the ordering of grayscale values.

Algorithm 1 Computing the initial solution v^0

- 1: **procedure** COMPUTEINITIALSOLUTION(*image*)
 - 2: choose \tilde{c}_i ($i = 1, \dots, k$) with $0 \leq \tilde{c}_1 \leq \dots \leq \tilde{c}_k \leq 1$
 - 3: set region $R_i := \{x \in \Omega : |\langle a, I(x) \rangle - \tilde{c}_i| \leq |\langle a, I(x) \rangle - \tilde{c}_j| \text{ (} j = 1, \dots, k \text{)}\}$ ($i = 1, \dots, k$)
 ▷ in order to fulfill (c2), cf. Section 4.2
 - 4: set $\tilde{v}_i(x) := \begin{cases} 1 & x \in R_i \\ 0 & \text{otherwise} \end{cases}$ ($i = 1, \dots, k$)
 - 5: set $v_i := 1 - \sum_{j=1}^i \tilde{v}_j$ ($i = 0, \dots, k$)
 ▷ so $v \in V$ according to def. (7)
 - 6: **return** $v = (v_0, \dots, v_k)$
 - 7: **end procedure**
-

5.2. Computing the Gradient

Let us first define $f(v, w)$ as the objective function of Problem (8), i.e.

$$f(v, w) := \int_{\Omega} \int_{\Omega} d(I(x), I(y)) w(x) \langle \tilde{v}(x), \tilde{v}(y) \rangle dx dy \quad (12)$$

The gradient vector can be obtained by differentiating with respect to v_i ($i = 0, \dots, k$). Due to the structure of the objective function $f(v, w)$ we only need to know the following differential

$$\frac{d}{dv_i} \langle \tilde{v}(x), \tilde{v}(y) \rangle \quad (13)$$

which can be obtained by taking the directional derivative in the direction $h^i := (0, \dots, 0, h_i, 0, \dots, 0)^T$ ($i = 1, \dots, k$). This directional derivative follows from the fact that only \tilde{v}_i and \tilde{v}_{i+1} depend on v_i , hence

$$\begin{aligned} & \frac{d}{dv_i} \langle \tilde{v}(x), \tilde{v}(y) \rangle (h^i) = \\ & \frac{d}{dt} \left\langle \left(\begin{array}{c} (v_{i-1} - (v_i + th_i))(x) \\ ((v_i + th_i) - v_{i+1})(x) \end{array} \right), \left(\begin{array}{c} (v_{i-1} - (v_i + th_i))(y) \\ ((v_i + th_i) - v_{i+1})(y) \end{array} \right) \right\rangle \Bigg|_{t=0} = \\ & \left\langle \frac{d}{dt} \left(\begin{array}{c} v_{i-1}(x) - (v_i + th_i)(x) \\ (v_i + th_i)(x) - v_{i+1}(x) \end{array} \right), \left(\begin{array}{c} \tilde{v}_i(y) \\ \tilde{v}_{i+1}(y) \end{array} \right) \right\rangle \Bigg|_{t=0} + \\ & \left\langle \left(\begin{array}{c} \tilde{v}_i(x) \\ \tilde{v}_{i+1}(x) \end{array} \right), \frac{d}{dt} \left(\begin{array}{c} v_{i-1}(y) - (v_i + th_i)(y) \\ (v_i + th_i)(y) - v_{i+1}(y) \end{array} \right) \right\rangle \Bigg|_{t=0} = \\ & (\tilde{v}_{i+1}(y) - \tilde{v}_i(y))h_i(x) + (\tilde{v}_{i+1}(x) - \tilde{v}_i(x))h_i(y) \end{aligned} \quad (14)$$

We then obtain the gradient of the objective function f with respect to v via considering

$$\begin{aligned} & \frac{d}{dt} f(v + th^i, w) \Big|_{t=0} \stackrel{(14)}{=} \\ & \int_{\Omega} \int_{\Omega} d(I(x), I(y)) w(x) (\tilde{v}_{i+1}(y) - \tilde{v}_i(y))h_i(x) + \\ & \int_{\Omega} \int_{\Omega} d(I(x), I(y)) w(x) (\tilde{v}_{i+1}(x) - \tilde{v}_i(x))h_i(y) dx dy = \\ & \int_{\Omega} \int_{\Omega} d(I(x), I(y)) (w(x) + w(y)) (\tilde{v}_{i+1}(y) - \tilde{v}_i(y)) dy h_i(x) dx \end{aligned} \quad (15)$$

where the last equality follows from Fubini's theorem and the symmetry of the metric $d(\cdot, \cdot)$.

