

SPARSE TEMPLATE-BASED VARIATIONAL IMAGE SEGMENTATION

DIRK BREITENREICHER, JAN LELLMANN
and CHRISTOPH SCHNÖRR

*Image and Pattern Analysis Group,
Heidelberg Collaboratory of Image Processing,
Department of Mathematics and Computer Science,
University of Heidelberg, Heidelberg, Germany*

We introduce a variational approach to image segmentation based on sparse coverings of image domains by shape templates. The objective function combines a data term that achieves robustness by tolerating overlapping templates with a regularizer enforcing sparsity. A suitable convex relaxation leads to the variational approach that is amenable to large-scale convex programming.

Our approach takes implicitly into account prior knowledge about the shape of objects and their parts, without resorting to combinatorially difficult problems of variational inference. We illustrate our approach by numerical examples and indicate how prior knowledge acquisition may be achieved by learning from examples.

Keywords: Image segmentation; Nesterov method; fast primal-dual method.

1. Introduction

Approaches to image segmentation require additional information, besides given image data, to be able to decompose an image into meaningful parts. Such information is typically provided by a model of object appearance that suitably constrains the space of image segmentations. The unknown subset of the image domain covered by one or several objects that is to be segmented, is called *foreground*, and its complement is called *background*. Models of object appearance represent both the foreground *region* in terms of features (color, texture) that are characteristic for an object class of interest, and the delimiting *contour* separating fore- and background, i.e., object *shape*. In this paper, we study a variational approach to image segmentation that implicitly takes into account shape prior knowledge in terms of a dictionary of shape templates.

Variational approaches to image segmentation that utilize shape prior knowledge include statistical models of parameterized contours [Cremers *et al.* (2002, 2003)], level-set based segmentation [Cremers *et al.* (2006, 2007); Chan and Zhu (2005)], and discrete combinatorial approaches in terms of Markov random field

(MRF) models [Kohli *et al.* (2008); Bergtholdt *et al.* (2010)]. A common property of these approaches is the inherent *nonconvexity* introduced by the respective shape prior model. Therefore, and in view of recent *convex* formulations of the basic fore-/background separation problem [Chan *et al.* (2006)] and its extension to the nonbinary case [Lellmann *et al.* (2009); Chambolle *et al.* (2008); Lellmann and Schnörr (2010)], we focus in this paper on a *convex* variational approach to foreground/background separation based on shape prior knowledge.

Two major aspects distinguish our approach from prior work. First, rather than representing partitions of the image domain pixelwise by a corresponding indicator vector, we aim at separating fore- and background by covering the foreground with a collection of shape templates. While this seems to amount to replace standard binary basis functions by another set of basis functions, this is not quite true because shape templates may overlap and their addition does not yield an indicator function. Second, shape regularization by the common length penalty term [Mumford and Shah (1989)] through a total variation prior in the convex formulation [Chan *et al.* (2006)] is replaced by enforcing sparsity instead, i.e., by penalizing the number of shape templates required to “explain” given image data — see Fig. 1.

Our work is also motivated by the basis pursuit framework [Chen *et al.* (2001)] and the striking performance of “ ℓ_1 -decoding” by convex programming in underdetermined compressed sensing scenarios [Candes *et al.* (2005); Cohen *et al.* (2009)]. The need for (approximate) invariance of image segmentation and object recognition necessitates to cope with underdetermined settings, and although our dictionaries do not satisfy corresponding strong mathematical assumptions like the restricted isometry property, our numerical experiments reveal promising performance and a significant potential for real-world applications.

Related work in the field of computer vision includes the work of Borenstein and Ullman [Borenstein and Ullman (2002, 2008)] on segmentation using image fragments. Unlike shape templates, image fragments model not only shape but

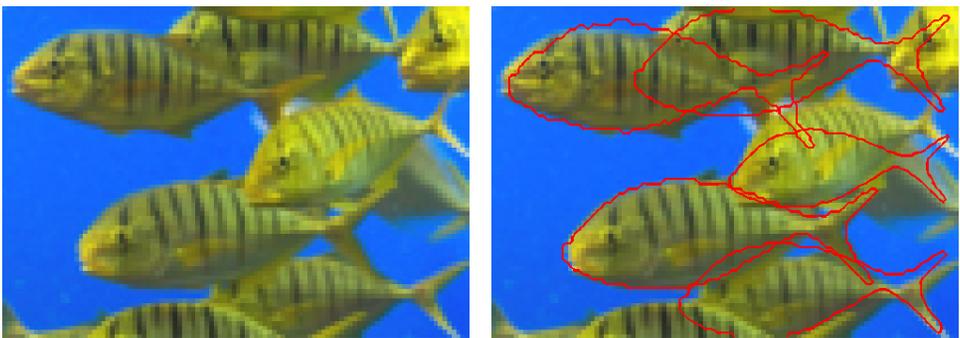


Fig. 1. Separating fishes from the background and from each other by convex optimization in terms of a sparse covering of the image by shape templates. The dictionary of shape templates was generated from a single template by translation, rotation, and scaling. The approach presented in this work copes with a significant amount of overlapping templates and occlusion.

also the image intensity function and image features for a particular object class of interest, like “horse.” Accordingly, the focus of this work is on corresponding image features, whereas variational inference is based on an MRF model, and simplifications are made to keep it computationally tractable. By contrast, in the present paper, we focus on the variational model from the optimization point of view and largely ignore the issue of feature extraction, keeping in mind that recent convex variational formulations [Lellmann *et al.* (2009); Chambolle *et al.* (2008); Lellmann and Schnörr (2010)] do not impose any restrictions on the choice of image features.

Finally, we point out further recent work [Aharon *et al.* (2006); Yu *et al.* (2010)] on dictionary based image processing. Although the scope of this work is confined to image denoising without semantical interpretation like image segmentation and object recognition, recent extensions toward *learning* of task-specific dictionaries [Rubinstein *et al.*, to appear] exhibit a general and highly relevant research direction. We present a few numerical experiments indicating that adopting and properly modifying this idea for shape template learning is promising indeed.

Organization.

Our paper is structured as follows. We present our variational approach in Sec. 2. The primary issue here is to deal with overlapping templates and object occlusion. In Sec. 3, we derive a sufficiently tight convex relaxation that is amenable to large-scale optimization. Two such approaches are examined in Sec. 4 with respect to robustness and speed of convergence to a global optimum. We empirically evaluate and validate our approach in Sec. 5 with a range of numerical examples and briefly address the issue of knowledge acquisition by learning from examples.

