Analysis of Convex Relaxation Approaches for Minimal Partitions

Additional Project for "Convex Optimization", ws 2021

Jiahui Li, Lingyi Zhou

June 23, 2022

Abstract

Image segmentation is a fundamental task in image processing referring to the process of subdividing a digital image into multiple image sub-regions. As a classic algorithm, Chambolle-Pock algorithm (also known as primal-dual algorithm) is considered as an efficient method for image segmentation. In this project, we try to implement the Chambolle-Pock algorithm and compare it with Zach's algorithm which uses a simpler relaxation. We apply these algorithms to reconstruction of a triple/quad point, segmentation of a scenery image and segmentation of a portrait. Experiments show that primal-dual algorithm outperforms Zach's algorithm, and achieves competitive results. Finally, we discuss the convergence rate of Chambolle-Pock algorithm and preconditioning technique of accelerating the original primal-dual algorithm.

1 Introduction

Image segmentation partitions a digital image into multiple image segments. The purpose of image segmentation is to simplify or change the representation of an image, making it easier to understand and analyze. The applications of image segmentation include medical imaging, face recognition, traffic control system and so on. There are many methods for image segmentation including thresholding, k-means clustering, discrete MRF and so on.

The Chambolle-Pock (CP) algorithm (also called primal-dual algorithm) is a classic method which can be applied to various convex optimization problems such as image reconstruction and image segmentation. In this project we implement Chambolle-Pock algorithm for computing minimal partitions. This method is based on rewriting the minimal partition problem in terms of a primal dual Total Variation functional [1]. The CP algorithm consists in alternating a gradient-like ascent in the dual variable and a gradient-like descent in the primal variable. Besides, an over-relaxation in the primal variable is performed. The specific update scheme is explained in paper [2].

Our implementation is based on paper [3] and paper [2]. The comparison algorithm is based on paper [4]. We apply the algorithms to different images and the results show the effectiveness of Chambolle-Pock (CP) algorithm.

2 Algorithm Description

In this section, we first describe two projection algorithms, namely, projection onto L_2 ball and projection onto simplex which are used in the primal-dual algorithm for segmentation. And we briefly describe Dykstra's algorithm which is used for projecting to finite intersection of convex sets. Then we describe in details the primal-dual algorithm for image segmentation and how we can derive it from the original brief primal-dual algorithm. At last, we will briefly introduce other comparison algorithms like Zach's algorithm.

2.1 Projection onto L_2 Ball

We first introduce projection onto unit L_2 ball. Given $y \in \mathbb{R}^n$, the projection onto a set $S \subseteq \mathbb{R}^n$ is a function $P_s(y)$ given by

$$\hat{x} = P_s(y) = \underset{x \in \mathbb{R}^n}{\operatorname{argmin}} \|x - y\|_2 \tag{1}$$

The unit L_2 ball is defined as $S = \{x \in \mathbb{R}^n : \|x\|_2 \leq 1\}$. Projection onto L_2 ball has a unique global minimizer, which is

$$\underset{\|x\|_2 \leq 1}{\operatorname{argmin}} \|x - y\|_2 = \frac{y}{max\{1, \|y\|_2\}} \tag{2}$$

The intuition for this solution is as follows: If $y \in S$, then y is the closet one to itself and we get $\hat{x} = y$. If $y \notin S$, then $\|y\|_2 > 1$ and the closet point $\hat{x}$ is $\frac{y}{\|y\|_2}$ since the norm of $\frac{y}{\|y\|_2}$ is 1.

For L_2 ball with radius $r(r \neq 1)$, we simply replace the 1 by r in the equation.

2.2 Projection onto Simplex

There are many different algorithms for projection onto simplex. We introduce several simple algorithms here.

The first one due to Shalev-Shwartz and Singer is based on sorting.

The procedure is summarized below [5].

