Image Segmentation by Branch-and-Mincut

Victor Lempitsky, Andrew Blake, and Carsten Rother

Microsoft Research Cambridge

Abstract. Efficient global optimization techniques such as graph cut exist for energies corresponding to binary image segmentation from low-level cues. However, introducing a high-level prior such as a shape prior or a color-distribution prior into the segmentation process typically results in an energy that is much harder to optimize. The main contribution of the paper is a new global optimization framework for a wide class of such energies. The framework is built upon two powerful techniques: graph cut and branch-and-bound. These techniques are unified through the derivation of lower bounds on the energies. Being computable via graph cut, these bounds are used to prune branches within a branch-and-bound search.

We demonstrate that the new framework can compute globally optimal segmentations for a variety of segmentation scenarios in a reasonable time on a modern CPU. These scenarios include unsupervised segmentation of an object undergoing 3D pose change, category-specific shape segmentation, and the segmentation under intensity/color priors defined by Chan-Vese and GrabCut functionals.

1 Introduction

Binary image segmentation is often posed as a graph partition problem. This is because efficient graph algorithms such as mincut permit fast global optimization of the functionals measuring the quality of the segmentation. As a result, difficult image segmentation problems can be solved efficiently, robustly, and independently of initialization. Yet, while graphs can represent energies based on localized low-level cues, they are much less suitable for representing *non-local* cues and priors describing the foreground or the background segment as a whole.

Consider, for example, the situation when the shape of the foreground segment is known *a priori* to be similar to a particular template (segmentation with shape priors). Graph methods can incorporate such a prior for a single pre-defined and pre-located shape template[14, 21]. However, once the pose of the template is allowed to change, the relative position of each graph edge with respect to the template becomes unknown, and the *non-local* property of shape similarity becomes hard to express with local edge weights. Another example would be the segmentation with non-local color priors, when the color of the foreground and/or background is known a priori to be described by some parametric distribution (e.g. a mixture of the Gaussians as in the case of GrabCut [26]). If the parameters of these distributions are allowed to change, such a non-local prior depending on the segment as a whole becomes very hard to express with the local edge weights.

An easy way to circumvent the aforementioned difficulties is to alternate the graph partitioning with the reestimation of non-local parameters (such as the template pose or the color distribution). A number of approaches [6, 17, 26, 16] follow this path. Despite the use of the global graph cut optimization inside the loop, local search over the prior parameters turns these approaches into local optimization techniques akin to variational segmentation [7, 9, 24, 29]. As a result, these approaches may get stuck in local optima, which in many cases correspond to poor solutions.

The goal of this paper is to introduce a new framework for computing *globally* optimal segmentations under non-local priors. Such priors are expressed by replacing fixed-value edge weights with edge weights depending on *non-local parameters*. The global minimum of the resulting energy that depends on both the graph partition and the non-local parameters is then found using the branch-and-bound tree search. Within the branch-and-bound, lower bounds over tree branches are efficiently evaluated by computing minimal cuts on a graph (hence the name *Branch-and-Mincut*).

The main advantage of the proposed framework is that the globally optimal segmentation can be obtained for a broad family of functionals depending on non-local parameters. Although the worst case complexity of our method is large (essentially, the same as the exhaustive search over the space of non-local parameters), we demonstrate that our framework can obtain globally optimal image segmentation in a matter of seconds on a modern CPU. Test scenarios include globally optimal segmentation with shape priors where the template shape is allowed to deform and to appear in various poses as well as image segmenta-

tion by the optimization of the Chan-Vese [7] and the GrabCut [26] functionals. In all cases, bringing in high-level non-local knowledge allows to solve difficult segmentation problems, where local cues (considered by most current global optimization approaches) were highly ambiguous.

2 Related Work

Our framework employs the fact that a submodular quadratic function of boolean variables can be efficiently minimized via minimum cut computation in the associated graph [2, 11, 19]. This idea has been successfully applied to binary image segmentation [3] and quickly gained popularity. As discussed above, the approach [3] still has significant limitations, as the high-level knowledge such as shape or color priors are hard to express with fixed local edge weights. These limitations are overcome in our framework, which allows the edge weights to vary.

In the restricted case, when unary energy potentials are allowed to vary and depend on a single scalar non-local parameter monotonically, efficient algorithms known as *parametric maxflow* have been suggested (see e.g. [20]). Our framework is however much more general then these methods (at a price of having higher worst-case complexity), as we allow both unary and pairwise energy terms to depend non-monotonically on a single or multiple non-local parameters. Such generality gives our framework flexibility in incorporating various high-level priors while retaining the globality of the optimization.

Image segmentation with non-local shape and color priors has attracted a lot of interest in the last years. As discussed above, most approaches use either local continuous optimization [29, 7, 24, 9] or iterated minimization alternating graph cut and search over non-local parameter space [26, 6, 17]. Unfortunately, both groups of methods are prone to getting stuck in poor local minima. Global-optimization algorithms have also been suggested [12, 27, 28] . In particular, simultaneous work [10] presented a framework that also utilizes branch-and-bound ideas (paired with continuous optimization in their case). While all these global optimization methods are based on elegant ideas, the variety of shapes, invariances, and cues that each of them can handle is limited compared to our method.

Finally, our framework may be related to branch-and-bound search methods in computer vision (e.g. [1, 22]). In particular, it should be noted that the way our framework handles shape priors is related to previous approaches [15, 13] that used tree search over shape hierarchies. However, neither of those approaches accomplish pixel-wise image segmentation.