Notation.

For the reader’s convenience, we briefly summarize the notations used in this paper. Matrices and vectors are denoted by upper-case and lower-case letters, respectively. We refer to the k th row of a matrix A by A_k . We denote by e the vector consisting of ones, $e = (1, \dots, 1)^\top$. The Hadamard product between two equally sized vectors or matrices A, B is given by $A \odot B$. Additionally, we define $A \odot x := A \odot (ex^\top)$ and $x \odot A := (xe^\top) \odot A$. For a matrix $A \in \mathbb{R}^{m \times n}$ (possibly a vector if $m = 1$), $\text{vecmax}(A)$ denotes the vector $v \in \mathbb{R}^m$ such that $v_k = \max\{A_{k,1}, \dots, A_{k,n}\}$. Finally, \mathcal{C} and \mathcal{D} denote two closed convex sets, where the corresponding (Euclidean) orthogonal projections are given by $\Pi_{\mathcal{C}}, \Pi_{\mathcal{D}}$.

2. Sparse Template-Based Shape Representation

In the sparse representation framework, images (or generally signals) $f \in \mathbb{R}^m$ are assumed to be additively composed of a small number of basis functions drawn

from an overcomplete basis of n functions, $A = (a_1, \dots, a_n) \in \mathbb{R}^{m \times n}$, i.e.

$$f = Ax, \quad x \in \mathbb{R}^n, \quad (1)$$

and x is assumed to be sparse in the sense that it contains only a few nonzero entries. Finding the set of basis functions for a given image is then achieved by solving the minimum-norm solution optimization problem

$$\min_{x \in \mathbb{R}^n} \|x\|_0 \quad \text{subject to } Ax = f, \quad (2)$$

where $\|x\|_0$ refers to the ℓ_0 pseudonorm, i.e., the number of non-zero entries of x .

While solving Eq. (2) directly is a difficult combinatorial problem unless A comprises an orthogonal basis, it can be shown that under some circumstances one may replace $\|\cdot\|_0$ by $\|\cdot\|_1$ [Candes *et al.* (2005)]. Moreover, to account for noise the constraint is in practice commonly enforced approximately by a penalty term, resulting in the well-known problem

$$\min_{x \in \mathbb{R}^n} \{\mu \|x\|_1 + \|Ax - f\|_2^2\}. \quad (3)$$

In the following, we examine how this approach can be extended to sparse shape representation. The basic idea is to transfer the methods from the image domain to the shape domain via the indicator function representation. Therefore we assume that the basis functions are indicator functions of some prototypical shapes, i.e., $a_i \in \{0, 1\}^m$, and $f \in \{0, 1\}^m$ is the characteristic function of a shape that is the union of several of the basis shapes.

From these definitions, it becomes clear that using Eq. (3) to recover the basis shapes is bound to fail: current sparse representation methods are based on the assumption that the basis functions overlay in an additive fashion. In contrast, in the shape context, the basis functions are in a sense “opaque,” as the “union of basis shapes” principle dictates that they do not add up in regions of intersection, but rather stay at 1. This desired behavior is illustrated in Fig. 2.

We, therefore, replace the additivity assumption (1) by the concept

$$f_k = \max\{A_{k,1}x_1, \dots, A_{k,n}x_n\}, \quad x \in \{0, 1\}^n, \quad (4)$$

where the maximum has to be taken on each component separately. Thus

$$f = \text{vecmax}(A \odot x), \quad (5)$$

with “vecmax” extended to apply to each row of $A \odot x$ separately.

The overall problem then reads

$$\min_{x \in \{0,1\}^n} \{\mu \|x\|_1 + \|\text{vecmax}(A \odot x) - f\|_2^2\}. \quad (6)$$

Unfortunately, even after relaxing to $x \in [0, 1]^n$, this problem is no longer convex. However, it will turn out below that by switching to an appropriate convex relaxation, good solutions can be obtained by finding the global optimum of a convex problem.

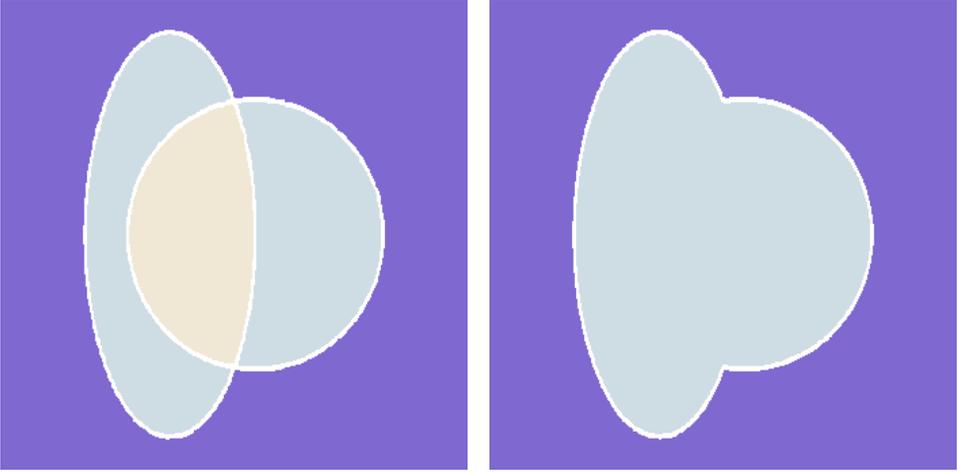


Fig. 2. Illustration of the “union of basis shapes” principle. Left: ordinary addition of basis functions yields inappropriate representation of the regions of intersection. Right: desired behavior of overlapping shapes. The goal of this work is to approximately achieve this property by a convex optimization approach.

3. Convex Envelope Local Relaxation

First note that we are dealing exclusively with indicator functions. Hence, $f \in \{0, 1\}^m$ and

$$\|\text{vecmax}(A \odot x) - f\|_2^2 = \sum_k (\text{vecmax}(A_{k,\cdot} \odot x) - f_k)^2 \quad (7)$$

$$= \sum_{k, f_k=1} (1 - \text{vecmax}(A_{k,\cdot} \odot x)) + \sum_{k, f_k=0} \text{vecmax}(A_{k,\cdot} \odot x) \quad (8)$$

$$= \sum_k \left\{ f_k \underbrace{(1 - \text{vecmax}(A_{k,\cdot} \odot x))}_{=:g(A_{k,\cdot} \odot x)} + (1 - f_k) \underbrace{\text{vecmax}(A_{k,\cdot} \odot x)}_{=:h(A_{k,\cdot} \odot x)} \right\}. \quad (9)$$

Still, as we are dealing exclusively with indicator functions, g and h are defined solely on binary vectors in $\{0, 1\}^n$ (cf. the black dots in Fig. 3 for the two-component case). The optimal convex extension of Eq. (9) to $[0, 1]^n$ is given by its convex envelope, that is the largest closed convex function majorized by f . The maximality of the convex envelope ensures that a minimal amount of artificial nonbinary solutions are introduced into the problem. However, finding the global convex envelope is generally a hard problem. Instead, we construct an approximation by constructing the envelope over the individual terms involving g and h .