- **Input:** $\hat{\beta} \in \mathbb{R}^n$ and a scalar $z > 0$
- Sort $\hat{\beta}$ into μ such that $\mu_1 \geq \mu_2 \geq \cdots \geq \mu_n$
- Find $\rho = max\{m \mid \mu_m > \frac{1}{m}\left(\sum_{r=1}^{m} \mu_r - z\right)\}$
- Set $\tau = \frac{1}{\rho}\left(\sum_{r=1}^{\rho} \mu_r - z\right)$
- **Output:** β where $\beta_m = max(\hat{\beta}_m - \tau, 0)$

Another algorithm is due to Michelot. It is a recursive algorithm for projecting onto canonical simplex in $\mathbb{R}^n$. The maximum number of iterations is equal to the initial dimension of the space [6]. And this algorithm converges with at most n iterations. First, we define the notations.

$$K = \{x \mid \sum_{i=1}^{n} x_i = 1, x \geq 0, x \in \mathbb{R}^n\} \tag{3}$$

$$I_n = \{1, 2, \ldots, n\} \tag{4}$$

$$V = \{x \mid \sum_{i=1}^{n} x_i = 1, x \in \mathbb{R}^n\} \tag{5}$$

For a subset I of I_n, $X_I = \{x \in \mathbb{R}^n \mid x_i = 0, \forall i \in I\}$ and $V_I = X_I \cap V$. y is the point we want to project. The algorithm is as follows [6]:

- Initialization: $I = \varnothing$ and $x = y$
- Iteration:
  - Let $x \in X_I$ and $x \notin V_I$.
  - Compute $\hat{x} = P_{V_I}(x)$
  - If $\hat{x} \geq 0$, stop and output $\hat{x}$
  - Else, $I = I \cup \{i \mid \hat{x}_i < 0\}$, $x = P_{X_I}(\hat{x})$

2.3 Dykstra's Algorithm

Dykstra's algorithm is used in the projection of $\sigma_i \in \mathbb{R}^{(k \times d)}$ onto the convex K. The problem that needs to be solved is: we have k convex sets $K_1, K_2, \ldots, K_k$ in $\mathbb{R}^n$. And we want to compute the projection of a vector x onto $\cap_{i=1}^{k} K_i$. Let $\psi_i(z) = 0$ for $z \in K_i$ and $\psi_i(z) = \infty$ for $z \notin K_i$. Our objective is [3]:

$$\min_{z \in \mathbb{R}^N} \frac{\mid x - z \mid^2}{2} + \sum_{i=1}^{k} \psi_i(z) \tag{6}$$

The derivation of Dykstra's algorithm is to consider the dual problem:

$$\min_{y \in \mathbb{R}^N} \frac{\mid x - y \mid^2}{2} + \left(\sum_{i=1}^{k} \psi_i\right)^* (y) \tag{7}$$

But we omit the details here and only show the update scheme here [7].

- $x_i^{n+1} = \Pi_{K_i}(x_{i-1}^n + y_i^n)$
- $y_i^{n+1} = x_{i-1}^n + y_i^n - x_i^{n+1}$

2.4 Primal-Dual Algorithm for Image Segmentation

2.4.1 The General Problem and Primal-Dual Algorithm

The Chambolle-Pock (CP) algorithm [8] is a primal-dual algorithm. It solves the primal optimization problem and its dual problem at the same time. The CP algorithm applies to the following primal problem:

$$\min_{x} \{F(Kx) + G(x)\} \tag{8}$$

The dual problem is:

$$\max_{y} \{-F^*(y) - G^*(-K^T y)\} \tag{9}$$

About the notation: X and Y are two finite-dimensional real vector spaces with an inner product. x and y are vectors in the spaces X and Y respectively. K is a a continuous linear operator from X to Y. G and F are proper, convex and lower semi-continuous functions mapping the X and Y to $\mathbb{R}_{\geq 0}$ [8].

The original CP algorithm is summarized as follows [8]. In practice, we usually set θ to 1.