3 Optimization Framework

In this section, we discuss our global energy optimization framework for obtaining image segmentations under non-local priors[1]. In the next sections, we detail how it can be used for the segmentation with non-local shape priors (Section 4) and non-local intensity/color priors (Section 5).

[1] The C++ code for this framework is available at the webpage of the first author.

3.1 Energy Formulation

Firstly, we introduce notation and give the general form of the energy that can be optimized in our framework. Below, we consider the pixel-wise segmentation of the image. We denote the pixel set as $\mathcal{V}$ and use letters p and q to denote individual pixels. We also denote the set of edges connecting adjacent pixels as $\mathcal{E}$ and refer to individual edges as to the pairs of pixels (e.g. p, q). In our experiments, the set of edges consisted of all 8-connected pixel pairs in the raster.

The segmentation of the image is given by its $0{-}1$ labeling $\mathbf{x} \in 2^{\mathcal{V}}$, where individual pixel labels x_p take the values 1 for the pixels classified as the foreground and 0 for the pixels classified as the background. Finally, we denote the non-local parameter as ω and allow it to vary over a discrete, possibly very large, set Ω. The general form of the energy function that can be handled within our framework is then given by:

$$E(\mathbf{x}, \omega) = C(\omega) + \sum_{p \in \mathcal{V}} F^p(\omega) \cdot x_p + \sum_{p \in \mathcal{V}} B^p(\omega) \cdot (1 - x_p) + \sum_{p,q \in \mathcal{E}} P^{pq}(\omega) \cdot |x_p - x_q| \ . \quad (1)$$

Here, $C(\omega)$ is a constant potential, which does not depend directly on the segmentation $\mathbf{x}$; $F^p(\omega)$ and $B^p(\omega)$ are the unary potentials defining the cost for assigning the pixel p to the foreground and to the background respectively; $P^{pq}(\omega)$ is the pairwise potential defining the cost of assigning adjacent pixels p and q to different segments. In our experiments, the pairwise potentials were taken non-negative to ensure the tractability of $E(\mathbf{x}, \omega)$ as the function of $\mathbf{x}$ for graph cut optimization [19].

All potentials in our framework depend on the non-local parameter $\omega \in \Omega$. In general, we assume that Ω is a discrete set, which may be large (e.g. millions of elements) and should have some structure (although, it need not be linearly or partially ordered). For the segmentation with shape priors, Ω will correspond to the product space of various poses and deformations of the template, while for the segmentation with color priors Ω will correspond to the set of parametric color distributions.

3.2 Lower Bound

Our approach optimizes the energy (1) exactly, finding its global minimum using branch-and-bound tree search [8], which utilizes the lower bound on (1) derived as follows:

$$\min_{x\in 2^{\mathcal{V}},\omega\in\Omega} E(\mathbf{x},\omega) = \min_{x\in 2^{\mathcal{V}}}\min_{\omega\in\Omega}\left[C(\omega)+\sum_{p\in\mathcal{V}}F^p(\omega)\cdot x_p+\sum_{p\in\mathcal{V}}B^p(\omega)\cdot(1-x_p)+\right.$$

$$\left.\sum_{p,q\in\mathcal{E}}P^{pq}(\omega)\cdot|x_p-x_q|\right] \geq \min_{\mathbf{x}\in 2^{\mathcal{V}}}\left[\min_{\omega\in\Omega}C(\omega)+\sum_{p\in\mathcal{V}}\min_{\omega\in\Omega}F^p(\omega)\cdot x_p+\right.$$

$$\left.\sum_{p\in\mathcal{V}}\min_{\omega\in\Omega}B^p(\omega)\cdot(1-x_p)+\sum_{p,q\in\mathcal{E}}\min_{\omega\in\Omega}P^{pq}(\omega)\cdot|x_p-x_q|\right] =$$

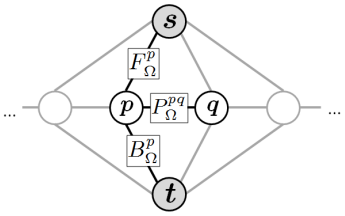
$$\min_{x\in 2^{\mathcal{V}}}\left[C_\Omega+\sum_{p\in\mathcal{V}}F^p_\Omega\cdot x_p+\sum_{p\in\mathcal{V}}B^p_\Omega\cdot(1-x_p)+\sum_{p,q\in\mathcal{E}}P^{pq}_\Omega\cdot|x_p-x_q|\right] = L(\Omega)\ . \quad (2)$$

Here, C_Ω, F^p_Ω, B^p_Ω, P^{pq}_Ω denote the minima of $C(\omega)$, $F^p(\omega)$, $B^p(\omega)$, $P^{pq}(\omega)$ over $\omega\in\Omega$ referred below as *aggregated potentials*. $L(\Omega)$ denotes the derived lower bound for $E(\mathbf{x},\omega)$ over $2^{\mathcal{V}}\otimes\Omega$. The inequality in (2) is essentially the Jensen inequality for the minimum operation.

The proposed lower bound possesses three properties crucial to the Branch-and-Mincut framework:

Monotonicity. For the nested domains of non-local parameters $\Omega_1\subset\Omega_2$ the inequality $L(\Omega_1)\geq L(\Omega_2)$ holds (the proof is given in the Appendix).