To do this, we note that for any (convex or nonconvex) function f , the (Legendre–Fenchel) biconjugate f^{**} is a convex function and its epigraph is the convex closure (i.e., the closure of the convex hull) of the epigraph of f , $\text{epi } f^{**} = \text{cl}(\text{epi } f)$. Moreover, f^{**} is the convex envelope of f . Thus it remains to

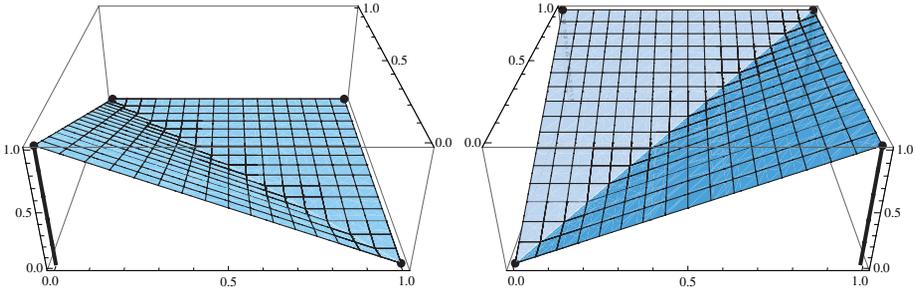


Fig. 3. Values of the distance function (black dots) depicting the desired behavior for nonzero and zero signal entries. Using convex envelopes for both foreground and background, proper relaxations can be obtained. Left: relaxation for g , i.e., at locations where $f_k \neq 0$ (foreground). Right: relaxation for h , i.e., at locations where $f_k = 0$ (background).

compute g^{**} and h^{**} in order to approximate the convex envelope of the objective, see Eq. (9).

Proposition 3.1. *Let*

$$g(p) = \begin{cases} 1, & p = 0, \\ 0, & p \in \{0, 1\}^n, \quad p \neq 0, \\ +\infty, & p \notin \{0, 1\}^n, \end{cases} \quad h(p) = \begin{cases} 0, & p = 0, \\ 1, & p \in \{0, 1\}^n, \quad p \neq 0, \\ +\infty, & p \notin \{0, 1\}^n. \end{cases} \quad (10)$$

Then the convex envelopes are given by

$$g^{**}(p) = \begin{cases} \max\{0, 1 - e^\top p\}, & p \in [0, 1]^n, \\ +\infty, & \text{otherwise,} \end{cases} \quad h^{**}(p) = \begin{cases} \text{vecmax}(p), & p \in [0, 1]^n, \\ +\infty, & \text{otherwise.} \end{cases} \quad (11)$$

Proof. See Appendix. □

The full local relaxation of problem (6) is thus

$$\min_{x \in \{0,1\}^n} \left\{ \mu \|x\|_1 + \sum_k (f_k g^{**}(A_{k,\cdot} \odot x) + (1 - f_k) h^{**}(A_{k,\cdot} \odot x)) \right\}. \quad (12)$$

The function h^{**} is considerably more difficult to handle numerically than g^{**} , as it requires to introduce a large amount of KKT multipliers resp. dual variables. However, experiments showed that very good results can be obtained by replacing $h^{**}(p)$ with the upper bound $e^\top p$, yielding our final relaxation

$$\min_{x \in \{0,1\}^n} \left\{ \mu \|x\|_1 + \sum_k (f_k \max\{0, 1 - \langle A_{k,\cdot}, x \rangle\} + (1 - f_k) \langle A_{k,\cdot}, x \rangle) \right\}. \quad (13)$$

Problem (13) now allows to compute good approximate solutions to the combinatorial problem (6). However, it remains to cope with the large problem size and nonsmoothness of the objective.

3.1. Saddle-point formulation

In order to handle the nonsmoothness of Eq. (13), we turn to considering the optimization problem as bilinear saddle-point problem of the form

$$\min_{x \in \mathcal{C}} \max_{v \in \mathcal{D}} \{ \langle d, x \rangle + \langle v, Lx \rangle - \langle b, v \rangle \}. \quad (14)$$

By explicitly introducing the dual variables v , this allows to naturally handle non-smooth objectives as commonly done in image denoising and segmentation [Zach *et al.* (2008); Pock *et al.* (2009); Lellmann *et al.* (2009)].

Using the notation introduced above, we can rewrite the “max” term in Eq. (13) as $\max_{v_k \in [0,1]} (v_k(1 - A_k, \cdot))$. Thus, the optimization problem (13) can be represented according to Eq. (14) by setting $\mathcal{C} = [0, 1]^n$, $\mathcal{D} = [0, 1]^m$, $d = A^\top(1 - f) + \mu e$, $b = -f$, and $L = -f \odot A$, respectively.

The primal and dual objectives are consequently given by

$$f_p(x) := \max_{v \in \mathcal{D}} g(x, v) \quad \text{and} \quad f_d(v) := \min_{x \in \mathcal{C}} g(x, v), \quad (15)$$

respectively. The dual problem then consists of maximizing $f_d(v)$ over \mathcal{D} . As \mathcal{C} and \mathcal{D} are bounded, it follows from [Rockafellar (1970), Corollary 37.6.2] that a saddle point (x^*, v^*) of g exists. With [Rockafellar (1970), Lemma 36.2], this implies strong duality, i.e.

$$\min_{x \in \mathcal{C}} f_p(x) = f_p(x^*) = g(x^*, v^*) = f_d(v^*) = \max_{v \in \mathcal{D}} f_d(v). \quad (16)$$

In our case, \mathcal{C}, \mathcal{D} encode simple box constraints, which allows to compute f_d as well as the orthogonal projections $\Pi_{\mathcal{C}}$ and $\Pi_{\mathcal{D}}$ efficiently, a fact that will prove important in the algorithmic part.