- Initialization: Choose τ, $\sigma > 0$, $\theta \in [0, 1]$, (x^0, y^0) and $\bar{x}^0 = x^0$
- Iterations: update x^n, y^n and $\bar{x}^n$
  - $y^{n+1} = (I + \sigma\partial F^*)^{-1}(y^n + \sigma K\bar{x}^n)$
  - $x^{n+1} = (I + \tau\partial G)^{-1}(x^n - \tau K^* y^{n+1})$
  - $\bar{x}^{n+1} = x^{n+1} + \theta(x^{n+1} - x^n)$

But we can not apply this update scheme directly and we need to derive the particular update scheme with respect to the specific problem. First we show some necessary notations and equations. Given a convex function H of a vector $z \in Z$, its conjugate can be computed by the Legendre transform:

$$H^*(z) = \max_{z'} \{\langle z, z' \rangle_Z - H(z')\} \tag{10}$$

And the proximal mapping is defined as:

$$prox_\sigma[H]_z = \underset{z'}{\text{argmax}} \{H(z') + \frac{\| z - z' \|_2^2}{2\sigma}\} \tag{11}$$

There are 5 steps as described in the paper [9] about CT image reconstruction. With these 5 steps we can derive a specific primal-dual algorithm for image segmentation, though in practice we have found the specific update scheme from paper [2]. And we will discuss it in the next subsection.

- Map the optimization problem to equation 8
- Using the Legendre transform equation 10 to derive the dual problem 9
- Derive the proximal mappings of F and G using equation 11
- Substitute the results of the third step into the generic CP algorithm described above to obtain a specific algorithm
- Run the algorithm and monitoring the primal-dual gap for convergence

2.4.2 Discrete Setting and Update Scheme

The formal definition for computing minimal partitions is as follows. We want to find a segmentation of an image into k pairwise disjoint regions, which minimizes the total interface between the sets. The following Mumford-Shah problem describes our objective:

$$\min_{(R_l)_{l=1}^k,(c_l)_{l=1}^k} \frac{1}{2}\sum_{l=1}^{k} Per(R_l;\Omega) + \frac{\lambda}{2}\sum_{l=1}^{k}\int_{R_l} | g(x) - c_l |^2 \, dx \tag{12}$$

About the notations: $g : \Omega \to \mathbb{R}$ is the input image. $c_l \in \mathbb{R}$ are optimal mean values. The regions R_l form a disjoint partition of Ω satisfying $\cup_{l=1}^k R_l = \Omega$.The parameter λ is used to balance the data term and the length term [8].

The more generic representation is:

$$\min_{(u_l)_{l=1}^k} J(u) + \sum_{l=1}^{k}\int_\Omega u_l f_l dx \tag{13}$$

where $u_l(x) \geq 0$, $\sum_{l=1}^k u_l(x) = 1$ for all $x \in \Omega$. $u = (u_l)_{l=1}^k$ is the labeling function and $f_l = \frac{\lambda}{2} \mid g(x) - c_l \mid^2$ or other weighting functions [8]. The relaxation of CP algorithm is

$$J(u) = \int_\Omega \Psi(Du) \tag{14}$$

where $p = (p_1, \ldots, p_k)$, $q = (q_1, \ldots, q_k)$ and

$$\Psi(p) = \sup_q \{\sum_{l=1}^{k}\langle p_l, q_m\rangle : \mid p_l - q_m \mid \leq 1, 1 \leq l < m \leq k\} \tag{15}$$

We need to consider discretization of total variation. First we show the relaxed convex problem from Zach's algorithm (which we will describe shortly in next section) [2].

$$\min_{u\in\mathcal{B}} \sup_{\xi\in\mathcal{K}^{\lambda g}} \{\sum_{i=1}^{n}\int_\Omega u_i f_i dx - \sum_{i=1}^{n}\int_\Omega u_i \, div\xi_i dx\} \tag{16}$$

where $\mathcal{B}$ is the set of function u and $\mathcal{K}^{\lambda g}$ in CP algorithm is

$$\begin{aligned}\mathcal{K}^{\lambda g} = \{\xi \in C_c^1(\Omega, \mathbb{R}^2)^{n+1} \mid \mid \xi_i(x) - \xi_j(x) \mid \leq \lambda g(x), \\ 1 \leq i < j \leq (n+1), \forall x \in \Omega\}\end{aligned} \tag{17}$$

