

Efficient Convex Optimization for Minimal Partition Problems with Volume Constraints

Thomas Möllenhoff, Claudia Nieuwenhuis, Eno Töppe, and Daniel Cremers

Department of Computer Science, Technical University of Munich, Germany

Abstract. Minimal partition problems describe the task of partitioning a domain into a set of meaningful regions. Two important examples are image segmentation and 3D reconstruction. They can both be formulated as energy minimization problems requiring minimum boundary length or surface area of the regions. This common prior often leads to the removal of thin or elongated structures. Volume constraints impose an additional prior which can help preserve such structures. There exist a multitude of algorithms to minimize such convex functionals under convex constraints. We systematically compare the recent Primal Dual (PD) algorithm [1] to the Alternating Direction Method of Multipliers (ADMM) [2] on volume-constrained minimal partition problems. Our experiments indicate that the ADMM approach provides comparable and often better performance.

1 Introduction

Both segmentation and 3D reconstruction approaches aim at partitioning a two or three dimensional domain into a set of 'meaningful' regions. For segmentation such regions usually correspond to the projections of three-dimensional objects in an image, whereas for 3D reconstruction the regions indicate the three dimensional objects themselves, e.g. in a voxel grid. Such minimal partition problems come with regularization assumptions which either promote a minimal boundary length [3–5] or a minimal surface area [6, 7]. These constraints often over-smooth the object boundaries. In segmentation the removal of thin or elongated structures is the consequence. In 3D reconstruction the resulting objects tend to be flat due to the minimal surface assumption. Therefore, constraints restricting the object volume for specific regions are useful as an additional regularization assumption to help preserve thin structures and obtain plastic object shapes, see Figure 1. Such volume constraints can be adapted interactively if the corresponding optimization problem can be solved in an efficient manner. Minimal partition problems

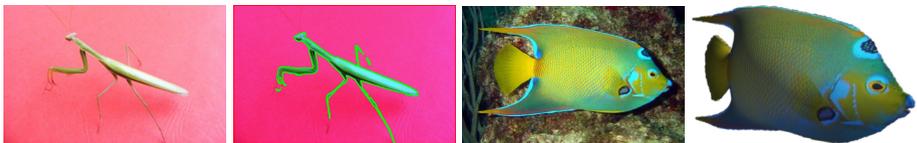


Fig. 1. Results of minimal partition problems (2D segmentation and 3D reconstruction from a single image) with additional volume prior

can be formulated as energy minimization problems over a continuous domain based on the commonly used total variation functional [8–10]. A bunch of algorithms exists for optimization of the resulting functionals. In this paper the focus will especially lie on the identification of efficient optimization algorithms with respect to the integration of additional constraints such as the volume priors.

2 Related Work

The PD and ADMM algorithms have been previously applied to optimize energies resulting from minimal partition problems. In particular, Goldstein et al. [11] used ADMM to solve the two region segmentation problem amongst other geometrical problems. Recently, ADMM was also applied within several multi-region segmentation approaches [12, 13]. Niethammer et al. [14] solved the two region segmentation problem with volume constraints using a branch and bound method. They employ ADMM to solve a specific sub-problem in each iteration.

The PD algorithm [15, 16, 10, 1] has become a standard algorithm for solving continuous constrained convex optimization problems with saddle-point structure. It was applied to multi-region segmentation without volume constraints [10, 5] as well as to single view 3D reconstruction [6].

3 Contributions

Both the PD and ADMM algorithm are well known and thoroughly studied, and they have been applied to minimal partition problems before. In this paper we make the following contributions:

- We impose volume constraints on the standard n region minimal partition problem and describe how to apply the ADMM and PD algorithm to minimize the resulting energies. We show that especially ADMM is able to handle projections onto the resulting constraint sets without closed form solution for their combined projection.
- We systematically compare the performance of both algorithms. While ADMM and PD have been compared on standard problems in Computer Vision before, (see [10]) we compare them on a lesser studied problem and also provide performance comparisons on a recent GPU.

4 Minimal Partition Problems with Volume Constraints

Let $\Omega \subset \mathbb{R}^d$ denote the partition domain, i.e. $d = 2$ for segmentation of 2D images, and $d = 3$ for 3D reconstruction. The task of partitioning the domain Ω into a set of n pairwise disjoint regions Ω_i , with $\Omega = \bigcup_{i=1}^n \Omega_i$, $\Omega_i \cap \Omega_j = \emptyset \quad \forall i \neq j$, can be solved by computing a labeling $l : \Omega \rightarrow \{1, \dots, n\}$ indicating which of the n regions each pixel/voxel belongs to: $\Omega_i = \{x \in \Omega : l(x) = i\}$.