Computability. The key property of the derived lower bound is the ease of its evaluation. Indeed, this bound equals the minimum of a submodular quadratic pseudo-boolean function. Such function can be realized on a network graph such that each configuration of the binary variables is in one-to-one correspondence with an *st*-cut of the graph having the weight equal to the value of the function (plus a constant C_Ω) [2, 11, 19]. The minimal *st*-cut corresponding to the minimum of $L(\Omega)$ then can be computed in a low-polynomial of $|\mathcal{V}|$ time e.g. with the popular algorithm [5].



The fragment of the network graph realizing $L(\Omega)$ (edge weights shown in boxes). *(see e.g.[19] for details)*

Tightness. For a singleton Ω the bound is *tight*: $L(\{\omega\}) = \min_{x\in 2^{\mathcal{V}}} E(\mathbf{x},\omega)$. In such case, the minimal *st*-cut also yields the segmentation $\mathbf{x}$ optimal for this ω ($x_p = 0$ *iff* the respective vertex belongs to the *s*-component of the cut).

Note, that the fact that the lower bound (2) may be evaluated via *st*-mincut gives rise to a whole family of looser, but cheaper, lower bounds. Indeed, the minimal cut on a network graph is often found by pushing *flows* until the flow becomes maximal (and equal to the weight of the mincut) [5]. Thus, the sequence of intermediate flows provides a sequence of the increasing lower bounds on (1) converging to the bound (2) (**flow bounds**). If some upper bound on the minimum value is imposed, the process may be terminated earlier without computing the full maxflow/mincut. This happens when the new flow bound

exceeds the given upper bound. In this case it may be concluded that the value of the global minimum is greater than the imposed upper bound.

3.3 Branch-and-Bound Optimization

Finding the global minimum of (1) is, in general, a very difficult problem. Indeed, since the potentials can depend arbitrarily on the non-local parameter spanning arbitrary discrete set Ω, in the worst-case any optimization has to search exhaustively over Ω. In practice, however, any segmentation problem has some specifically-structured space Ω. This structure can be efficiently exploited by the branch-and-bound search detailed below.

We assume that the discrete domain Ω can be hierarchically clustered and the binary tree of its subregions $T_\Omega = \{\Omega = \Omega_0, \Omega_1, \dots \Omega_N\}$ can be constructed (binarity of the tree is not essential). Each non-leaf node corresponding to the subregion Ω_k then has two children corresponding to the subregions $\Omega_{ch1(k)}$ and $\Omega_{ch2(k)}$ such that $\Omega_{ch1(k)} \subset \Omega_k$, $\Omega_{ch2(k)} \subset \Omega_k$. Here, $ch1(\cdot)$ and $ch2(\cdot)$ map the index of the node to the indices of its children. Also, leaf nodes of the tree are in one-to-one correspondence with singleton subsets $\Omega_l = \{\omega_t\}$.

Given such tree, the global minimum of (1) can be efficiently found using the *best-first* branch-and-bound search [8]. This algorithm propagates a *front* of nodes in the top-down direction (Fig. 1). During the search, the front contains a set of tree nodes, such that each top-down path from the root to a leaf contains exactly one active vertex. In the beginning, the front contains the tree root Ω_0. At each step the active node with the smallest lower bound (2) is removed from the active front, while two of its children are added to the active front (by monotonicity property they have higher or equal lower bounds). Thus, an active front moves towards the leaves making local steps that increase the lowest lower bound of all active nodes. Note, that at each moment, this lowest lower bound of the front constitutes a lower bound on the global optimum of (1) over the whole domain.

At some moment of time, the active node with the smallest lower bound turns out to be a leaf $\{\omega'\}$. Let $\mathbf{x}'$ be the optimal segmentation for ω' (found via minimum st-cut). Then, $E(x', \omega') = L(\omega')$ (tightness property) is by assumption the lowest bound of the front and hence a lower bound on the global optimum over the whole domain. Consequently, (x', ω') is a global minimum of (1) and the search terminates without traversing the whole tree. In our experiments, the number of the traversed nodes was typically very small (two-three orders of magnitude smaller then the size of the full tree). Therefore, the algorithm performed global optimization much faster than exhaustive search over Ω.

In order to further accelerate the search, we exploit the coherency between the mincut problems solved at different nodes. Indeed, the maximum flow as well as auxiliary structures such as shortest path trees computed for one graph may be "reused" in order to accelerate the computation of the minimal st-cut on another similar graph [3, 18]. For some applications, this trick may give an order of magnitude speed-up for the evaluation of lower bounds.

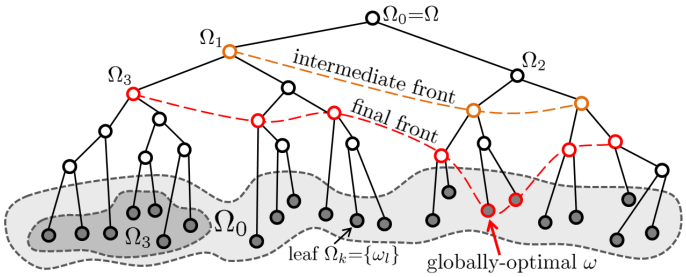


Fig. 1. *Best-first branch-and-bound* optimization on the tree of nested regions finds the globally-optimal ω by the top-down propagation of the *active front* (see text for details). At the moment when the lowest lower bound of the front is observed at leaf node, the process terminates with the global minimum found without traversing the whole tree.