We apply 17 to the energy in 16 by means of Lagrange multipliers $\mu_{ij} : \Omega \to \mathbb{R}^2$ and additional variables $q_{ij} = \xi_i - \xi_j \in \mathcal{Q}$ where $\mathcal{Q}$ is defined as

$$\begin{aligned}\mathcal{Q} = \{q_{ij} \in C_c^1(\Omega, \mathbb{R}^2) \mid \mid q_{ij}(x) \mid \leq \lambda g(x), \\ 1 \leq i < j \leq (n+1), \forall x \in \Omega\}\end{aligned} \tag{18}$$

and then we obtain ($\xi_{n+1} = 0$) [2]:

$$\begin{aligned}\min_{u\in\mathcal{B},\mu_{\xi_i,q_{ij}}} \sup \{&\sum_{i=1}^{n}\int_\Omega u_i(f_i - div \, \xi_i)dx \\ &+ \sum_{1\leq i<j\leq(n+1)}\int_\Omega \mu_{ij}(\xi_i - \xi_j - q_{ij})dx\}\end{aligned} \tag{19}$$

For this discrete setting we have following update scheme from paper [2]:

- $\xi^{t+1} = \xi^t + \frac{1}{2+n}\left(\nabla\bar{u}^t + (\sum_{j>i}\bar{\mu}_{ij}^t - \sum_{i>j}\bar{\mu}_{ji}^t)_{1\leq i\leq n}\right)$
- $q^{t+1} = \Pi_{\mathcal{Q}}(q^t + (-\bar{\mu}))$
- $u^{t+1} = \Pi_{\mathcal{B}}\left(u^t - \frac{1}{4}(f - div \, \xi^{t+1})\right)$
- $\mu^{t+1} = \mu^t - \frac{1}{3}(\xi_i - \xi_j - q_{ij})$
- $\bar{u}^{t+1} = 2u^{t+1} - u^t$
- $\bar{\mu}^{t+1} = 2\mu^{t+1} - \mu^t$

Notice that the $\Pi_{\mathcal{B}}$ is the projection onto simplex. And $\Pi_{\mathcal{Q}}$ is the projection to intersection of unit ball where we use the Dykstra's algorithm.

2.5 A brief description of Zach's Algorithm

Zach's algorithm is one of the comparison algorithms we used in this project. It uses a simple relaxation which is the sum of the total variation of each labeling function v_i. This relaxation is not as tight as in the primal-dual algorithm and we will show the comparison of results later. The relaxation is as follows [4]:

$$J(v) = \frac{1}{2} \sum_{i=1}^{k} \int_{\Omega} \mid Dv_i \mid \tag{20}$$

And the update scheme is also simpler. About notation: π_S is projection onto canonical simplex. The update scheme is as follows [4].

- $v_x = \pi_S({u_x}^{(t)} - \lambda\theta c_x)$
- ${u_x}^{(t+1)} = v_x + \theta\nabla \cdot {p_x}^{(t)}$
- $\tilde{p}_x = {p_x}^{(t)} + \frac{\tau}{\theta}\nabla {u_x}^{(t+1)}$
- ${p_x}^{(t+1)} = max(-1, min(1, \tilde{p}_x))$

3 Program Implementation

A primal-dual method is used to implement the Potts model by the approach of both Chambolle and Zach's convex relaxation, which are based on the algorithm described above. We use **OpenCV**, **matplotlib** packages via python to read, write and show images, **sklearn.cluster** packages to do K-means clustering and also take advantage of **scipy.sparse** packages for fast matrices computation.

3.1 Chambolle's potts model

The main step is constructed by function *potts_chambolle* which receives image as K (number of partitions) fields as input and outputs the segmented fields of the image. In addition, we still need to deal with the read image for certain number (K) of phases and show it correctly after multi-label partitions.

In the key function of potts model, firstly, initialization should be carried out on two aspects: one is step sizes of primal-dual method chosen using diagonal preconditioning, and the other one is primal variables and dual variables according to the shape of image fields and dual constraints. Secondly, we compute forward difference by discrete gradient operators *diffop_1d* and *diffop_2d* using sparse matrix. *diffop_1d* creates 1-dimensional sparse difference operator ∂d, such that

$$\partial d = \begin{cases} -1, & i = j < d \\ +1, & j = i+1 \\ 0, & otherwise. \end{cases}$$