Recently, minimal partition problems have been formulated as energy minimization problems on the basis of functions of bounded variation (BV), i.e. functions for which

the total variation (TV) is finite. The key idea is to encode the regions Ω_i by their indicator function $u \in \text{BV}(\Omega, \{0, 1\})^n$

$$u_i(x) = \begin{cases} 1, & \text{if } l(x) = i \\ 0, & \text{otherwise} \end{cases} \quad \forall i = 1, \dots, n \quad (1)$$

and to solve for u_i using convex relaxation techniques. For segmentation and 3D reconstruction based on volumetric constraints we can formulate the following convex relaxed optimization problem:

$$\min_{u \in \text{BV}(\Omega, [0,1])^n} \sum_{l=1}^n \int_{\Omega} \|Du_l(x)\| + \int_{\Omega} f_l(x)u_l(x) dx \quad (2)$$

$$\text{s.t. } u_l(x) \geq 0, \quad \sum_{l=1}^n u_l(x) = 1 \quad \forall x \in \Omega, \quad \int_{\Omega} u_l(x) dx = \mathcal{V}_l, \quad l \in \{1, \dots, n\}. \quad (3)$$

The total variation of u in (2) measures the length of the region boundaries or the size of the region surfaces, respectively, and the data term $f : \Omega \rightarrow \mathbb{R}^n$ indicates how strongly a pixel is associated with each of the n regions. The first two constraints in (3) together form a simplex constraint, which ensures that each pixel is assigned to exactly one label. The linear volume constraints impose a specific volume \mathcal{V}_l on each label. To obtain a binary solution after solving the relaxed problem each pixel is assigned to the label with maximum indicator function value $u_i(x)$. In the two label case, i.e. $n = 2$ this operation will yield a globally optimal solution of the binary problem [17]. Due to the simplex constraint we obtain $u_2 = 1 - u_1$ for this case and thus

$$\min_{u_1 \in \text{BV}(\Omega, [0,1])} \int_{\Omega} \|Du_1(x)\| + \int_{\Omega} (f_1(x) - f_2(x))u_1(x) dx \quad (4)$$

$$\text{s.t. } 0 \leq u_1(x) \leq 1, \quad \int_{\Omega} u_1(x) dx = \mathcal{V}_1, \quad (5)$$

which is the original foreground/background segmentation and single view 3D reconstruction functional of [6]. In this case, the data term $f_1 - f_2$ is chosen so the projection of the segmentation onto the image plane corresponds to the silhouette of the object.

The main problem in the optimization of these functionals are the constraints in (3), which must all be fulfilled simultaneously. One of the simplest optimization strategies for constrained optimization is *Projected Gradient Descent*. However, since the total variation is non-smooth, this method cannot be applied directly. Furthermore, there exists no closed form projection onto the feasible set in (3). Projecting the primal variable onto the intersection of several feasible sets by Dykstra's projection algorithm [18] in every iteration is inexact and inefficient.

A well-known algorithm for non-smooth problems involving the total variation is the *Fast Iterative Shrinkage-Thresholding Algorithm (FISTA)* [19]. However, it turns out that this algorithm also requires a projection onto the feasible set in (3) in every iteration, which makes the algorithm slow and inexact.

Therefore, algorithms which can handle constraints simultaneously without closed form solution for their combined projection are required to obtain fast and accurate

solutions. ADMM can handle such constraints by separate projections on each feasible set. Apart from ADMM there is the Primal Dual algorithm [1, 10], which can handle constraints by the method of Lagrange multipliers.

5 Efficient Convex Optimization

In the following we will present the ADMM and the PD method in detail and see how to apply them to the volume constrained minimal partition task. In general, these algorithms solve problems of the form

$$\min_{x \in \mathbb{R}^n} F(Kx) + G(x) \quad (6)$$

where $K \in \mathbb{R}^{n \times m}$ is a linear operator, and $F : \mathbb{R}^m \rightarrow \mathbb{R} \cup \{\infty\}$, $G : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{\infty\}$ are proper, closed and convex functions.

Alternating Direction Method of Multipliers. We consider a slightly generalized version of ADMM in *scaled form*, which is given by the update scheme

$$\begin{cases} x^{k+1} = \arg \min_{x \in \mathbb{R}^n} G(x) + \frac{\tau}{2} \|Kx - y^k + \lambda^k\|_2^2 \\ r^{k+1} = \alpha Kx^{k+1} + (1 - \alpha)y^k \\ y^{k+1} = \arg \min_{y \in \mathbb{R}^m} F(y) + \frac{\tau}{2} \|r^{k+1} - y + \lambda^k\|_2^2 \\ \lambda^{k+1} = \lambda^k + r^{k+1} - y^{k+1} \end{cases} \quad (7)$$

where $\alpha \in (0, 2)$ is an over- or under-relaxation factor, and τ is a step size parameter. For a derivation of this algorithm we refer the interested reader to [20]. A possible interpretation of this algorithm is, that it alternately optimizes the Augmented Lagrangian L_τ of the primal problem (6) in the variables x , y and λ :

$$L_\tau(x, y, \lambda) = F(y) + G(x) + \langle \lambda, Kx - y \rangle + \frac{\tau}{2} \|Kx - y\|_2^2$$

This algorithm was first studied by Eckstein et. al. in [2], a proof of convergence of the algorithm is given by Theorem 8 of that paper.