4. Optimization

To optimize the saddle point formulation (14), we have to take into account the large-scale nature and the inherent nonsmoothness of the objective. While interior-point solvers are known to be very fast for small- to medium-sized problems, they are not particularly well suited for massively parallel computation, such as on the upcoming GPU platforms, due to the expensive inner Newton iterations.

Although $L \in \mathbb{R}^{m \times n}$ with $n \gg m$ is sparse in general, columns are usually nonorthogonal, i.e., the sets of the nonzero entries’ indices overlap significantly. Thus, $L^\top L$ might be dense such that Newton steps are potentially very expensive with respect to space and time.

We instead focus on first-order methods involving only evaluations of L and L^\top and projections on \mathcal{C} and \mathcal{D} , as these operations can be highly parallelized due to their local nature. To this end, we investigate a fast primal-dual method based on [Pock *et al.* (2009)] as well as Nesterov’s multi step method [Nesterov (2004)].

Algorithm 1 Fast primal-dual method

- 1: Choose $x^{(0)} \in \mathcal{C}, v^{(0)} \in \mathcal{D}$.
 - 2: Choose $\tau_p > 0, \tau_d > 0, N \in \mathbb{N}$.
 - 3: $y^{(0)} \leftarrow x^{(0)}$
 - 4: **for** $k = 0, \dots, N - 1$ **do**
 - 5: $v^{(k+1)} \leftarrow \Pi_{\mathcal{D}}(v^{(k)} + \tau_d(Lx^{(k)} - b))$.
 - 6: $y^{(k+1)} \leftarrow \Pi_{\mathcal{C}}(y^{(k)} - \tau_p(L^\top v^{(k+1)} + d))$.
 - 7: $x^{(k+1)} \leftarrow 2y^{(k+1)} - y^{(k)}$.
 - 8: **end for**
-

4.1. Fast primal-dual method

One of the most straightforward approaches for optimizing Eq. (14) is to fix small primal and dual step sizes τ_p resp. τ_d , and alternately apply projected gradient descent resp. ascent on the primal resp. dual variables, i.e., an Arrow–Hurwicz approach [Arrow *et al.* (1964)]. However, it seems nontrivial to derive sufficient conditions for convergence. Because of this, in [Pock *et al.* (2009)] the authors propose the fast primal-dual (FPD) method, a variant of the Popov’s saddle point method [Popov (1980)]. The algorithm is summarized in Algorithm 1.

Due to the explicit steps involved, there is an upper-bound condition on the step size to assure convergence, which can be shown to be $\tau_p \tau_d < 1/\|L\|^2$ [Pock *et al.* (2009)]. As in our case, L is explicitly given and evaluations of L and L^\top can be computed efficiently, we use the power iteration method to extract the maximal eigenvalue of $L^\top L$. Additionally, as both the primal and dual iterates are always feasible, a stopping criterion based on the primal-dual gap as outlined in Sec. 3.1 can be employed.

4.2. Nesterov method

Next, we provide a short summary of the application of Nesterov’s multi-step method [Nesterov (2004)] to the saddle-point problem (14) as proposed in the context of image labeling in [Lellmann *et al.* (2009)]. In contrast to the FPD method, the nonsmoothness is treated by first applying a smoothing step and then using a smooth constrained optimization method. The amount of smoothing is balanced in such a way that the overall number of iterations to produce a solution with a specific accuracy is minimized.

The algorithm has a theoretical worst-case complexity of $O(1/\varepsilon)$ for finding an ε -optimal solution, and has been shown to give accurate results for denoising [Aujol (2008)] and general ℓ_1 -norm based problems [Becker *et al.* (2009)]. Besides the desired accuracy, no other parameters have to be provided. The complete algorithm for our saddle-point formulation is shown in Algorithm 2.

Algorithm 2 Nesterov's multi-step method

- 1: Let $c_1 \in \mathcal{C}$, $c_2 \in \mathcal{D}$ and $r_1, r_2 \in \mathbb{R}$ s.t. $\mathcal{C} \subseteq \mathcal{B}_{r_1}(c_1)$ and $\mathcal{D} \subseteq \mathcal{B}_{r_2}(c_2)$.
 - 2: Choose $x^{(0)} \in \mathcal{C}$ and $N \in \mathbb{N}$.
 - 3: Let $\mu \leftarrow \frac{2\|L\|}{N+1} \frac{r_1}{r_2}$.
 - 4: Set $g^{(-1)} = 0, w^{(-1)} = 0, y^{(0)} = x^{(0)}$.
 - 5: **for** $k = 0, \dots, N$ **do**
 - 6: $v \leftarrow \Pi_{\mathcal{D}} \left(c_2 + \frac{1}{\mu} (Ly^{(k)} - b) \right)$.
 - 7: $w^{(k)} \leftarrow w^{(k-1)} + (k+1)v$.
 - 8: $v^{(k)} \leftarrow \frac{2}{(k+1)(k+2)} w^{(k)}$.
 - 9: $g \leftarrow d + L^\top v$.
 - 10: $g^{(k)} \leftarrow g^{(k-1)} + \frac{k+1}{2} g$.
 - 11: $x^{(k)} \leftarrow \Pi_{\mathcal{C}} \left(y^{(k)} - \frac{\mu}{\|L\|^2} g \right)$.
 - 12: $z^{(k)} \leftarrow \Pi_{\mathcal{C}} \left(c_1 - \frac{\mu}{\|L\|^2} g^{(k)} \right)$.
 - 13: $y^{(k+1)} \leftarrow \frac{2}{k+3} z^{(k)} + \left(1 - \frac{2}{k+3} \right) x^{(k)}$.
 - 14: **end for**
-

The only expensive operations are the projections $\Pi_{\mathcal{C}}$ and $\Pi_{\mathcal{D}}$, which are efficiently computable in our scenario. The algorithm converges in any case and provides explicit suboptimality bounds.

Proposition 4.1. *In Algorithm 2, iterates $x^{(k)}, v^{(k)}$ are primal resp. dual feasible, i.e., $x^{(k)} \in \mathcal{C}$ and $v^{(k)} \in \mathcal{D}$. Moreover, for any solution x^* of the relaxed problem (14), the relation*

$$f_p(x^{(N)}) - f_p(x^*) \leq f_p(x^{(N)}) - f_d(v^{(N)}) \leq \frac{2r_1 r_2 \|L\|}{(N+1)} \quad (17)$$

holds for the final iterates $x^{(N)}, v^{(N)}$, where N denotes the total number of iterations, and c_1, c_2, r_1, r_2 refer the centers and radii of balls enclosing \mathcal{C} resp. \mathcal{D} , i.e., $\mathcal{C} \subseteq \mathcal{B}_{r_1}(c_1)$ and $\mathcal{D} \subseteq \mathcal{B}_{r_2}(c_2)$.