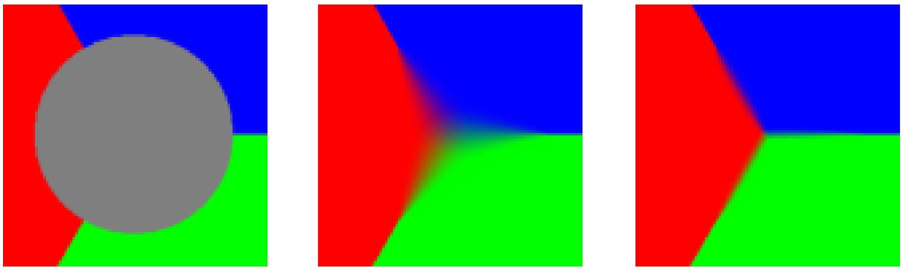
diffop_2d calls *diffop_1d* of the image shape in one field and compute Kronecker product of the output sparse matrices and identity matrices. The forward difference is obtained by concatenation of two Kronecker product results. Thirdly, we perform primal and dual updates. In the primal update step, each layer of the fields is updated via Chambolle's convex relaxation. Then we need a simplex projection *proj_simplex* onto the primal variable of the same shape with the input image fields. After that, overrelaxation operation is applied through the difference between updated primal variables and original ones at current iteration. In the dual update, besides similar update steps, we need to compute projection of dual variable with dual constraints onto the l2-norm. Finally, we check the total variation (TV) value in gap step.

Before we apply *potts_chambolle* function described above, we need to set up mean intensities of the image clusters. We totally consider two cases: 1) For the problem of synthetic triple-junction and quad-junction image segmentation, whose optimal solution is given by a junction with relative angle, we use certain calibration of color intensity values. 2) For basic segmentation problems, we use **sklearn** K-means clustering to obtain clusters for each label. Then *squared_dist* is called to compute field that denotes distance between the original image and its means of clusters, which gives mean intensities of clusters as input of the main model. In the last step, we compute a piecewise-constant image using the mean intensities and output from *potts_chambolle*.

3.2 Zach's potts model for Comparison

The key function *potts_zach* is implemented in a similar way with *potts_chambolle*. The main differences lie in the relaxation method from Zach, in which additional dual constraints is not used during primal and dual update. What's more, instead of a loop for updating the projection of dual variable onto mixed $l_{1,2}$ norm, a vectorized matrix computation of all clusters is preferred here. It will be faster for much larger number of labels.

The run time of tested portrait segmentation step of Chambolle's relaxation for 1000 iterations, 6 labels is 150.5 seconds with CPU Intel(R) Core(TM) i5-8257U, 4 cores, clock rate of 1.4 GHz (3.9 GHz in turbo mode). While in the same condition, Zach's relaxation takes 89.0 seconds, which only costs 60% time of Chambolle's. As we use doubled labels, Zach's method takes



(a) input (b) Zach (c) Chambolle

Figure 1: Experimental comparison of different relaxations for reconstruction of a triple point.

231.6 seconds, while Chambolle's methods takes 667.9 seconds, nearly 3 times of the former one.

3.3 Explanation for Extra Functions

Due to no source of synthetic quad-junction image, we have to reformat the screenshot of the figure in the paper. *make_col4* is used to produce such image that transform the input image from 4 channels (RGBA) to 3 channels (RGB) and round the pixel values to get exact red, green, blue colors. Note here the center should be made into gray in order to achieve correct results.

The **matplotlib.image** library read image in different ways according to different formats of the image. We mainly test two kinds of images (.png and .jpg), so the latter one should be adjusted to range [0, 1].

4 Experimental Results

There are three convex relaxations of the minimal partition in the problem of multi-label segmentation, proposed by Lellmann[10], by Zach[4] and by Chambolle[1]. The differences of these approaches lie in the regularization norm, in computational complexity and in tightness of the relaxation. Since the Lellmann's relaxations was shown to be not as tight as the latter two, we decide to concentrate on more accurate ones in the following experiment.

4.1 Reconstruction of Triple and Quad points

We first consider the problem of synthetic three-label image segmentation, where the optimal segmentation is given by a 120° triple junction. We set the three color intensities as (1, 0, 0), (0, 1, 0), (0, 0, 1). It's shown in Figure 1 that the proposed method finds a binary solution whereas Zach's method ends up with a non-binary solution.