So overall we have

$$\begin{aligned} & \frac{d}{dv_i} f(v, w) = \\ & \int_{\Omega} d(I(x), I(y)) (w(x) + w(y)) (\tilde{v}_{i+1}(y) - \tilde{v}_i(y)) dy \end{aligned} \quad (16)$$

with $D_v f(v, w)(h) = \langle D_v f(v, w), h \rangle$, where $\langle \cdot, \cdot \rangle$ denotes the scalar product in $L_2(\Omega)$ and $D_v f(v, w) = \left(\frac{d}{dv_0} f(v, w), \dots, \frac{d}{dv_k} f(v, w) \right)$. So computing the gradient of the objective function f is done via Algorithm 2.

Algorithm 2 Computing the gradient of the objective function f

- 1: **procedure** COMPUTEGRADIENT(v, w)
 - 2: set $\frac{d}{dv_0} f(v, w) = \frac{d}{dv_k} f(v, w) = 0$ ▷ by def. of set V
 - 3: **for** $i = 1, \dots, k - 1$ **do**
 - 4: set $\frac{d}{dv_i} f(v, w)$ according to (16) for all $x \in \Omega$
 - 5: **end for**
 - 6: **return** $D_v f(v, w) = \left(\frac{d}{dv_0} f(v, w), \dots, \frac{d}{dv_k} f(v, w) \right)$
 - 7: **end procedure**
-

5.3. Projection Scheme

One of the major constraints in Problem (8) that might easily be overlooked is that $v \in V$, with the restriction set V being defined in (7). Given a function v , projecting onto V is the same as projecting each vector $v(x)$ ($x \in \Omega$) onto the set

$$V(x) := \left\{ y = (1, y_1, \dots, y_{k-1}, 0) \in \mathbb{R}^{k+1} \mid \begin{array}{l} 1 \geq y_1 \geq \dots \geq y_{k-1} \geq 0 \end{array} \right\} \quad (17)$$

Obtaining the projection $\tilde{y} = (1, \tilde{y}_1, \dots, \tilde{y}_{k-1}, 0)$ of a vector $y = (1, y_1, \dots, y_{k-1}, 0) \in \mathbb{R}^{k+1}$ onto $V(x)$ is no trivial task. An efficient algorithm for doing that has been introduced in [CCP12, App. B] and is represented by Algorithm 3.

Algorithm 3 Projection on restriction set V

```

1: procedure PROJECTONTOV( $y$ )
2:   set  $\tilde{y}_i := y_i$  and define  $k$  clusters  $Z_i := \{i\}$ 
   ( $i = 1, \dots, k-1$ )
3:   identify index  $i \in \{1, \dots, k-2\}$  with  $\tilde{y}_i < \tilde{y}_{i+1}$ , if
   there is no such index jump to 5
4:   set clusters  $Z_j := Z_i \cup Z_{i+1} \forall j \in Z_i \cup Z_{i+1}$ , set
    $\tilde{y}_{i+1} := \tilde{y}_i := \sum_{j \in Z_i} (\tilde{y}_j) / |Z_i|$  and go back to 3
5:   set  $\tilde{y}_i := \max(\min(\tilde{y}_i, 1), 0)$   $\triangleright$  clipping to  $[0, 1]$ 
6:   return  $\tilde{y}$ 
7: end procedure