In addition to the best-first branch-and-bound search we also tried the *depth-first* branch-and-bound [8]. When problem-specific heuristics are available that give good initial solutions, this variant may lead to moderate (up to a factor of 2) time savings. Interestingly, the depth-first variant of the search, which maintains upper bounds on the global optimum, may benefit significantly from the use of flow bounds discussed above. Nevertheless, we stick with the best-first branch-and-bound for the final experiments due to its generality (no need for initialization heuristics).

In the rest of the paper we detail how the general framework developed above may be used within different segmentation scenarios.

4 Segmentation with Shape Priors

4.1 Constructing Shape Prior

We start with the segmentation with shape priors. The success of such segmentation crucially depends on the way shape prior is defined. Earlier works have often defined this prior as a Gaussian distribution of some geometrical shape statistics (e.g. control point positions or level set functions) [29, 24]. In reality, however, pose variance and deformations specific to the object of interest lead to highly non-Gaussian, multi-modal prior distributions. For better modeling of prior distributions, [9] suggested the use of non-parametric kernel densities. Our approach to shape modeling is similar in spirit, as it also uses exemplar-based prior. Arguably, it is more direct, since it involves the distances between the binary segmentations themselves, rather than their level set functions. Our approach to shape modeling is also closely related to [15] that used shape hierarchies to detect or track objects in image edge maps.

We assume that the prior is defined by the set of exemplar binary segmentations $\{\mathbf{y}^{\omega}|\omega \in \Omega\}$, where Ω is a discrete set indexing the exemplar segmentations.

Then the following term introduces a joint prior over the segmentation and the non-local parameter into the segmentation process:

$$E_{\text{prior}}(\mathbf{x}, \omega) = \rho(\mathbf{x}, \mathbf{y}^{\omega}) = \sum_{p \in \mathcal{V}} (1 - y_p^{\omega}) \cdot x_p + \sum_{p \in \mathcal{V}} y_p^{\omega} \cdot (1 - x_p) \ , \tag{3}$$

where ρ denotes the Hamming distance between segmentations. This term clearly has the form (1) and therefore its combinations with other terms of this form can be optimized within our framework. Being optimized over the domain $2^{\mathcal{V}} \otimes \Omega$, this term would encourage the segmentation $\mathbf{x}$ to be close in the Hamming distance to some of the exemplar shapes. Note, that the Hamming distance in the continuous limit may be interpreted as the $L1$-distance between shapes. It is relatively straightforward to modify the term (3) to replace the Hamming distance with discrete approximations of other distances ($L2$, truncated $L1$ or $L2$, data-driven Mahalonobis distance, etc.).

The full segmentation energy then may be defined by adding a standard contrast-sensitive edge term [3]:

$$E_{\text{shape}}(\mathbf{x}, \omega) = E_{\text{prior}}(\mathbf{x}, \omega) + \sum_{p,q \in \mathcal{E}} \lambda \frac{e^{-\frac{||K_p - K_q||}{\sigma}}}{|p - q|} \cdot |x_p - x_q| \ , \tag{4}$$

where $||K_p - K_q||$ denote the SAD (L1) distance between RGB colors of the pixels p and q in the image (λ and σ were fixed throughout the experiments described in this section), $|p-q|$ denotes the distance between the centers of the pixels p and q (being either 1 or $\sqrt{2}$ for the 8-connected grid). The functional (4) thus incorporates the shape prior with edge-contrast cues.

In practice, the set Ω_{shape} could be huge, e.g. tens of millions exemplars. Therefore, representation and hierarchical clustering of the exemplar segmentations $y^{\omega}, \omega \in \Omega$ may be challenging. In addition, the aggregated potentials for each node of the tree should be precomputed and stored in memory. Fortunately, this is accomplishable in many cases when the translation invariance is exploited. In more detail, the set Ω_{shape} is factorized it into the Cartesian product of two sets $\Omega_{\text{shape}} = \Delta \otimes \Theta$. The factor set Δ indexes the set of all exemplar segmentations y_{δ} centered at the origin (this set would typically correspond to the variations in scale, orientation as well as non-rigid deformations). The factor set Θ then corresponds to the shift transformations and ensures the translation invariance of the prior. Any exemplar segmentation $y_{\omega}, \omega = \delta \otimes \theta$ is then defined as some exemplar segmentation y_{δ} centered at the origin and then shifted by the shift θ.

Being much smaller than Ω_{shape}, both factor sets can be clustered in hierarchy trees. For the factor set Δ we used agglomerative clustering (a complete linkeage algorithm that uses the Hamming distance between the exemplar segmentations). The factor set Θ uses the natural hierarchical clustering of the quad-tree. Then the tree over Ω_{shape} is defined as a "product" of the two factor trees (we omit the details about the particular implementation). The aggregated potentials F_{Ω} and B_{Ω} in (2) for tree nodes are precomputed in a bottom-up