Next we consider the problem of synthetic four-label image segmentation, where the optimal segmentation is given by a 90° triple junction. We set the three color intensities as (1, 0, 0), (0, 1, 0), (0, 0, 1), (1, 1, 1). Surprisingly, both methods cannot find a binary solution, but there's difference between their performance. To be more clear, we present gray scale of different channels (shown as white region) for the two methods in Figure 3. The segmentation boundary in Chambolle's solution is sharper, which seems that it find a convex combination of the two minimal binary solutions.

4.2 Basic Piecewise Constant Segmentation

As is shown in Figure 4, we test a scenery photo for 6 and 12 labels separately. The two methods achieve nearly the same results and they both have different regions segmented successfully. However, as the number of label increases, one white region at upper left disappears and one green region at middle right appears, which may be false. Therefore, we modify the fidelity parameter λ to be 5, which results in more accurate performance in Figure 5.

Next, we test both methods on a portrait for $\lambda = 5$. As we can see in Figure 6, the two methods do a good job in piecewise constant Mumford-Shah segmentation[11], which produce spatially consistent boundaries. As we double the number of labels, the color of each region becomes more precise.

5 Convergence Analysis

In this section we discuss the preconditioning, acceleration and convergence of Chambolle-Pock algorithm.

As we described above, in practice we usually use primal-dual algorithm with $\theta = 1$. Under this setting,

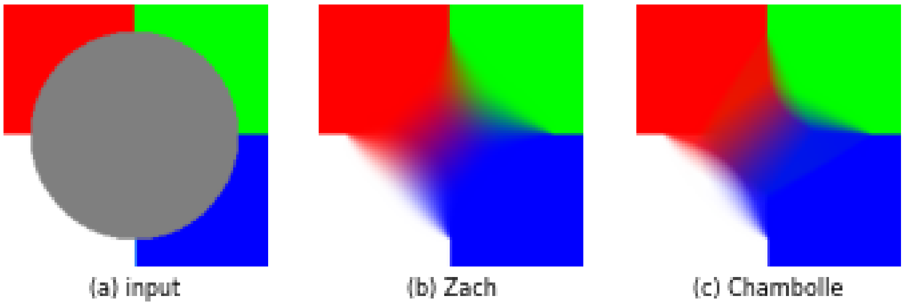


Figure 2: Experimental comparison of different relaxations for reconstruction of a quad point.

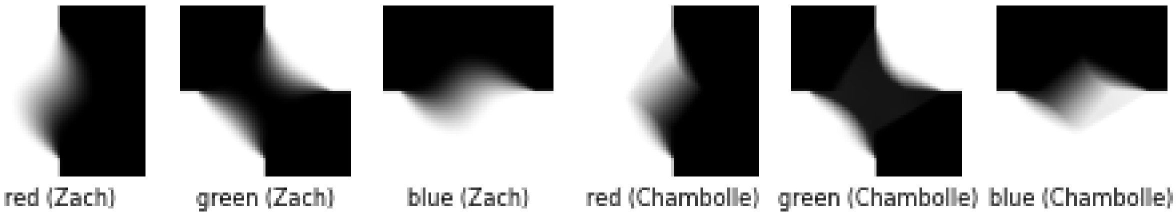


Figure 3: Gray scale comparison in separate red, green, blue channels of different relaxations for reconstruction of a quad point.

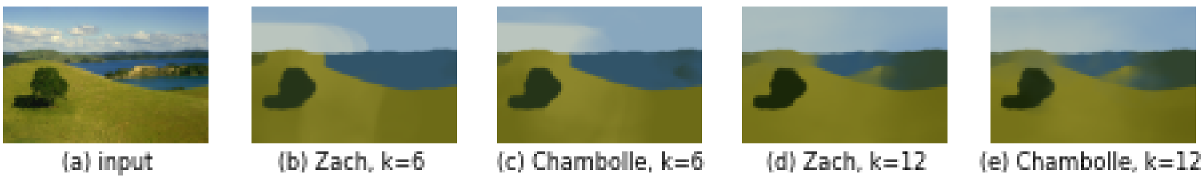


Figure 4: Segmentation of a scenery image using a different number of labels for different relaxations, $\lambda = 1$.