Preconditioned Primal Dual Algorithm. We also consider the recent algorithm by Chambolle et al. [16, 10, 1] for finding a saddle point of the primal-dual formulation of the problem (6)

$$\min_{x \in \mathbb{R}^n} \max_{y \in \mathbb{R}^m} \langle Kx, y \rangle + G(x) - F^*(y) \quad (8)$$

where F^* denotes the convex conjugate of F . The update steps of the algorithm are

$$\begin{cases} x^{k+1} = (\text{id} + T\partial G)^{-1}(x^k - TK^T y^k) \\ \bar{x}^{k+1} = x^{k+1} + \theta(x^{k+1} - x^k) \\ y^{k+1} = (\text{id} + \Sigma\partial F^*)^{-1}(y^k + \Sigma K(\bar{x}^{k+1})) \end{cases} \quad (9)$$

where T and Σ are preconditioning matrices that can be seen as generalized step sizes. This algorithm alternatingly performs gradient descent and ascent steps on (8) in x and y . For further information on the algorithm we refer the interested reader to [1]. Furthermore, it has been shown in [10] that the ADMM algorithm (with $\alpha = 1$) and the PD algorithm are identical, if K is equal to the identity matrix. However, in the general case it is often necessary to solve a least squares problem involving the matrix K in the ADMM algorithm, while the PD algorithm has a closed for solution for the corresponding step.

5.1 Notation and Discretization

In the following we discretize the continuous domain into a regular grid of width W , height H and in the 3D case depth D . To make notation easier we introduce the index set $\Omega = [1 \dots W] \times [1 \dots H]$ to address the individual grid elements of $x \in \mathbb{R}^{WH}$ as x_i for some $i \in \Omega$ (and we define everything analogously in the 3D case).

For the discretization of the gradient operator ∇ we use forward differences and von Neumann boundary conditions. Due to the duality of the gradient and divergence operator the divergence operator is defined as $\text{div} = -\nabla^T$, in order to satisfy the identity $\langle \nabla x, y \rangle = -\langle x, \text{div} y \rangle$. We define the discrete Laplacian as $\Delta = -\nabla^T \nabla$.

Furthermore, we denote the isotropic discretization of the weighted TV

$$TV_g(x) = \int_{\Omega} g(t) \|\nabla x(t)\|_2 dt \quad \text{as} \quad \|\nabla x\|_g = \sum_{i \in \Omega} g_i \|(\nabla x)_i\|_2.$$

We will also make frequent use of the indicator function ι_C of a set C

$$\iota_C(x) := \begin{cases} 0, & x \in C \\ \infty, & x \notin C \end{cases}$$

5.2 Application to Single View 3D Reconstruction

In the discrete setting we can write the continuous formulation of the single view 3D reconstruction problem (4) as

$$\min_{u \in U \cap V} \|\nabla u\|_g + \langle u, f \rangle \quad (10)$$

where the set U corresponds to the simplex constraint and the set V to the volume constraint in (3). In particular, we define the constraints as

$$U = \{u \in X : 0 \leq u_i \leq 1\} \quad \text{and} \quad V = \left\{ u \in X : \sum_{i \in \Omega} u_i = \mathcal{V} \right\}$$

with $X = \mathbb{R}^{WHD}$ for a domain of width W , height H and depth D .

Alternating Direction Method of Multipliers. We will now apply ADMM to problem (10). We set $K : X \rightarrow Y$ as $Ku = (\nabla u \ u)^T$ and write the constraints as indicator functions ι_U and ι_V to arrive at the following equivalent formulation:

$$\min_{u \in X} \underbrace{\|\nabla u\|_g + \iota_U(u)}_{F(Ku)} + \underbrace{\langle u, f \rangle + \iota_U(u)}_{G(u)} \quad (11)$$

Applying the ADMM scheme (7) yields the following algorithm:

$$\begin{cases} u^{k+1} = \arg \min_{u \in X} \langle u, f \rangle + \iota_U(u) + \frac{\tau}{2} \|(\nabla u) - (v^k) + (\lambda_1^k) + (\lambda_2^k)\|_2^2 \\ (r_1^{k+1}, r_2^{k+1}) = \alpha (\nabla u^{k+1}) + (1 - \alpha) (v^k) \\ (v^{k+1}, w^{k+1}) = \arg \min_{(v \ w)^T \in Y} \|v\|_g + \iota_V(w) + \frac{\tau}{2} \|(r_1^{k+1}) - (v) + (\lambda_1^k) + (\lambda_2^k)\|_2^2 \\ (\lambda_1^{k+1}, \lambda_2^{k+1}) = (\lambda_1^k) + (r_2^{k+1}) - (v^{k+1}, w^{k+1}) \end{cases} \quad (12)$$

The update steps in r_1, r_2, λ_1 and λ_2 are just simple arithmetic operations. It remains to show how to solve the sub-optimization problems in u, v and w . Note that the coupled optimization problem in v and w is separable into two independent sub-problems.

Solving the sub-problem in u . Ignoring the constraint ι_U which corresponds to $u \in U$, the minimization problem is differentiable. The required optimality conditions are given by the Euler-Lagrange equation:

$$\begin{aligned} \frac{\partial E}{\partial u} &= \frac{f}{\tau} + (u - w^k + \lambda_2^k) - \operatorname{div}(\nabla u - v^k + \lambda_1^k) = 0 \\ \Leftrightarrow u &= (\mathcal{I} - \Delta)^{-1} \left(-\frac{f}{\tau} + w^k - \lambda_2^k - \operatorname{div} v^k + \operatorname{div} \lambda_1^k \right) \end{aligned}$$

This smooth, constrained problem can be approximately solved by the *Projected Gradient Descent* method:

$$\hat{u}^{i+1} = \operatorname{proj}_U(\hat{u}^i - \gamma \frac{\partial E}{\partial u})$$

with stepsize γ for $i = 1 \dots N - 1$ and $\hat{u}^1 = u^k$, and finally setting $u^{k+1} = \hat{u}^N$.