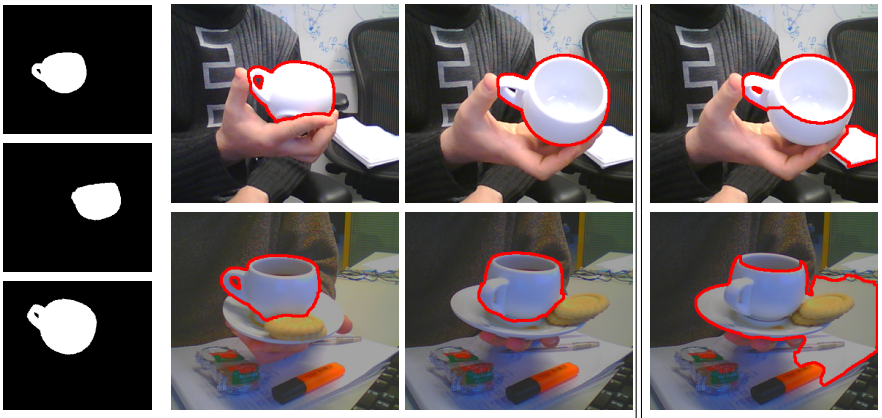


Fig. 2. Using the shape prior constructed from the set of exemplars (left column) our approach can accomplish segmentation of an object undergoing general 3D pose changes within two differently illuminated sequences (two middle columns). Note the varying topology of the segmentations. For comparison, we give the results of a standard graph cut segmentation (right column): even with parameters tuned specifically to the test images, separation is entirely inaccurate.

pass and stored in memory. The redundancy arising from translation invariance is used to keep the required amount of memory reasonable.

Note the three properties of our approach to segmentation with shape priors. Firstly, since any shapes can be included in Ω_{shape}, general 3D pose transformations and deformations may be handled. Secondly, the segmentations may have general varying topology not restricted to segments with single-connected boundaries. Thirdly, our framework is general enough to introduce other terms in the segmentation process (e.g. regional terms used in a standard graph cut segmentation [3]). These properties of our approach are demonstrated within the following experiments.

4.2 Experiments

Single object+3D pose changes. In our first experiment, we constructed a shape prior for a single object (a coffee cup) undergoing 3D pose changes. We obtained a set of outlines using "blue-screening". We then normalized these outlines (by centering at the origin, resizing to a unit scale and orienting the principle axes with the coordinate axes). After that we clustered the normalized outlines using k-means. A representative of each cluster was then taken into the exemplar set. After that we added scale variations, in-plane rotations, and translations. As a result, we got a set $\{y_\omega | \omega \in \Omega_{\text{shape}}\}$ containing about 30,000,000 exemplar shapes(while the set Δ contained about 1900 shapes).

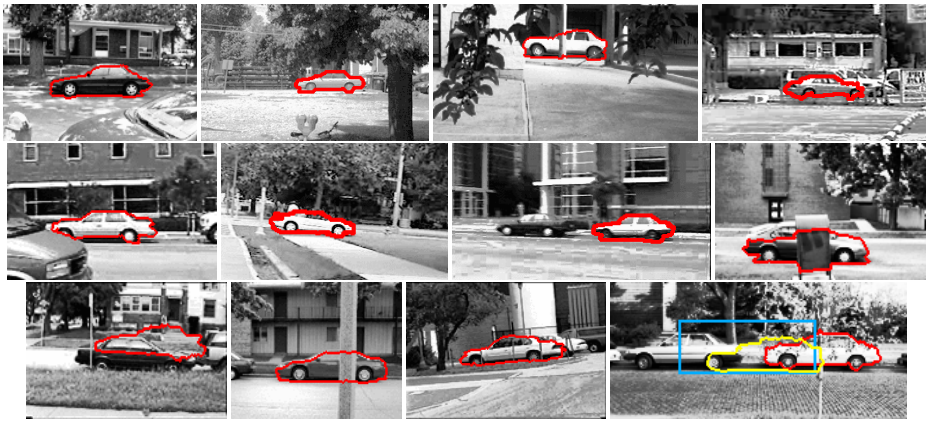


Fig. 3. Results of the global optimization of (5) on some of the 170 UIUC car images including 1 of the 2 cases where localization failed (bottom left). In the case of the bottom right image, the global minimum of (4) (yellow) and the result of our feature-based car detector (blue) gave erroneous localization, while the global minimum of their combination (5) (red) represented an accurate segmentation.

The results of the global optimization of the functional (4) for the frames from the two sequences containing clutter and camouflage are shown in Fig. 2. On average, we observed that segmenting 312x272 image took about 30 seconds of an Intel-2.40 GHz CPU and less than 1 Gb of RAM. The proportion of the nodes of the tree traversed by the active front was on average about $1:5000$. Thus, branch-and-bound tree search used in our framework improved very considerably over exhaustive search, which would have to traverse all leaves ($1:2$ of the tree).

As a baseline algorithm, we considered the segmentation with a "standard" graph cut functional, replacing non-local shape prior term with a local intensity-based term $\sum_{p\in\mathcal{V}}(I-I_p)\cdot x_p$, adjusting the constant I for each frame so that it gives the best results. However, since the intensity distributions of the cup and the backgrounds overlapped significantly, the segmentations were grossly erroneous (Fig. 2 – right column).

Object class+translation invariance. In the second experiment, we performed the segmentation with shape priors on UIUC car dataset (the version without scale variations), containing 170 images with cars in uncontrolled environment (city streets). The prior set Δ was built by manual segmentation of 60 training images coming with the dataset. The set of shifts Θ was defined by the varying size of test images. While the test image sizes varyied from 110x75 to 360x176, the size of Ω_{shape} varied from 18,666 to 2,132,865. We computed the globally optimal segmentations under the constructed prior using the energy (4).

Using the bounding boxes of the cars provided with the dataset, we found that in 6.5% of the images the global minima corresponded to clutter rather than cars. To provide a baseline for localization accuracy based on edge cues and a

set of shape templates, we considered Chamfer matching (as e.g. in [15]). For the comparison we used the same set of templates, which were matched against truncated Canny-based chamfer distance (with optimally tuned truncation and Canny sensitivity parameters). In this way, the optimal localization failed (i.e. corresponded to clutter rather than a car) in 12.4% of the images.