```

5.4. Update Scheme for the Weighting Function

The weighting function w has to be chosen such that the constraints of Problem (8) are fulfilled. So the challenge is to find w , which at least approximately satisfies the restrictions (c1), (c2), (c3) of Problem (8) given a vector $v = (v_0, \dots, v_k) \in V$. As we do know how the optimal solution $\hat{z}(x, y) = \hat{w}(x) \langle \hat{v}(x), \hat{v}(y) \rangle$ looks like we intend to set $w(x) := 1/|R_i|$ for $x \in R_i$ ($i = 1, \dots, k$), where R_i is given by the current state of v . This corresponds to setting:

$$w(x) := \sum_{i=1}^k \frac{\tilde{v}_i(x)}{\max\left(\int_{\Omega} \tilde{v}_i(y) dy, \varepsilon\right)} \quad (18)$$

$\approx |R_i|$

for some small value $\varepsilon > 0$ and the max expression only compensating for the case of $\int_{\Omega} \tilde{v}_i(y) dy = 0$, which corresponds to $\tilde{v}_i \equiv 0$. This formulation fulfills the constraint (c1) in the case of binary $\tilde{v}_i \in \{0, 1\}$ ($i = 1, \dots, k$) and approximates the equality in the general case of $\tilde{v}_i \in [0, 1]$ ($i = 1, \dots, k$). Besides approximating (c1) very well, w defined by (18) has the property

$$\int_{\Omega} w(x) dx = \sum_{i=1}^k \frac{\int_{\Omega} \tilde{v}_i(x) dx}{\int_{\Omega} \tilde{v}_i(y) dy} = k \quad (19)$$

which is another indicator of (18) being a reasonable choice for updating w . So we arrive at Algorithm 4.

Algorithm 4 Updating weighting function w

```

1: procedure UPDATEWEIGHTS( $v$ )
2:   set  $w(x)$  according to (18) for all  $x \in \Omega$ 
3:   return  $w$ 
4: end procedure

```

Note that another reasonable choice for updating w would be to use a gradient descent scheme for solving a least squares problem for enforcing (c1). In practice this turned out to work fine, but has no benefits compared to the simpler approach represented by Algorithm 4.

5.5. Overall Numeric Algorithm

Putting all the previously mentioned steps together we arrive at the final version of our iterative algorithm:

- Taking an initial solution v^0 for the optimization variable v , which corresponds to the layer structure representing the current segmentation. This can be done according to Section 5.1.
- Updating the weighting function w according to Section 5.4.
- Taking the gradient from Section 5.2 to update the optimization variable v .
- Projecting the obtained solution on the restriction set V according to Section 5.3.

Algorithm 5 Solving the Clustering-based Segmentation Problem

```

1:  $v^0 = \text{COMPUTEINITIALSOLUTION}(\text{image}) \triangleright$  fulfill (c2)
2: procedure CBSEGMENTATION( $\text{image}, v^0$ )
    $\triangleright$  image: feature vectors  $I(x)$ ,  $v^0$ : initial segmentation
3:   while  $\|v^{j+1} - v^j\| > \varepsilon$  do
4:      $w = \text{UPDATEWEIGHTS}(v^j)$   $\triangleright$  Section 5.4
5:      $v^{j+1} = v^j - \tau \cdot \text{COMPUTEGRADIEN}(v, w)$ 
        $\triangleright$  gradient descent step, Section 5.2
6:      $v^{j+1} = \text{PROJECTONTOV}(v^{j+1})$ 
        $\triangleright$  projection on restriction set  $V$ , Section 5.3
7:      $v^j = v^{j+1}$ 
8:   end while
9:   return  $v^j$ 
10: end procedure

```

Note that this algorithm can be combined with any optimization framework that uses some sort of gradient descent method and a layer structure similar to the one presented in Section 4.1.

6. Results

In this section we evaluate our regularizer in various test scenarios. We first show that our optimization approach from Section 5 works as a standalone algorithm, which means that it successfully solves the relaxed k -means Problem (3) and Problem (8). We then show a comparison to standard clustering algorithms and finally show results of our approach incorporated in an elaborate image segmentation framework based on [HKB*15].

We present all segmentation (clustering) results as grayscale images, each gray-level corresponding to one distinct segment.