However, we chose to approximately solve for u by alternating Jacobi iterations with point-wise projection to the constraint set U in the same fashion as [11]. In our experiments we found this to be slightly faster than Projected Gradient Descent, while producing the same final result. We noticed that using only two iterations to solve this sub-problem seem to suffice to make the ADMM algorithm converge.

Solving the sub-problem in v . The closed-form solution for the minimizer of the optimization problem

$$v^{k+1} = \arg \min_v \|v\|_g + \frac{\tau}{2} \|v - (r_1^k + \lambda_1^k)\|_2^2$$

is given by the *coupled shrinkage* formula

$$v^{k+1} = \text{shrink}_g(r_1^k + \lambda_1^k, \tau)$$

which we define similarly to [11, 21] as

$$(\text{shrink}_g(x, \tau))_i = \begin{cases} 0, & \text{if } \|x_i\|_2 \leq \frac{g_i}{\tau} \\ x_i - \frac{g_i}{\tau} \frac{x_i}{\|x_i\|_2}, & \text{otherwise.} \end{cases}$$

Solving the sub-problem in w . The closed form solution to the optimization problem in w is given by the orthogonal projection of $r_2^k + \lambda_2^k$ onto the convex set V .

$$w^{k+1} = \arg \min_w \iota_V(w) + \frac{\tau}{2} \|w - (r_2^k + \lambda_2^k)\|_2^2 = \text{proj}_V(r_2^k + \lambda_2^k).$$

Furthermore, a short calculation shows that

$$\text{proj}_V(x) = x - \frac{1}{|\Omega|} \left(\sum_{i \in \Omega} x_i - \mathcal{V} \right)$$

where the set V is defined as above.

Preconditioned Primal Dual Algorithm. We formulate the single view 3D reconstruction functional as the saddle-point problem

$$\min_{u \in X} \max_{\xi \in Y_1, \lambda \in Y_2} \underbrace{\langle \nabla u, \xi \rangle + \langle \lambda, \mathbf{1}^T u \rangle}_{\langle Ku, (\xi \ \lambda)^T \rangle} + \underbrace{\langle u, f \rangle + \iota_U(u)}_{G(u)} - \underbrace{(\iota_P(\xi) + \langle \lambda, \mathcal{V} \rangle)}_{F^*((\xi \ \lambda)^T)} \quad (13)$$

where $P = \{\xi \in Y_1 : \|\xi\|_\infty \leq g\}$. Applying scheme (9) yields:

$$\begin{cases} u^{k+1} = \text{proj}_U(u^k - T(-\text{div } \xi^k + \lambda^k - f)) \\ \bar{u}^{k+1} = u^{k+1} + \theta(u^{k+1} - u^k) \\ \xi^{k+1} = \text{proj}_P(\xi^k + \Sigma_1 \nabla \bar{u}^{k+1}) \\ \lambda^{k+1} = \lambda^k + \Sigma_2 (\sum_i \bar{u}_i^{k+1} - \mathcal{V}) \end{cases} \quad (14)$$

For a detailed derivation of these update equations, we refer the interested reader to [1], where the algorithm is applied to similar problems without the additional volume constraint. We pick T and $\Sigma = \begin{pmatrix} \Sigma_1 & 0 \\ 0 & \Sigma_2 \end{pmatrix}$ according to the diagonal preconditioning scheme [1] (with $\alpha = 1$), note that in our case K has full rank.

5.3 Application to Volume Constrained Multi-region Segmentation

In the general multi-region case for $N \geq 2$, the discretization of the continuous problem formulation (2) is given by

$$\min_{u \in U^N \cap S \cap V} \sum_{l=1}^N \|\nabla u_l\|_g + \langle u_l, f_l \rangle \quad (15)$$

where $u = (u_l)_{l=1}^N$. The constraints are implemented by the three sets $U = \{u \in X : u_i \geq 0\}$, $V = V_1 \times \dots \times V_N$ and

$$S = \left\{ u \in X^N : \sum_{l=1}^N (u_l)_i = 1 \right\}, \text{ where } V_l = \left\{ u_l \in X : \sum_{i \in \Omega} (u_l)_i = \mathcal{V}_l \right\}$$

with $X = \mathbb{R}^{WH}$. The intersection $U \cap S$ is the simplex constraint and the set V the volume constraint. Furthermore the target volumes for the individual regions are given by $(\mathcal{V}_l)_{l=1}^N$.