Clearly, segmenting images using (4) takes into account the shape prior and edge-contrast cues, but ignores the appearance typical for the object category under consideration. At the same time, there exists a large number of algorithms working with image appearance cues and performing object detection based on these cues (see e.g. [23] and references therein). Typically, such algorithms produce the likelihood of the object presence either as a a function of a bounding box or even in the form of per-pixel "soft segmentation" masks. Both types of the outputs can be added into the functional (1) either via constant potential $C(\Omega)$ or via unary potentials. In this way, such appearance-based detectors can be integrated with shape prior and edge-contrast cues.

As an example of such integration, we devised a simple detector similar in spirit to [23]. The detector looked for the appearance features typical for cars (wheels) using normalized cross-correlation. Each pixel in the image then "voted" for the location of the car center depending on the strength of the response to the detector and the relative position of the wheels with respect to the car center observed on the training dataset. We then added an additional term $C_{\text{vote}}(\omega)$ in our energy (1) that for each ω equaled minus the accumulated strength of the votes for the center of y_ω:

$$E_{\text{shape\&detect}}(\mathbf{x},\omega) = C_{\text{vote}}(\omega) + E_{\text{prior}}(\mathbf{x},\omega) + \sum_{p,q\in\mathcal{E}} \lambda \frac{e^{-\frac{||K_p - K_q||}{\sigma}}}{|p-q|} \cdot |x_p - x_q| \ , \quad (5)$$

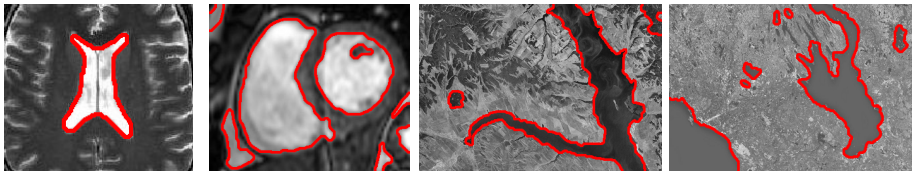
Adding the appearance-based term improved the robustness of the segmentation, as the global optima of (5) corresponded to clutter only in 1.2% of the images. The global minima found for some of the images are shown in Fig. 3. Note, that for our simple detector on its own the most probable bounding box corresponded to clutter on as much as 14.7% of the images.

In terms of the performance, on average, for the functional (5) the segmentation took 1.8 seconds and the proportion of the tree traversed by the active front was $1:441$. For the functional (4), the segmentation took 6.6 seconds and the proportion of the tree traversed by the active front was $1:131$. This difference in performance is natural to branch-and-bound methods: the more difficult and ambiguous is the optimization problem, the larger is the portion of the tree that has to be investigated.

5 Segmentation with Color/Intensity Priors

Our framework can also be used to impose non-local priors on the intensity or color distributions of the foreground and background segments, as the examples below demonstrate.

5.1 Segmenting Grayscale Images: Chan-Vese Functional



183x162, time=3s, proportion=1:115 | 300x250, time=4s, proportion=1:103 | 385x264, time=16s, proportion=1:56 | 371x255, time=21s, proportion=1:98

Fig. 4. The global minima of the Chan-Vese functional for medical and aerial images. These global minima were found using our framework in the specified amount of time; specified proportion of the tree was traversed.

In [7] Chan and Vese have proposed the following popular functional for the variational image segmentation problem:

$$E(S, c^f, c^b) = \mu \int_{\partial S} dl + \nu \int_S dp + \lambda_1 \int_S \left(I(p) - c^f\right)^2 dp + \lambda_2 \int_{\bar{S}} \left(I(p) - c^b\right)^2 dp \,, \tag{6}$$

where S denotes the foreground segment, and $I(p)$ is a grayscale image. The first two terms measure the length of the boundary and the area, the third and the forth terms are the integrals over the fore- and background of the difference between image intensity and the two intensity values c^f and c^b, which correspond to the average intensities of the respective regions. Traditionally, this functional is optimized using level set framework [25]converging to one of its local minima.

Below, we show that the discretized version of this functional can be optimized globally within our framework. Indeed, the discrete version of (6) can be written as (using notation as before):

$$\begin{aligned} E(\mathbf{x}, (c^f, c^b)) = & \sum_{p,q \in \mathcal{E}} \frac{\mu}{|p-q|} \cdot |x_p - x_q| + \\ & \sum_{p \in \mathcal{V}} \left(\nu + \lambda_1 (I(p) - c^f)^2\right) \cdot x_p + \sum_{p \in \mathcal{V}} \lambda_2 \left(I(p) - c^b\right)^2 \cdot (1 - x_p) \,. \end{aligned} \tag{7}$$

Here, the first term approximates the first term of (6) (the accuracy of the approximation depends on the size of the pixel neighborhood [4]), and the last two terms express the last three terms of (6) in a discrete setting.

The functional (7) clearly has the form (1) with non-local parameter $\omega = \{c^f, c^b\}$. Discretizing intensities $c^f$and c^b into 255 levels and building a quad-tree over their joint domain, we can apply our framework to find the global minima of (6). Example globally optimal segmentations are shown on Fig. 4.