Proof. Apply [Nesterov (2004), Theorem 3] with $\hat{f}(x) = \langle x, d \rangle$, $A = L$, $\hat{\phi}(v) = \langle b, v \rangle$, $d_1(x) := \frac{1}{2} \|x - c_1\|^2$, $d_2(v) := \frac{1}{2} \|v - c_2\|^2$, $D_1 = \frac{1}{2} r_1^2$, $D_2 = \frac{1}{2} r_2^2$, $\sigma_1 = \sigma_2 = 1$, $M = 0$. \square

Corollary 4.2. *For given $\varepsilon > 0$, applying Algorithm 2 with*

$$N = \left\lceil \frac{2r_1 r_2 \|L\|}{\varepsilon} - 1 \right\rceil \quad (18)$$

yields an ε -optimal solution of Eq. (14), i.e., $f_p(x^{(N)}) - f_p(x^) \leq \varepsilon$.*

5. Numerical Evaluation

To empirically evaluate the proposed formulation along with the presented optimization algorithms, we implemented the FPD algorithm [Pock *et al.* (2009)] as well as Nesterov’s multi-step method [Nesterov (2004)] using MATLAB research code, along with optimized C coding for the evaluation of the matrix products, and applied the framework to different applications.

5.1. Template-based segmentation

5.1.1. Run-time comparison

To compare the performance of the algorithms presented above, we considered the academic example visualized in Fig. 4. The image consists of a centered circle that has considerable overlap with four equally sized squares. We supplemented the image with noise to simulate real-world scenarios as well as imperfect features and local classification. As basis dictionary we used squares and circle templates shifted to all possible image locations.

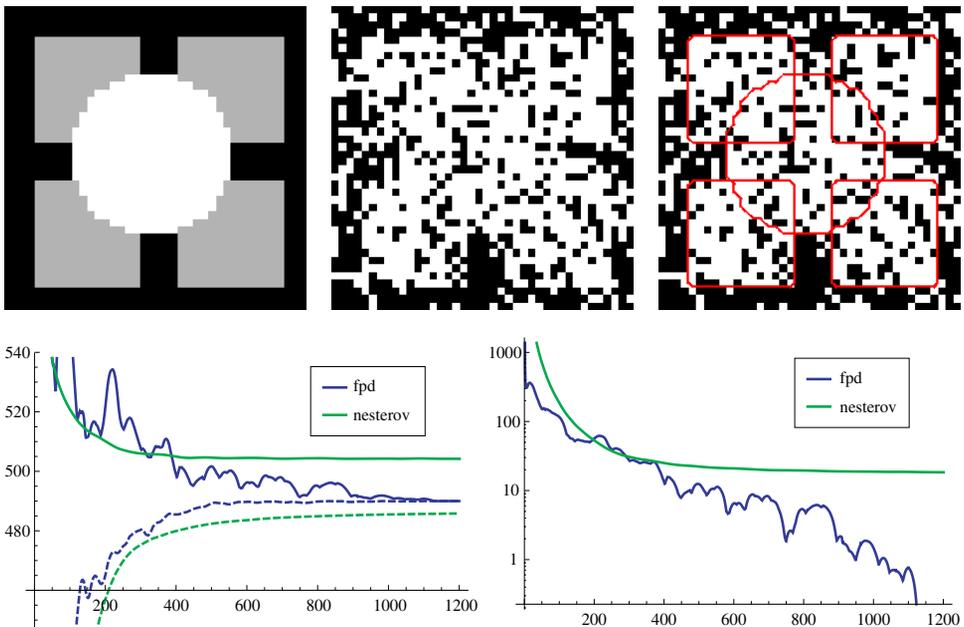


Fig. 4. Application of the proposed approach to an academic toy example. While both FPD and Nesterov’s approach converge to the global optimum, the former requires typically far less iterations. Top left: original image consisting of a circle and four squares. Top center: local segmentation based on foreground probabilities with overlaid noise. This pre-segmentation is used as an input to the proposed method. Top right: result obtained by our algorithm. In this case, the output vector has exactly five nonzero entries corresponding to the five visible templates. Bottom left: primal (solid) and dual (dashed) function value for both approaches in each iteration. Bottom right: corresponding primal-dual gap vs. number of iterations.

Running the FPD algorithm and Nesterov’s multi-step method reveals that the former typically provides faster convergence to the global optimizer. A direct comparison of both approaches is given in the lower panel of Fig. 4.

Although there is no explicit termination criteria for FPD compared to the worst-case bound of Nesterov’s approach, in the following we use FPD to infer the optimal template configuration. The primal-dual gap is used as corresponding termination criteria.

We think that both algorithms can be further tuned to provide faster convergence. On the one hand, Nesterov’s procedure uses only worst-case parameters. Additional knowledge of the expected sparsity of the result would enable to provide tighter bounds on the parameters involved and thus speed up the entire approach. Considering the FPD method, on the other hand, one possibility to provide faster convergence is to find a more appropriate, possibly asymmetric choice for the primal and dual step sizes.

5.1.2. Real world images

Next, we demonstrate the applicability of the proposed approach to real images. To this end, we used the color images presented in Figs. 1, 5, and 6.

For the color image, we extracted the pre-segmentation by computing local features from histograms over regions pre-selected by the user (Fig. 1), or by inspecting the distance to the characteristic background (Fig. 5) resp. foreground color (Fig. 6), and a simple local thresholding operation. These initial segmentations are depicted in Figs. 5 and 6.

The regularization parameter λ is set by hand and varies between the different experiments. It roughly reflects the minimal amount of pixels that have to be exclusively covered by a certain basis function in order to justify its presence. The remaining parameters τ_p, τ_d are set equally such that they fulfill the convergence property $\tau_p \tau_d \leq \frac{1}{\|L\|^2}$.



Fig. 5. Application of the proposed approach to an image containing overlapping coins. Although not all objects are detected our approach reveals promising results and recognizes even highly occluded models. Left: input image. Center: local pre-segmentation obtained by inspecting the distance to the background color and local thresholding; used as input to the proposed method. Right: segmented result.



Fig. 6. Application of the proposed template-based segmentation approach to a real-world image. Left to right: original image, local pre-segmentation using a thresholded distance to the color red as foreground indicator, shape templates used for segmentation, and the final result. Note that even highly overlapping parts are labeled correctly.