Alternating Direction Method of Multipliers. We again rewrite the constraints as indicator functions and arrive at

$$\min_{u \in X^N} \underbrace{\iota_{V \cap S}(u) + \sum_{l=1}^N \|\nabla u_l\|_g}_{F(Ku)} + \underbrace{\sum_{l=1}^N \langle u_l, f_l \rangle + \iota_U(u)}_{G(u)} \quad (16)$$

where $K : X^N \rightarrow Y$, $Y := Y_1^N \times Y_2^N$ with $Ku = (\nabla u_1 \dots \nabla u_N \ u_1 \dots u_N)^T$. By applying the ADMM scheme (7) to this formulation we arrive at the following algorithm for the multi-region segmentation problem. Each step is performed for $l = 1..N$, since the minimization problem can be solved separately for each u_l and v_l .

$$\left\{ \begin{array}{l} u_l^{k+1} = \arg \min_{u_l \in X} \langle u_l, f_l \rangle + \iota_U(u_l) + \frac{\tau}{2} \|(\nabla u_l) - (v_l^k) + (\lambda_{2,l}^k)\|_2^2 \\ \begin{pmatrix} r_{1,l}^{k+1} \\ r_{2,l}^{k+1} \end{pmatrix} = \alpha \begin{pmatrix} \nabla u_l^{k+1} \\ u_l^{k+1} \end{pmatrix} + (1 - \alpha) \begin{pmatrix} v_l^k \\ w_l^k \end{pmatrix} \\ v_l^{k+1} = \arg \min_{v_l \in Y_1} \|v_l\|_g + \frac{\tau}{2} \|r_1^{k+1} - v_l + \lambda_{2,l}^k\|_2^2 \\ w^{k+1} = \arg \min_{w \in Y_2^N} \iota_{V \cap S}(w) + \frac{\tau}{2} \|r_2^{k+1} - w + \lambda_2^k\|_2^2 \\ \begin{pmatrix} \lambda_{1,l}^{k+1} \\ \lambda_{2,l}^{k+1} \end{pmatrix} = \begin{pmatrix} \lambda_{1,l}^k \\ \lambda_{2,l}^k \end{pmatrix} + \begin{pmatrix} r_{1,l}^{k+1} \\ r_{2,l}^{k+1} \end{pmatrix} - \begin{pmatrix} v_l^{k+1} \\ w_l^{k+1} \end{pmatrix} \end{array} \right. \quad (17)$$

Solving the sub-problems in u_l and v_l . These minimization problems are solved in the same way as above.

Solving the sub-problem in w . Solving the minimization problem in w amounts to finding the orthogonal projection onto the convex set $V \cap S$:

$$\begin{aligned} w^{k+1} &= \arg \min_w \iota_{V \cap S}(w) + \frac{\tau}{2} \|w - (r_2^k + \lambda_2^k)\|_2^2 \\ &= \text{proj}_{V \cap S}(r_2^k + \lambda_2^k) \end{aligned}$$

For the projection of $x = (x_l)_{l=1}^N \in X^N$ onto this convex set we find the following analytical solution

$$\text{proj}_{V \cap S}(x) = \left(x_l - \frac{1}{N} \sum_{j=1}^N x_j - \frac{1}{|\Omega|} \left(\sum_{i \in \Omega} (x_l)_i - \frac{1}{N} \sum_{j=1}^N \sum_{i \in \Omega} (x_j)_i - \mathcal{V}_l \right) \cdot \mathbf{1}_X \right)_{l=1}^N$$

which can be evaluated very efficiently using only basic arithmetic operations. This closed form projection looks somewhat daunting, but can be obtained in a straightforward manner by writing the problem in w as an unconstrained optimization problem with Lagrange multipliers.

Preconditioned Primal Dual Algorithm. Rewriting (15) as a saddle point problem with Lagrange multipliers yields

$$\min_{u \in X} \max_{(\xi, \lambda, \mu) \in Y} \underbrace{\sum_{l=1}^N \langle \nabla u_l, \xi_l \rangle + \langle \lambda_l, \mathbf{1}^T u_l \rangle + \langle \mu, \sum_{l=1}^N u_l \rangle}_{\langle Ku, (\xi \ \lambda \ \mu)^T \rangle} + \underbrace{\sum_{l=1}^N \langle u_l, f_l \rangle + \iota_U(u_l)}_{G(u)} \quad (18)$$

$$- \underbrace{(\iota_P(\xi) + \langle \lambda, \mathcal{V} \rangle + \langle \mu, \mathbf{1} \rangle)}_{F^*((\xi \ \lambda \ \mu)^T)}$$

with $P = \{\xi \in Y_1^N : \|\xi_l\|_\infty \leq g\}$, where $Y := Y_1^N \times Y_2^N \times Y_3$. Applying the primal-dual algorithm (9) yields the following update scheme:

$$\begin{cases} u_l^{k+1} = \text{proj}_U(u_l^k - T(-\text{div } \xi_k^k + \lambda_l^k + \mu^k - f)) \\ \bar{u}_l^{k+1} = u_l^{k+1} + \theta(u_l^{k+1} - u_l^k) \\ \xi_l^{k+1} = \text{proj}_P(\xi_l^k + \Sigma_1 \nabla \bar{u}_l^{k+1}) \\ \lambda_l^{k+1} = \lambda_l^k + \Sigma_2 (\sum_i (\bar{u}_l^{k+1})_i - \mathcal{V}_l) \\ \mu^{k+1} = \mu^k + \Sigma_3 (\sum_{l=1}^N (\bar{u}_l^{k+1}) - \mathbf{1}_X) \end{cases} \quad (19)$$

Again, we omit the detailed derivation of the update equations and refer to [1] to see how to apply the PD algorithm to saddle point problems. We again chose $\Sigma = \begin{pmatrix} \Sigma_1 & 0 & 0 \\ 0 & \Sigma_2 & 0 \\ 0 & 0 & \Sigma_3 \end{pmatrix}$ and T according to the diagonal preconditioning scheme with $\alpha = 1$.