5.2 Segmenting Color Images: GrabCut functional

In [26], the *GrabCut* framework for the interactive color image segmentation based on Gaussian mixtures was proposed. In GrabCut, the segmentation is driven by the following energy:

$$E_{\text{GrabCut}}(\mathbf{x}, (GM^f, GM^b)) = \sum_{p\in V} -\log(\mathrm{P}(K_p|\, GM^f))\cdot x_p + \\ + \sum_{p\in V} -\log(\mathrm{P}(K_p|\, GM^b))\cdot(1-x_p) + \sum_{p,q\in\mathcal{E}} \frac{\lambda_1 + \lambda_2 \cdot e^{-\frac{||K_p - K_q||^2}{\beta}}}{|p-q|}\cdot|x_p - x_q| \,. \tag{8}$$

Here, GM^f and GM^b are Gaussian mixtures in RGB color space and the first two terms of the energy measure how well these mixtures explain colors K_p of pixels attributed to fore- and background respectively. The third term is the contrast sensitive edge term, ensuring that the segmentation boundary is compact and tends to stick to color region boundaries in the image. In addition to this energy, the user provides supervision in the form of a bounding rectangle and brush strokes, specifying which parts of the image should be attributed to the foreground and to the background.

The original method [26] minimizes the energy within EM-style process, alternating between (i) the minimization of (8) over $\mathbf{x}$ given GM^f and GM^b and (ii) refitting the mixtures GM^f and GM^b given $\mathbf{x}$. Despite the use of the global graph cut optimization within the segmentation update step, the whole process yields only a local minimum of (8). In [26], the segmentation is initialized to the provided bounding box and then typically shrinks to one of the local minima.

The energy (8) has the form (1) and therefore can be optimized within Branch-and-Mincut framework, provided that the space of non-local parameters (which in this case is the joint space of the Gaussian mixtures for the foreground and for the background) is discretized and the tree of the subregions is built. In this scenario, however, the dense discretization of the non-local parameter space is infeasible (if the mixtures contain n Gaussians then the space is described by $20n-2$ continuous parameters). It is possible, nevertheless, to choose a much smaller discrete subset Ω that is still likely to contain a good approximation to the globally-optimal mixtures.

To construct such Ω, we fit a mixture of $M = 8$ Gaussians $G_1, G_2, ...G_M$ with the support areas $a_1, a_2, ...a_M$ to the *whole* image. The support area a_i here counts the number of pixels p such as $\forall j\ \mathrm{P}(K_p|G_i) \geq \mathrm{P}(K_p|G_j)$. We assume that the components are ordered such that the support areas decrease ($a_i > a_{i+1}$). Then, the Gaussian mixtures we consider are defined by the binary vector $\beta = \{\beta_1, \beta_2 \ldots \beta_M\} \in \{0,1\}^M$ specifying which Gaussians should be included into the mixture: $\mathrm{P}(K|\, GM(\beta)) = \sum_i \beta_i a_i \mathrm{P}(K|G_i) \,/\, \sum_i \beta_i a_i$.

The overall set Ω is then defined as $\{0,1\}^{2M}$, where odd bits correspond to the foreground mixture vector β^f and even bits correspond to the background mixture vector β^b. Vectors with all even bits and/or all odd bits equal to zero do not correspond to meaningful mixtures and are therefore assigned an infinite

cost. The hierarchy tree is naturally defined by the bit-ordering (the first bit corresponding to subdivision into the first two branches etc.).

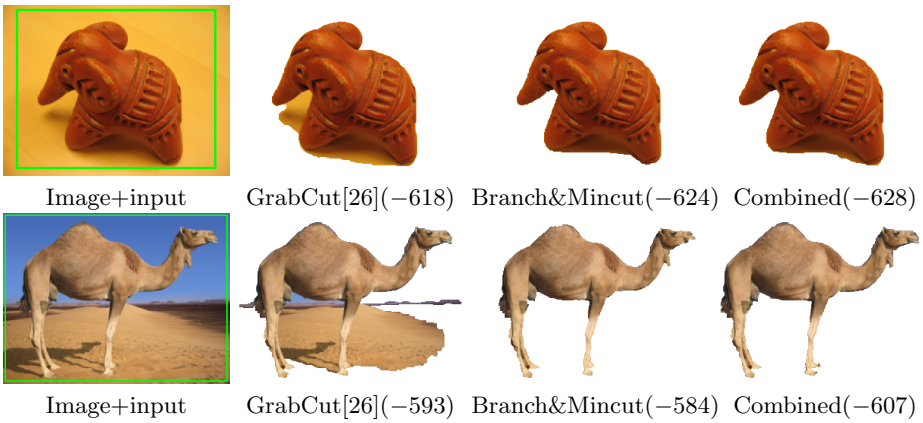


Fig. 5. Being initialized with the user-provided bounding rectangle (shown in green in the first column) as suggested in [26], EM-style process [26] converges to a local minimum (the second column). Branch-and-Mincut result (the third column) escapes that local minimum and after EM-style improvement lead to the solution with much smaller energy and better segmentation accuracy (the forth column). Energy values are shown in brackets.

Depending on the image and the value of M, the solutions found by Branch-and-Mincut framework may have larger or smaller energy (8) than the solutions found by the original EM-style method [26]. This is because Branch-and-Mincut here finds the global optimum over the subset of the domain of (8) while [26] searches locally but within the continuous domain. However, for all 15 images in our experiments, improving Branch-and-Mincut solutions with a few EM-style iterations [26] gave lower energy than the original solution of [26]. In most cases, these additional iterations simply refit the Gaussians properly and change very few pixels near boundary (see Fig. 5).