In Section 4 we briefly talked about possible choices for the distance function d . If not stated explicitly otherwise, we choose negative Parzen density (inspired by mean-shift like clustering as in [FS10]), i.e.

$$d(I(x), I(y)) := -\frac{1}{2\pi\alpha^2} e^{-\frac{(\|I(x) - I(y)\|^2 / (2\alpha^2))}{2}} \quad (20)$$

where we set $\alpha = 1/10$. This rather extreme choice yields faster convergence and more plausible results than using squared Euclidean distance.

6.1. Evaluating the Regularizer

We apply our Algorithm 5 to different input images in order to compute clusters with respect to their RGB color information. All the results shown in Figure 3(c,d) are solutions to our clustering-based segmentation Problem (8). It can be observed, that our algorithm yields a convenient cluster structure with results comparable to k -means clustering.

6.1.1. Choosing Different Distance Functions

We compare the results of our algorithm using different distance functions $d(\cdot, \cdot)$: Negative Parzen density (20) and squared Euclidean distance. Both distance functions fulfill our basic requirement stated in Section 4.2, which is that $d(I(x), I(y)) = \psi(\|I(x) - I(y)\|)$, with some function ψ being strictly monotonically increasing on \mathbb{R}^+ . The results in Figure 3(c,d) show that the distance function has some influence on the clustering process. While computing about hundred test cases we discovered that most of the time the final clustering result differs only slightly for different distance functions, but the rate of convergence of our iterative Algorithm 5 might be influenced considerably.

6.1.2. Comparison to k-Means Clustering

Our clustering-based segmentation Algorithm 5 solves a convex problem, so we are guaranteed a globally optimal solution. Figure 3(f,g) shows a side by side comparison of our results with results from traditional k -means clustering computed in Mathematica [Res15]. As mentioned previously, k -means clustering algorithms might not yield a globally optimal solution. To investigate this behavior we ran Mathematica's k -means implementation 200 times for each image, each time using a different random initial solution. For each of those images the algorithm found a globally optimal solution in approximately 50% of all test cases. The rest of the results belonged to undesirable local optima, which deviated strongly from the desired global optima (Figure 3(f)).

6.2. Incorporation into Segmentation Framework

The intention of this section is to briefly show how our regularizer works in a sophisticated image segmentation environment. We choose the framework from [HKB*15], which is built upon solving the following optimization problem:

$$\begin{aligned} \operatorname{argmin}_{v \in V} & \underbrace{\int_{\Omega \setminus \mathcal{B}} \|\nabla I(x)\| dx}_{\text{intensity variation on segments}} + \underbrace{\lambda_1 \int_{\mathcal{B}} \left\| \frac{\nabla I(x)}{\|\nabla I(x)\|} - v_{\mathcal{B}}(x) \right\|^2 d\mathcal{H}^{n-1}}_{\text{directional deviation}} + \\ & \underbrace{\lambda_2 \int_{\mathcal{B}} \frac{1}{1 + \alpha \|\nabla I(x)\|} d\mathcal{H}^{n-1}}_{\text{weighted boundary length}} + \underbrace{\lambda_3 \int_{\Omega} \sum_{i=1}^k \tilde{v}_i(x) \|I(x) - \bar{I}_i\| dx}_{\text{image data}} \end{aligned} \quad (21)$$

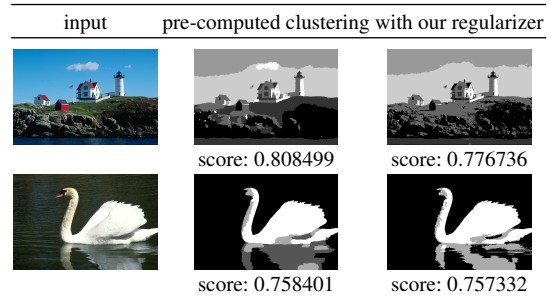


Figure 4: Evaluation on BSDS500 Images: The input image from the BSDS500 dataset [AMFM11] (left), the output of the image segmentation framework presented in [HKB*15] with pre-computed k -means clustering (middle) and the output using the same approach with incorporated regularizer (right). Below the output images we denoted the corresponding segmentation covering scores.

with \mathcal{B} being the boundary of the image segments, $v_{\mathcal{B}}$ the corresponding inner normal vector, \mathcal{H}^{n-1} being the $(n-1)$ -dimensional Hausdorff measure and I_i ($i = 1, \dots, k$) corresponding to mean color values on the different image segments R_i .