6 Numerical Experiments

For evaluating the ADMM and the PD algorithm with respect to runtime for minimal partition problems under additional volume constraints we apply these algorithms to two important problems: 3D reconstruction as formulated in [6] and image segmentation based on the formulation in [5] with additional volume constraints. Both algorithms were implemented according to the update rules described above and run until the RMSE error $\|u - u^*\|_2 / \sqrt{|\Omega|}$ dropped below a threshold of $\varepsilon = 5.0 * 10^{-3}$. The optimal solution u^* of the energy minimization problem was computed by letting the algorithms run for a very long time.

We chose the ADMM parameters as $\tau = 1$, $\alpha = 1.5$ and we used 2 Jacobi iterations for the multi-region segmentation problem and 5 for the 3D reconstruction problem. For the PD algorithm we set $\theta = 1$ and chose T and Σ as described above.



Fig. 2. 3D reconstruction: Results from a single image, first row: original image, second row: reconstructed geometry, third row: textured result. The results look identical for PD and ADMM.

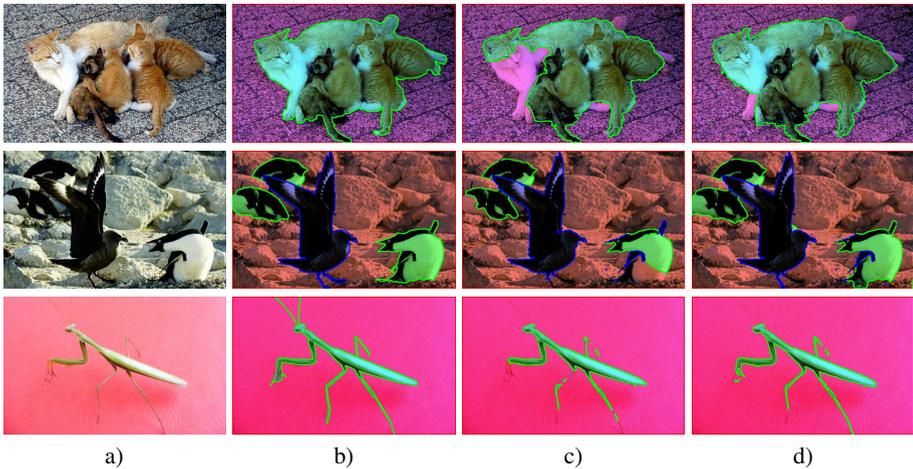


Fig. 3. Segmentation: a) original input images, b) ground truth, c) segmentation results without volume constraints and d) with volume constraints. The results look identical for PD and ADMM.

The algorithms were implemented in C++/CUDA and run on graphics processing units (GPUs). We used 32-bit floating point precision. All numerical experiments were carried out on a PC with a 3.4GHz Intel i7-3770 CPU with 32GB RAM and a NVIDIA GeForce GTX680 graphics card on a Linux distribution.

6.1 Results for 3D Reconstruction

For 3D reconstruction we used a total number of 18 test images and computed the reconstruction for each of them using PD and ADMM, see Figure 2. To examine how the algorithms scale with the resolution of the reconstruction domain Ω we used three different resolutions. A rather coarse resolution with few ($\approx 3.4 \cdot 10^4$) voxels (sm), an intermediate ($\approx 8.0 \cdot 10^5$) resolution (med) and a fine resolution with a large number ($\approx 3.6 \cdot 10^6$) of voxels (big).

For each resolution the average runtime and standard deviation over all test images is given in Table 1. Figure 4 shows an example for the different performances of PD and ADMM on the giraffe image in Figure 2. The horizontal line indicates the termination criterion where the desired accuracy of the solution is reached. Results on the other test images look similar. From the results we can conclude that the ADMM algorithm converges faster than the PD algorithm for the 3D reconstruction problem instance of the minimal partition problems with volume constraints.

Table 1. 3D reconstruction: Average and standard deviation of the runtime and number of iterations for the PD and ADMM algorithm over 18 different single view 3D Reconstruction examples. The results show that ADMM performs better.

$ \Omega $	Runtime in Seconds		Number of Iterations	
	PD	ADMM	PD	ADMM
sm	0.15 (± 0.10)	0.10 (± 0.05)	1015 (± 618)	300 (± 152)
med	13.47 (± 11.64)	8.35 (± 7.32)	5716 (± 5664)	1471 (± 1399)
big	115.30 (± 113.60)	79.16 (± 91.80)	11724 (± 12136)	3171 (± 3776)

Table 2. Segmentation: Average and standard deviation of the runtime and iteration numbers for the PD and ADMM algorithms on the Graz benchmark for interactive segmentation containing 262 test examples. We can conclude that ADMM outperforms the PD method.