For the templates we used pre-defined image patches shifted to all possible image locations. A specialized implementation of the corresponding matrix L enables us to perform matrix vector multiplications efficiently without computing L as a whole. This dramatically speeds up the entire approach without using any additional storage as the corresponding basis functions can be computed on-the-fly.

5.2. *Shape decomposition*

We conducted two more experiments to indicate the potential of our template-based representation of image segmentations for further processing.

Figure 7 shows the segmentation of the horse image obtained with circular templates of different sizes. The corresponding covering provides information in terms of the corresponding coefficients about the localization of the torso and the limbs.

In a related experiment we used pre-defined parts of a horse such as its head, the torso, and the legs, in order to recover them in a given image (Fig. 8). Under moderate variation of the underlying shape or the observers' viewpoint, the proposed approach can robustly identify the templates in different images and thus provides useful input data for further template adjustment or contextual processing steps.

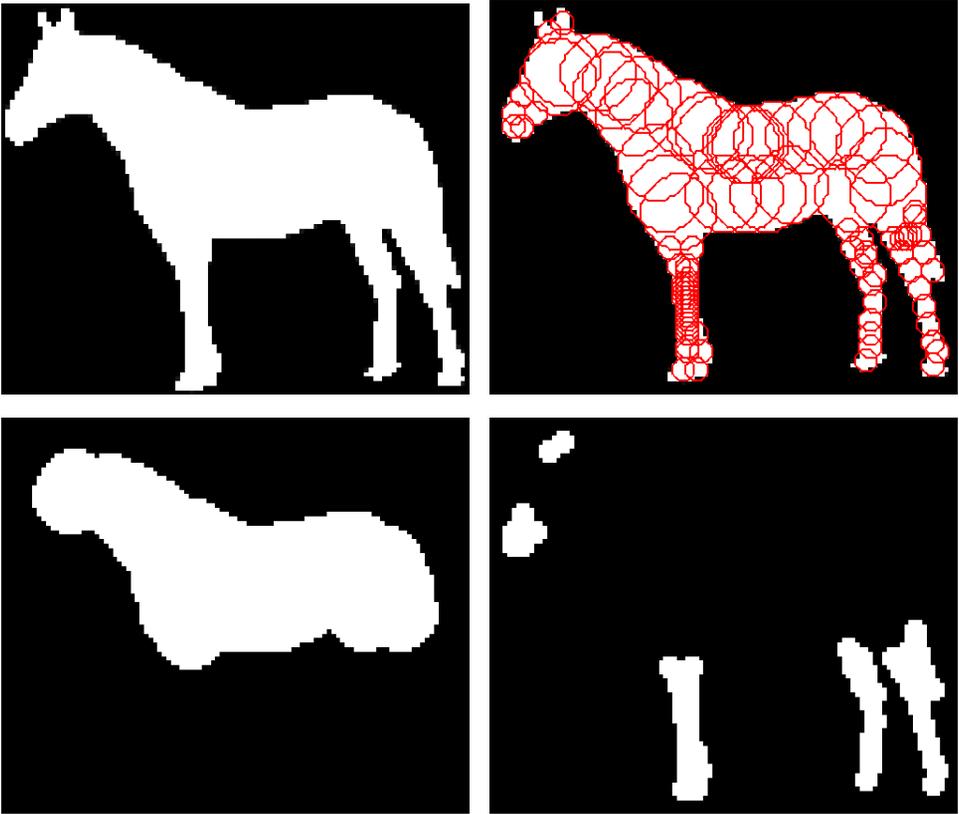


Fig. 7. Shape decomposition by covering with templates of different sizes. Top row: input image and segmented result. Bottom row: Decomposition into torso and limbs based on the nonvanishing variables x_i corresponding to large and small templates, respectively.

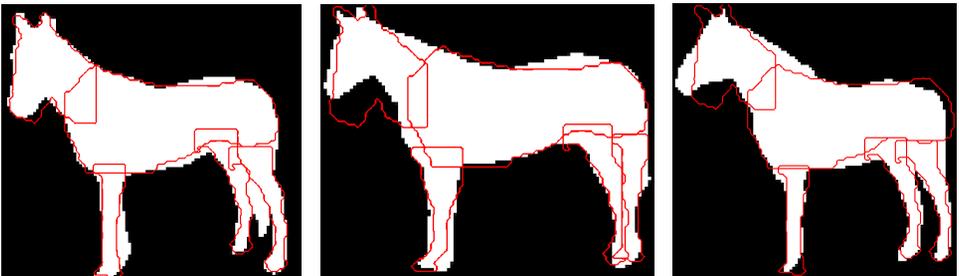


Fig. 8. Image segmentation and shape decomposition with fixed templates is fairly robust against moderate variation of the overall shape or the observer's viewpoint. No parameter tuning was involved.

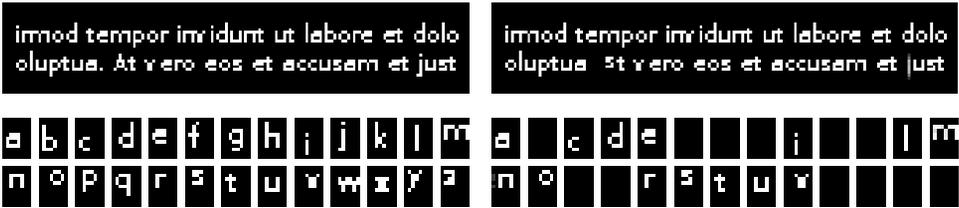


Fig. 9. Learning shape templates from examples. Top left: input image data. Bottom left: initial dictionary. Bottom right: learned dictionary. Top right: segmentation (decomposition) of the input data using the learned dictionary. Learning effectively reduces the initial number of templates by selecting those that are essential for decomposing the image data. Removed templates are automatically replaced by superpositions of remaining templates.

5.3. Shape template learning

Finally, we indicate the possibility to learn shape templates from sample data in order to disburden the user from defining templates by hand.

To this end, we added another regularizer to the objective function that enforces sparsity of the templates themselves. Unfortunately, determining the optimal templates along with the optimal segmentation is a highly nonconvex problem such that global optimality cannot be guaranteed and sufficiently accurate starting configurations are required.

In order to clearly demonstrate the essential points, Fig. 9 shows an academic scenario where the objective is to learn a small dictionary for recovering the text shown in the upper-left panel. As starting configuration we used the dictionary visualized in the lower left of Fig. 9, together with the corresponding optimal x resulting from the text segmentation, i.e., the global optimum of the inference procedure using fixed templates.