In order to incorporate our regularizer into this problem, we replace the "image data" term with our clustering-based segmentation Problem (8). The corresponding numerical algorithm then has to be adjusted in the gradient descent step according to Algorithm 5, but this is all that needs to be done.

Figure 3(e) shows segmentation results using this approach. It can clearly be observed that the additional information of weighted boundary length (of the individual segments) and boundary normal deviation add considerably to the quality of the segmentation.

To further investigate the output when using the proposed incorporated regularizer scheme, we apply the algorithm to a few images of the BSDS500 dataset, which are also presented in [HKB*15], and compute their segmentation covering score [AMFM11]. The results can be seen in Figure 4. Compared to the approach of solving Problem (21) with pre-computed k -means clustering (for computing mean color values \bar{I}_i in the image data term), we noticed only slight differences in the final output and segmentation covering scores. Subjectively we find the results computed with the incorporated regularizer slightly more appealing and we noticed a faster rate of convergence when using the proposed regularizer.

7. Conclusion

In this paper we have presented a general framework for incorporating k -means like clustering information into an image segmentation process, while maintaining convexity of the problem. We introduced a novel model for a clustering-based regularizer and a corresponding numerical algorithm to solve the optimization problem lying underneath the whole process. This algorithm is flexible enough to work

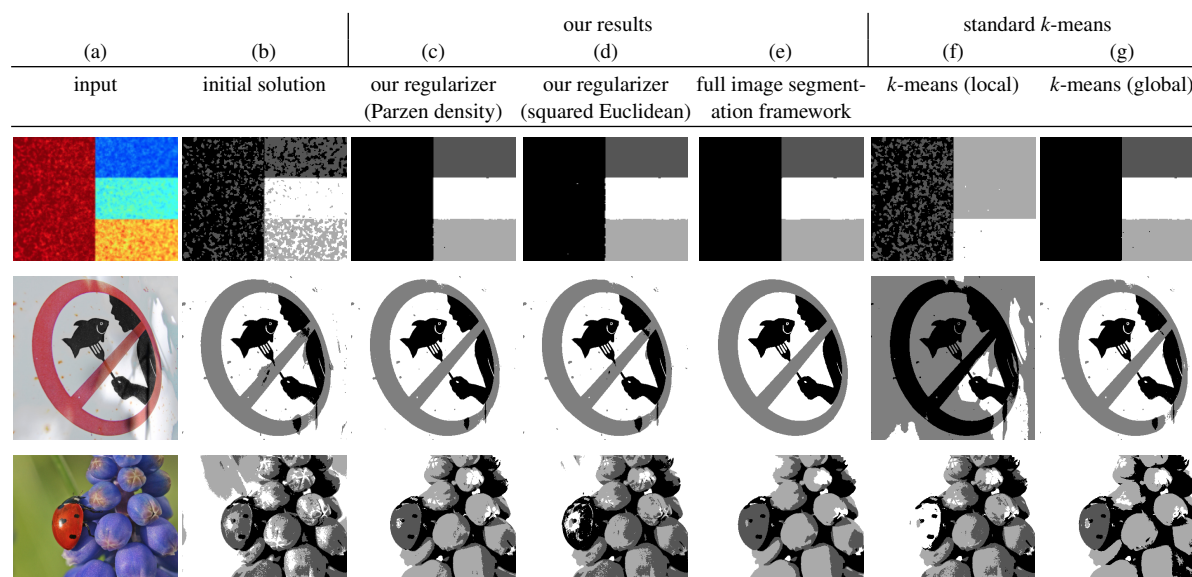


Figure 3: Side by Side Comparison of the Results: From left to right: This figure shows the input image (a), the initial solution based on closest grayscale distance (cf. Section 5.1) (b), the output of our clustering-based segmentation Algorithm 5 using negative Parzen density (20) (c), our clustering-based segmentation output with squared Euclidean distance (d), the results using a full-fledged image segmentation framework [HKB*15] (e), local optima of standard k -means clustering (f) and global optima of standard k -means clustering (g). The first picture, created by Farbman et. al. [FFLS08], is separated into 4 segments, the second one into 3 segments and the third one into 4 segments.