Runtime in Seconds		Number of Iterations	
PD	ADMM	PD	ADMM
4.17 (± 6.41)	3.52 (± 5.95)	6843 (± 8257)	2950 (± 3822)

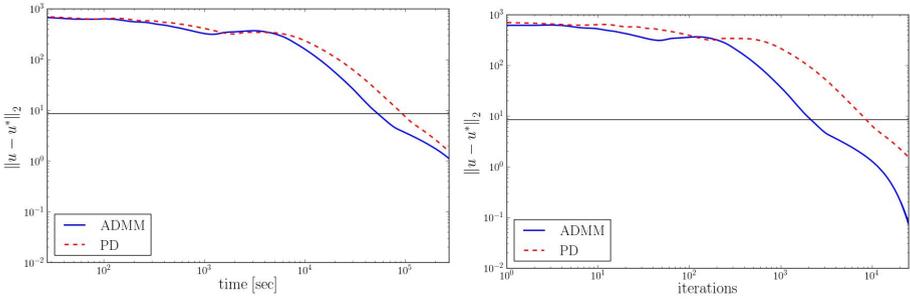


Fig. 4. 3D reconstruction: Exemplary convergence of the ADMM and the PD algorithm. The horizontal line indicates the termination criterion based on the accuracy of the algorithm (vertical axis). The ADMM algorithm converges significantly faster than the PD algorithm.

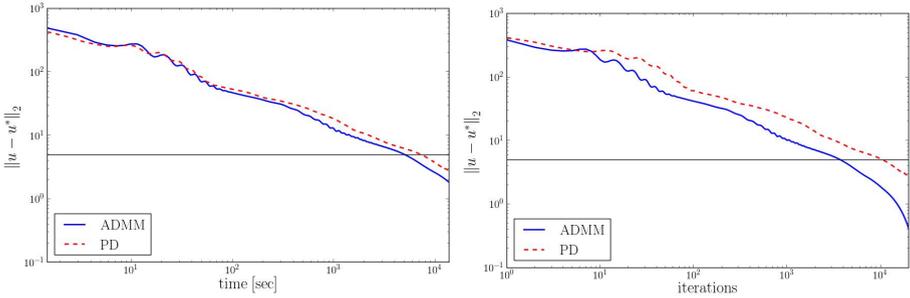


Fig. 5. Segmentation: Exemplary convergence of the ADMM and the PD algorithm for a single image from the Graz benchmark for interactive segmentation. The horizontal line indicates the termination criterion based on the accuracy of the algorithm (vertical axis). For the multi-region segmentation problem the ADMM algorithm converges slightly faster than the PD algorithm.

6.2 Results for Volume Constrained Multi-region Segmentation

For the performance evaluation of the multi-region segmentation approach we use the Graz interactive segmentation benchmark proposed by Santner et al. [4]. This benchmark contains 262 hand labeled pairs of user scribbles with ground truth. Since this benchmark is only used for evaluating the performance of the algorithm the volume constraints are computed from the ground truth segmentations for each image. The data terms f_i in (2) are computed based on [5], but are not part of the performance evaluation.

Figure 5 exemplarily shows the performance in terms of runtime and number of iterations for the first image of the Graz database, which contains four labels. The other performance plots look similar. On average we obtain the results in Table 2 over all images of the benchmark. From the results we can conclude that ADMM is also faster for the multi-region segmentation problem as an instance of the minimal partition problems with volume constraint. Figure 3 shows some qualitative segmentation results based on an optimized volume constraint.

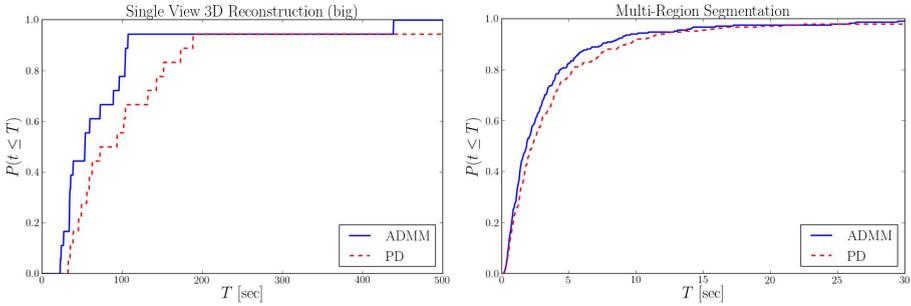


Fig. 6. Cumulative distribution functions over the runtimes for the 18 3D reconstruction examples and the 262 segmentation examples. The plots indicate for each runtime T in seconds (horizontal axis) the ratio of examples with runtimes below T seconds $P(t \leq T)$ (vertical axis). The faster the curve grows the more efficient is the algorithm. For both instances of partition problems the ADMM algorithm yields better performance results.

6.3 Cumulative Distribution Function of Runtimes

Finally, we show the cumulative distribution functions (CDF) of the runtimes for both partition problems instances, i.e. 3D reconstruction and multi-region segmentation, in Figure 6. The vertical axis shows the ratio of samples with runtimes below T seconds on the horizontal axis ($P(t \leq T)$). The 3D reconstruction results are based on 18 sample images, which causes the staircase pattern in the plot. For the segmentation results we had 262 sample images. The CDF plots confirm that the ADMM algorithm in general converges faster than the PD algorithm.