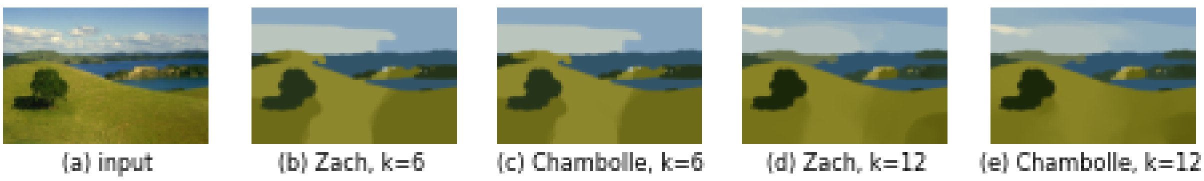
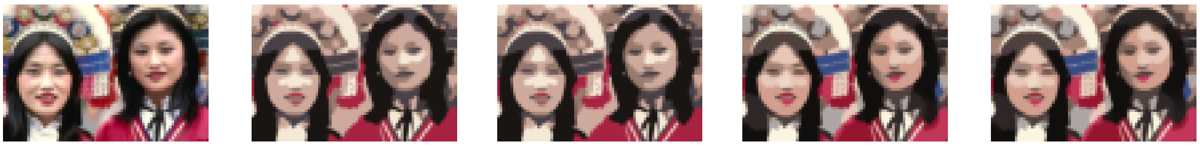


Figure 5: Segmentation of a scenery image using a different number of labels for different relaxations, $\lambda = 5$.



(a) input (b) Zach, k=6 (c) Chambolle, k=6 (d) Zach, k=12 (e) Chambolle, k=12

Figure 6: Segmentation of a portrait using a different number of labels for different relaxations, $\lambda = 5$.

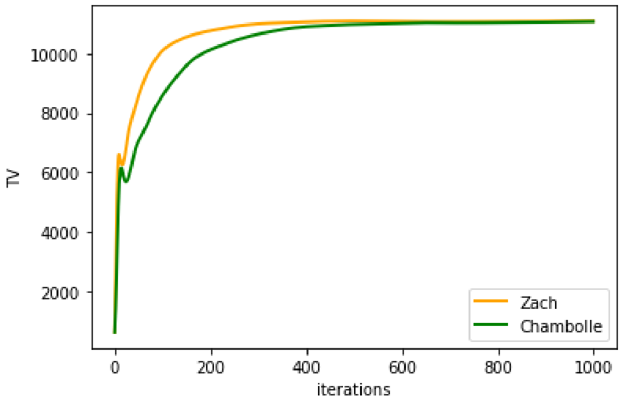


Figure 7: Convergence of the primal dual algorithm for portrait example of Figure 6, with k=12.

if additionally $\tau\sigma L^2 < 1$ with $L = \|K\|$ and X and Y are finite dimensional spaces, the primal-dual algorithm converges [8]. In our implementation, we noticed that the primal-dual algorithm for segmentation takes much time to get the result. From Figure 7 we can see that Zach's algorithm converges faster than primal-dual algorithm. This is due to the original primal-dual algorithm computes the projections by solving auxiliary optimization problems. The disadvantage is that the inner optimization problems have to be computed with sufficient accuracy such that the overall algorithm converges, which can heavily slow down the algorithm [12]. Thus, in order to overcome this shortcoming, another paper of Chambolle and Pock [12] discusses preconditioning techniques for the primal-dual algorithm. The preconditioned algorithm significantly accelerates the convergence on problems and is equivalent to the original algorithm. Compared with the original primal-dual algorithm, τ and σ are replaced with preconditioning matrices T and Σ. For image segmentation problem, they introduce slack variables and Lagrange multipliers λ and in the end yields a specific algorithm which does not need to solve any inner optimization problems. We can reasonably deduce that the preconditioned algorithm works faster than the original one. And the experiment of the paper [12] proves it by showing that in all test cases, the preconditioned algorithm outperforms the original one.

There is also another accelerated primal-dual algorithm described in the paper [8]. It is a variant of the original Chambolle-Pock algorithm and uses variable step sizes.