Applying alternatingly gradient ascent and gradient descent in the corresponding variables yield an improvement of the dictionary as depicted in the lower right of Fig. 9 while almost preserving reconstruction performance (upper-right panel). Nonexistent letters are removed from the dictionary, whereas letters occurring rarely are approximated by the superposition of more frequently occurring letters. For instance, both “p” and “b” are reconstructed using a combination of the letters “i,” “o,” and “l,” “o,” respectively.

6. Conclusion

We presented an approach to image segmentation based on sparse coverings with shape templates. To this end, we represented shapes by indicator functions and designed an objective function that takes into account overlapping templates. We derived a convex relaxation that provides a good compromise between approximation of the objective function and amenability to large-scale numerical optimization. Concerning the latter, we adopted a saddle-point formulation and investigated two

different strategies of sparse convex programming. Numerical experiments validate our approach and indicate its potential for computational image analysis.

Our further work will focus on the learning of shape templates. Because this problem is highly nonconvex due to the interleaving with image segmentation as a subroutine, there is considerable potential for further research. Concerning theoretical aspects, analyzing the reconstruction performance of our approach together with its robustness remains a challenge, because dictionaries of shape templates significantly differ from those considered in the compressed sensing literature. Additionally, we will augment our implementation with a stage for extracting image features that are significant for the segmentation of specific object categories.

Appendix

Proof. (Proposition 3.1) For the conjugate of h

$$\begin{aligned} h^*(q) &= \sup_{p \in \mathbb{R}^n} \{ \langle p, q \rangle - h(p) \} \\ &= \max \left\{ 0, \sup_{p \in \{0,1\}^n, p \neq 0} \langle p, q \rangle - 1 \right\}. \end{aligned}$$

If there is some $q_i \geq 0$, the maximizer p of the right-hand side can be set according to $p_i = 1 \Leftrightarrow q_i \geq 0$ (otherwise it contradicts the maximality assumption). If all $q_i < 0$ then the right-hand side must be smaller than $-1 = \sum_i (q_i)_+ - 1$, where $(q_i)_+ := \max\{0, q_i\}$. Thus, we have

$$h^*(q) = \max \left\{ 0, \sum_i (q_i)_+ - 1 \right\}. \tag{19}$$

For the biconjugate h^{**} , we have

$$\begin{aligned} h^{**}(r) &= \sup_{q \in \mathbb{R}^n} \{ \langle r, q \rangle - h^*(q) \} \\ &= \sup_{q \in \mathbb{R}^n} \left\{ \langle r, q \rangle - \max \left\{ 0, \sum_i (q_i)_+ - 1 \right\} \right\}. \end{aligned}$$

If there is some $r_i < 0$ we may let $q_i \rightarrow -\infty$ (and all other $q_j = 0$), so $h^{**}(r) = +\infty$. If there is some $r_i > 1$ we may let $q_i \rightarrow +\infty$ and again have $h^{**}(r) = +\infty$, thus

$$h^{**}(r) = \begin{cases} \sup_{q \in \mathbb{R}^n} \left(\langle q, r \rangle - \max \left\{ 0, \sum_i (q_i)_+ - 1 \right\} \right), & r \in [0, 1]^n \\ +\infty, & \text{otherwise.} \end{cases}$$

The first term is equivalent to

$$\begin{aligned} &\max \left\{ \sup_{q, \sum_i (q_i)_+ \leq 1} \langle q, r \rangle, \sup_{c > 1} \sup_{\sum_i (q_i)_+ = c} \left(\langle q, r \rangle - \sum_i (q_i)_+ + 1 \right) \right\} \\ &\stackrel{r_i \geq 0}{\equiv} \max \left\{ \text{vecmax}(r), \sup_{c > 1} \sup_{\sum_i (q_i)_+ = c} (\langle q, r \rangle - c + 1) \right\} \end{aligned}$$

$$\begin{aligned}
 &= \max \left\{ \text{vecmax}(r), \sup_{c>1} \left(1 - c + \sup_{\sum_i (q_i)_+ = c} \langle q, r \rangle \right) \right\} \\
 &= \max \left\{ \text{vecmax}(r), \sup_{c>1} (1 - c + c \text{vecmax}(r)) \right\} \\
 &\stackrel{r_i \leq 1 \Rightarrow \text{vecmax}(r) \leq 1}{=} \max \{ \text{vecmax}(r), \text{vecmax}(r) \} = \text{vecmax}(r)
 \end{aligned}$$

which implicates the assertion for h^{**} . In a similar fashion

$$\begin{aligned}
 g^*(q) &= \sup_{p \in \mathbb{R}^n} \{ \langle p, q \rangle - g(p) \} = \max \left\{ -1, \sup_{p \in \{0,1\}^n, p \neq 0} \langle p, q \rangle \right\} \\
 &= \max \left\{ -1, \begin{cases} \sum_i (q_i)_+, & \exists q_i \geq 0, \\ \text{vecmax}(q), & \text{otherwise,} \end{cases} \right\} \\
 &= \begin{cases} \sum_i (q_i)_+, & 0 \leq \text{vecmax}(q), \\ \text{vecmax}(q), & -1 < \text{vecmax}(q) < 0, \\ -1, & \text{vecmax}(q) \leq -1. \end{cases}
 \end{aligned}$$

The biconjugate is then defined as

$$g^{**}(r) = \sup_{q \in \mathbb{R}^n} \{ \langle q, r \rangle - g^*(q) \}.$$

Again, for some $r_i < 0$ let $q_1 = \dots = q_k \rightarrow -\infty$, thus $g^{**}(r) = +\infty$. For $r_i > 1$ let $q_i \rightarrow +\infty$. So, as expected, $g^{**}(r) = +\infty$ for $r \notin [0, 1]^n$. Thus

$$g^{**}(r) = \begin{cases} \sup_{q \in \mathbb{R}^n} \left(\langle q, r \rangle - \begin{cases} \sum_i (q_i)_+, & 0 \leq \text{vecmax}(q), \\ \text{vecmax}(q), & -1 < \text{vecmax}(q) < 0, \\ -1, & \text{vecmax}(q) \leq -1. \end{cases} \right) & r \in [0, 1]^n, \\ +\infty, & \text{otherwise.} \end{cases}$$