7 Conclusion

We have compared two algorithms for solving the minimal partition problem with additional volume constraints, PD and ADMM. In the PD algorithm the constraints are handled by Lagrange multipliers, and every iteration step has a closed form solution. Contrary, the ADMM algorithm requires an approximate solution of a least squares problem in every iteration, and the constraints are handled by orthogonal projections onto the corresponding sets. The results suggest that both algorithms can be extended in a straightforward manner in order to handle additional convex constraints.

We conducted several experiments on 3D reconstruction examples and on the Graz interactive segmentation benchmark. Our experiments indicate that the ADMM approach provides comparable and often better performance. On average, it yields shorter runtimes than the PD algorithm and requires significantly less iterations, while the quality of the results is identical.

References

1. Pock, T., Chambolle, A.: Diagonal preconditioning for first order primal-dual algorithms in convex optimization. In: Int. Conf. Comp. Vis. (2011)
2. Eckstein, J., Bertsekas, D.P.: On the Douglas-Rachford Splitting Method and the Proximal Point Algorithm for Maximal Monotone Operators. J. Math. Program. 55, 293–318 (1992)

3. Unger, M., Pock, T., Cremers, D., Bischof, H.: TVSeg - Interactive Total Variation Based Image Segmentation. In: Brit. Mach. Vis. Conf. (2008)
4. Santner, J., Pock, T., Bischof, H.: Interactive Multi-Label Segmentation. In: Kimmel, R., Klette, R., Sugimoto, A. (eds.) ACCV 2010, Part I. LNCS, vol. 6492, pp. 397–410. Springer, Heidelberg (2011)
5. Nieuwenhuis, C., Cremers, D.: Spatially Varying Color Distributions for Interactive Multi-Label Segmentation. IEEE Trans. Pattern Anal. Mach. Intell. (2012)
6. Töppe, E., Oswald, M.R., Cremers, D., Rother, C.: Image-based 3D Modeling via Cheeger Sets. In: Kimmel, R., Klette, R., Sugimoto, A. (eds.) ACCV 2010, Part I. LNCS, vol. 6492, pp. 53–64. Springer, Heidelberg (2011)
7. Kolev, K., Pock, T., Cremers, D.: Anisotropic Minimal Surfaces Integrating Photoconsistency and Normal Information for Multiview Stereo. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part III. LNCS, vol. 6313, pp. 538–551. Springer, Heidelberg (2010)
8. Zach, C., Gallup, D., Frahm, J.M., Niethammer, M.: Fast Global Labeling for Real-Time Stereo Using Multiple Plane Sweeps. In: Vision, Modeling and Visualization Workshop (2008)
9. Lellmann, J., Schnörr, C.: Continuous Multiclass Labeling Approaches and Algorithms. J. Imaging Sci. 4, 1049–1096 (2010)
10. Chambolle, A., Pock, T.: A First-Order Primal-Dual Algorithm for Convex Problems with Applications to Imaging. J. Math. Imaging Vis. 40, 120–145 (2011)
11. Goldstein, T., Bresson, X., Osher, S.: Geometric Applications of the Split Bregman Method: Segmentation and Surface Reconstruction. Technical report, UCLA (2009)
12. Paul, G., Cardinale, J., Sbalzarini, I.: An Alternating Split Bregman Algorithm for Multi-Region Segmentation. In: IEEE Asilomar Conf. Signals, Systems, and Computers, pp. 426–430 (2011)
13. Häuser, S., Steidl, G.: Convex Multiclass Segmentation with Shearlet Regularization. Int. J. Comp. Math. 90, 62–81 (2013)
14. Niethammer, M., Zach, C.: Segmentation with area constraints. Medical Image Analysis 17, 101–112 (2013)
15. Chambolle, A., Cremers, D., Pock, T.: A Convex Approach for Computing Minimal Partitions. Technical report, University of Bonn (2008)
16. Pock, T., Cremers, D., Bischof, H., Chambolle, A.: An Algorithm for Minimizing the Piecewise Smooth Mumford-Shah Functional. In: Int. Conf. Comp. Vis. (2009)
17. Chan, T., Esedoğlu, S., Nikolova, M.: Algorithms for Finding Global Minimizers of Image Segmentation and Denoising Models. J. Appl. Math. 66, 1632–1648 (2006)
18. Boyle, J.P., Dykstra, R.L.: A method for finding projections onto the intersection of convex sets in Hilbert spaces. Lecture Notes in Statistics, vol. 37, pp. 28–47
19. Beck, A., Teboulle, M.: A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems. J. Img. Sci. 2, 183–202 (2009)
20. Boyd, S., Parikh, N., Chu, E., Peleato, B., Eckstein, J.: Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers. J. Found. Trends Mach. Learn. 3, 1–122 (2011)
21. Setzer, S.: Split Bregman Algorithm, Douglas-Rachford Splitting and Frame Shrinkage. In: Tai, X.-C., Mørken, K., Lysaker, M., Lie, K.-A. (eds.) SSVM 2009. LNCS, vol. 5567, pp. 464–476. Springer, Heidelberg (2009)