6 Conclusion

In this report, we study, implement and compare two convex relaxations (Zach and Chambolle) multi-label image segmentation problem. First, we introduce the basic architecture and inference of Chambolle's potts model, with a brief description of Zach's relaxation. Then, we explain how we construct our program with the meaning of each function. Next, we show our results by comparing the two methods for two kind of problem. Finally, we analyzed the convergence in theory and experimental performance.

We encountered some interesting problems while going through this project. In the problem of reconstructing a triple point, we find that it can only work for a gray center (whose RGB value is (0.5, 0.5, 0.5)). If the center color is black, the black color will polluted the around colors; if the center color is white, there's no change after segmentation steps. We guess it's because the RGB value of gray gives equal probability for calibration of each color as it should be. In addition, it's surprising that we find a combination of two binary solutions in the problem of quad point reconstruction without any rotation, which is different from the original paper. What's more, it's amazing that Chambolle's method makes the combination entirely separated and not mixed. We can still see the boundary between two minimal solutions clearly, while in Zach's method, the solution is totally non-binary. Finally, we want to point out a question about how to choose a meaningful fidelity parameter. If it's low, we may miss some regions of the image which maybe really significant. If it's high, will we waste our work for a segmentation result same as the original image?

References

[1] Pock, Thomas ; Chambolle, Antonin ; Cremers, Daniel ; Bischof, Horst: A convex relaxation approach for computing minimal partitions. In: *2009 IEEE Conference on Computer Vision and Pattern Recognition* IEEE, 2009, S. 810–817 1, 5

[2] Nieuwenhuis, Claudia ; Töppe, Eno ; Cremers, Daniel: A survey and comparison of discrete and continuous multi-label optimization approaches for the Potts model. In: *International journal of computer vision* 104 (2013), Nr. 3, S. 223–240 1, 3

[3] Chambolle, Antonin ; Cremers, Daniel ; Pock, Thomas: A convex approach to minimal partitions. In: *SIAM Journal on Imaging Sciences* 5 (2012), Nr. 4, S. 1113–1158 1, 2

[4] Zach, Christopher ; Gallup, David ; Frahm, Jan-Michael ; Niethammer, Marc: Fast Global Labeling for Real-Time Stereo Using Multiple Plane Sweeps. In: *VMV*, 2008, S. 243–252 1, 4, 5

[5] Blondel, Mathieu ; Fujino, Akinori ; Ueda, Naonori: Large-scale multiclass support vector machine training via euclidean projection onto the simplex. In: *2014 22nd International Conference on Pattern Recognition* IEEE, 2014, S. 1289–1294 2

[6] Michelot, Christian: A finite algorithm for finding the projection of a point onto the canonical simplex of $\mathbb{R}^n$. In: *Journal of Optimization Theory and Applications* 50 (1986), Nr. 1, S. 195–200 2

[7] Boyle, James P. ; Dykstra, Richard L.: A method for finding projections onto the intersection of convex sets in Hilbert spaces. In: *Advances in order restricted statistical inference*. Springer, 1986, S. 28–47 2

[8] Chambolle, Antonin ; Pock, Thomas: A first-order primal-dual algorithm for convex problems with applications to imaging. In: *Journal of mathematical imaging and vision* 40 (2011), Nr. 1, S. 120–145 2, 3, 7

[9] Sidky, Emil Y. ; Jørgensen, Jakob H. ; Pan, Xiaochuan: Convex optimization problem prototyping for image reconstruction in computed tomography with the Chambolle–Pock algorithm. In: *Physics in Medicine & Biology* 57 (2012), Nr. 10, S. 3065 3

[10] Lellmann, Jan ; Kappes, Jörg ; Yuan, Jing ; Becker, Florian ; Schnörr, Christoph: Convex multi-class image labeling by simplex-constrained total variation. In: *International conference on scale space and variational methods in computer vision* Springer, 2009, S. 150–162 5

[11] Mumford, David ; Shah, Jayant: Optimal approximations by piecewise smooth functions and associated variational problems. In: *Communications on Pure and Applied Mathematics* 42 (1989), S. 577–685 5

[12] Pock, Thomas ; Chambolle, Antonin: Diagonal preconditioning for first order primal-dual algorithms in convex optimization. In: *2011 International Conference on Computer Vision* IEEE, 2011, S. 1762–1769 7