Again focusing on the first term, we get

$$\sup_{q \in \mathbb{R}^n} \left(\langle q, r \rangle - \begin{cases} \sum_i (q_i)_+, & 0 \leq \text{vecmax}(q), \\ \text{vecmax}(q), & -1 < \text{vecmax}(q) < 0, \\ -1, & \text{vecmax}(q) \leq -1. \end{cases} \right)$$

$$\begin{aligned}
 &= \max \left\{ \sup_{\text{vecmax}(q) \geq 0} \{ \langle q, r \rangle - \sum_i (q_i)_+ \}, \sup_{\text{vecmax}(q) \leq -1} \{ \langle q, r \rangle + 1 \}, \right. \\
 &\quad \left. - \sup_{-1 < \text{vecmax}(q) < 0} \{ \langle q, r \rangle - \text{vecmax}(q) \} \right\} \\
 &= \max \left\{ 0, 1 - e^\top r, \sup_{0 < c < 1} \left(c + \sup_{\text{vecmax}(q) = -c} \langle q, r \rangle \right) \right\} \\
 &= \max \left\{ 0, 1 - e^\top r, \sup_{0 < c < 1} c(1 - e^\top r) \right\} \\
 &= \begin{cases} \max\{0, 1 - e^\top r, 1 - e^\top r\}, & 1 - e^\top r \geq 0, \\ \max\{0, 1 - e^\top r, 0\}, & 1 - e^\top r < 0, \end{cases} \\
 &= \max\{0, 1 - e^\top r\} = (1 - e^\top r)_+,
 \end{aligned}$$

from which Proposition 3.1 follows. □

References

- Aharon, M., Elad, M. and Bruckstein, A. (2006). K-SVD: An algorithm for designing overcomplete dictionaries for sparse representations. *IEEE Trans. Signal Processing*, **54**: 4311–4322.
- Arrow, K. J., Hurwicz, L. and Uzawa, H. (1964). *Studies in Linear and Non-Linear Programming. With Contributions by Hollis B. Chenery [and others]*. Stanford University Press, Stanford, CA.
- Aujol, J.-F. (2008). Some algorithms for total variation based image restoration. *CMLA Preprint*, 2008–2005.
- Becker, S., Bobin, J. and Candès, E. J. (2009). *NESTA: A Fast and Accurate First-Order Method for Sparse Recovery*.
- Bergtholdt, M., Kappes, J., Schmidt, S. and Schnörr, C. (2010). A study of parts-based object class detection using complete graphs. *Int. J. Comp. Vision*, **87**(1–2): 93–117.
- Borenstein, E. and Ullman, S. (2002). Class-specific, top-down segmentation. In *European Conference on Computer Vision*, Vol. 2351 of *LNCS*, Springer, pp. 109–124.
- Borenstein, E. and Ullman, S. (2008). Combined top-down/bottom-up segmentation. *IEEE Trans. Patt. Anal. Mach. Intell.*, **30**(12): 2109–2125.
- Candès, E., Rudelson, M., Tao, T. and Vershynin, R. (2005). Error correcting via linear programming. In *46th Annual IEEE Symposium on Foundations of Computer Science*, pp. 295–308.
- Chambolle, A., Cremers, D. and Pock, T. (2008). A convex approach for computing minimal partitions, R.I. 649, CMAP, CNRS UMR 7641, Ecole Polytechnique.
- Chan, T., Esedoglu, S. and Nikolova, M. (2006). Algorithms for finding global minimizers of image segmentation and denoising models. *SIAM J. Appl. Math.*, **66**(5): 1632–1648.
- Chan, T. F. and Zhu, W. (2005). Level set based shape prior segmentation. In *CVPR*, Vol. 2, pp. 1164–1170.
- Chen, S., Donoho, D. and Saunders, M. (2001). Atomic decomposition by basis pursuit, *SIAM Rev.*, **43**(1): 129–159.
- Cohen, A., Dahmen, W. and DeVore, R. (2009). Compressed sensing and best k -term approximation. *J. Am. Math. Soc.*, **22**(1): 211–231.

- Cremers, D., Kohlberger, T. and Schnörr, C. (2003). Shape statistics in kernel space for variational image segmentation. *Pattern Recognition*, **36**(9): 1929–1943.
- Cremers, D., Rousson, M. and Deriche, R. (2007). A review of statistical approaches to level set segmentation: Integrating color, texture, motion and shape. *Int. J. Comp. Vision*, **72**(2): 195–215.
- Cremers, D., Sochen, N. and Schnörr, C. (2006). Multiphase dynamic labeling for variational recognition-driven image segmentation. *Int. J. Comp. Vision*, **66**(1): 67–81.
- Cremers, D., Tischhäuser, F., Weickert, J. and Schnörr, C. (2002). Diffusion snakes: Introducing statistical shape knowledge into the Mumford–Shah functional. *Int. J. Comp. Vision*, **50**(3): 295–313.
- Kohli, P., Rihan, J., Bray, M. and Torr, P. (2008). Simultaneous segmentation and pose estimation of humans using dynamic graph cuts. *Int. J. Comp. Vision*, **79**(3): 285–298.
- Lellmann, J., Becker, F. and Schnörr, C. (2009). Convex optimization for multiclass image labeling with a novel family of total variation based regularizers. In *International Conference on Computer Vision*, pp. 646–653.
- Lellmann, J. and Schnörr, C. (2010). *Continuous Multiclass Labeling Approaches and Algorithms*. Technical Report University Heidelberg, February.
- Mumford, D. and Shah, J. (1989). Optimal approximations by piecewise smooth functions and associated variational problems. *Comm. Pure Appl. Math.*, **42**: 577–685.
- Nesterov, Y. (2004). Smooth minimization of non-smooth functions. *Math. Prog.*, **103**(1): 127–152.
- Pock, T., Cremers, D., Bischof, H. and Chambolle, A. (2009). An algorithm for minimizing the Mumford–Shah functional. In *International Conference on Computer Vision*.
- Popov, L. D. (1980). A modification of the Arrow–Hurwicz method for search of saddle points. *Math. Notes*, **28**: 845–848.
- Rockafellar, R. (1970). *Convex Analysis*, Princeton up, Princeton, NJ.
- Rubinstein, R., Zibulevsky, M. and Elad, M. Double sparsity: Learning sparse dictionaries for sparse signal approximation. *IEEE Trans. Signal Processing* (to appear).
- Yu, G., Sapiro, G. and Mallat, S. (2010). Image modeling and enhancement via structured sparse model selection. In *Proceedings of IEEE International Conference on Image Processing*.
- Zach, C., Gallup, D., Frahm, J.-M. and Niethammer, M. (2008). Fast global labeling for real-time stereo using multiple plane sweeps. In *Vis. Mod. Vis.*, pp. 243–252.