with any direction of descent based segmentation framework. Our results show the applicability of our approach in various scenarios including a sophisticated image segmentation process. In addition to that, we have shown that the proposed method delivers plausible clustering results on its own and might be used as a standalone clustering algorithm, which guarantees convergence towards its global optimum. Future work includes further investigation of the incorporation into an image segmentation framework and a fast implementation of the algorithm for GPU computing (e.g. using CUDA). The presented algorithm is well suited for parallel computing and may be accelerated by exploiting certain symmetry properties in the gradient computation step.

Acknowledgment

The research leading to these results has received funding from the European Unions Seventh Framework Programme FP7/2007-2013 under grant agreement no. 256941, Reality CG.

References

- [ABC*14] AWASTHI P., BANDEIRA A. S., CHARIKAR M., KRISHNASWAMY R., VILLAR S., WARD R.: Relax, no need to round: integrality of clustering formulations, Aug. 2014. 1, 2
- [AMFM11] ARBELAEZ P., MAIRE M., FOWLKES C., MALIK J.: Contour detection and hierarchical image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* 33, 5 (May 2011), 898–916. 7
- [CCP12] CHAMBOLLE A., CREMERS D., POCK T.: A convex approach to minimal partitions. 1113–1158. 5

- [CPKC11] CREMERS D., POCK T., KOLEV K., CHAMBOLLE A.: Convex relaxation techniques for segmentation, stereo and multiview reconstruction. In *Markov Random Fields for Vision and Image Processing*. MIT Press, 2011. 3
- [FFLS08] FARBMAN Z., FATTAL R., LISCHINSKI D., SZELISKI R.: Edge-preserving decompositions for multi-scale tone and detail manipulation. In *ACM SIGGRAPH 2008 Papers* (New York, NY, USA, 2008), SIGGRAPH '08, ACM, pp. 67:1–67:10. 8
- [FS10] FULKERSON B., SOATTO S.: Really quick shift: Image segmentation on a GPU. In *Trends and Topics in Computer Vision - ECCV 2010 Workshops, Heraklion, Crete, Greece, September 10-11, 2010, Revised Selected Papers, Part II* (2010), pp. 350–358. 1, 6
- [HKB*15] HELL B., KASSUBECK M., BAUSZAT P., EISEMANN M., MAGNOR M.: An approach toward fast gradient-based image segmentation. *Image Processing, IEEE Transactions on* 24, 9 (Sept 2015), 2633–2645. 1, 3, 6, 7, 8
- [KMN*02] KANUNGO T., MOUNT D. M., NETANYAHU N. S., PIATKO C. D., SILVERMAN R., WU A. Y.: An efficient k -means clustering algorithm: Analysis and implementation. *IEEE Trans. Pattern Anal. Mach. Intell.* 24, 7 (July 2002), 881–892. 2
- [LG07] LASHKARI D., GOLLAND P.: Convex clustering with exemplar-based models. In *Advances in Neural Information Processing Systems 20, Proceedings of the Twenty-First Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 3-6, 2007* (2007), pp. 825–832. 1
- [PCBC09] POCK T., CHAMBOLLE A., BISCHOF H., CREMERS D.: A convex relaxation approach for computing minimal partitions. 3
- [Res15] RESEARCH W.: Mathematica 10.1, 2015. 7
- [XJ10] XIE J., JIANG S.: A simple and fast algorithm for global k -means clustering. *Education Technology and Computer Science, International Workshop on* 2 (2010), 36–40. 1