In terms of performance, for $M = 8$ the segmentation takes on average a few dozen seconds (10s and 40s for the images in Fig. 5) for 300x225 image. The proportion of the tree traversed by an active front is one to several hundred (1:963 and 1:283 for the images in Fig. 5).

This experiment suggests the usefulness of Branch-and-Mincut framework as a mean of obtaining good initial point for local methods, when the domain space is too large for an exact branch-and-bound search.

6 Conclusion

The Branch-and-Mincut framework presented in this paper finds global optima of a wide class of energies dependent on the image segmentation mask and non-local parameters. The joint use of branch-and-bound and graph cut allows efficient traversal of the solution space. The developed framework is useful within a variety of image segmentation scenarios, including segmentation with non-local shape priors and non-local color/intensity priors.

Future work includes the extension of Branch-and-Mincut to other problems, such as simultaneous stitching and registration of images, as well as deriving analogous branch-and-bound frameworks for combinatorial methods other than binary graph cut, such as minimum ratio cycles and multilabel MRF inference.

7 Acknowledgements

We would like to acknowledge discussions and feedback from Vladimir Kolmogorov and Pushmeet Kohli. Vladimir has also kindly made several modifications of his code of [5] that allowed to reuse network flows more efficiently.

References

1. S. Agarwal, M. Chandaker, F. Kahl, D. Kriegman, S. Belongie: Practical Global Optimization for Multiview Geometry. In ECCV 2006.
2. E. Boros, P. Hammer: Pseudo-boolean optimization. Discrete Applied Mathematics, 123(1-3), 2002.
3. Y. Boykov, M.-P. Jolly: Interactive Graph Cuts for Optimal Boundary and Region Segmentation of Objects in N-D Images. In ICCV 2001.
4. Y. Boykov, V. Kolmogorov: Computing Geodesics and Minimal Surfaces via Graph Cuts. In ICCV 2003.
5. Y. Boykov, V. Kolmogorov: An Experimental Comparison of Min-Cut/Max-Flow Algorithms for Energy Minimization in Vision. In PAMI, 26(9), 2004.
6. M. Bray, P. Kohli, Philip Torr: PoseCut: Simultaneous Segmentation and 3D Pose Estimation of Humans Using Dynamic Graph-Cuts. In ECCV 2006.
7. T. Chan, L. Vese: Active contours without edges. Trans. Image Process., 10(2), 2001.
8. J. Clausen: Branch and Bound Algorithms - Principles and Examples. Parallel Computing in Optimization (1997).
9. D. Cremers, S. Osher, S. Soatto: Kernel Density Estimation and Intrinsic Alignment for Shape Priors in Level Set Segmentation. IJCV 69(3), 2006.
10. D. Cremers, F. Schmidt, F. Barthel: Shape Priors in Variational Image Segmentation: Convexity, Lipschitz Continuity and Globally Optimal Solutions. In CVPR 2008.
11. D. Greig, B. Porteous, A. Seheult: Exact maximum a posteriori estimation for binary images. Journal of the Royal Statistical Society, 51(2), 1989.
12. P. Felzenszwalb: Representation and Detection of Deformable Shapes. PAMI 27(2), 2005.
13. D. Freedman: Effective tracking through tree-search. PAMI, 25(5), 2003.

14. D. Freedman, T. Zhang: Interactive Graph Cut Based Segmentation with Shape Priors. In CVPR 2005.
15. D. Gavrila, V. Philomin: Real-Time Object Detection for "Smart" Vehicles. In ICCV 1999.
16. R. Huang, V. Pavlovic, D. Metaxas: A graphical model framework for coupling MRFs and deformable models. In CVPR 2004.
17. J. Kim, R. Zabih: A Segmentation Algorithm for Contrast-Enhanced Images. ICCV 2003.
18. P. Kohli, P. Torr: Effciently Solving Dynamic Markov Random Fields Using Graph Cuts. In ICCV 2005.
19. V. Kolmogorov, R. Zabih: What Energy Functions Can Be Minimized via Graph Cuts? In ECCV 2002.
20. V. Kolmogorov, Y. Boykov, C. Rother: Applications of Parametric Maxflow in Computer Vision. In ICCV 2007.
21. M. Pawan Kumar, P. Torr, A. Zisserman: OBJ CUT. In CVPR 2005.
22. C. Lampert, M. Blaschko, T. Hofman. Beyond Sliding Windows: Object Localization by Efficient Subwindow Search. In CVPR 2008.
23. B. Leibe, A. Leonardis, B. Schiele: Robust Object Detection with Interleaved Categorization and Segmentation. IJCV 77(3), 2008.
24. M. Leventon, E. Grimson, O. Faugeras: Statistical Shape Influence in Geodesic Active Contours. In CVPR 2000.
25. S. Osher, J. Sethian: Fronts Propagating with Curvature Dependent Speed: Algorithms Based on Hamilton-Jacobi Formulations. J. of Comp. Phys., 79(8), 1988.
26. C. Rother, V. Kolmogorov, A. Blake: "GrabCut": interactive foreground extraction using iterated graph cuts. ACM Trans. Graph. 23(3), 2004.
27. T. Schoenemann, D. Cremers: Globally Optimal Image Segmentation with an Elastic Shape Prior. In ICCV 2007.
28. A. Kemal Sinop, L. Grady: Uninitialized, Globally Optimal, Graph-Based Rectilinear Shape Segmentation - The Opposing Metrics Method. In ICCV 2007.
29. Y. Wang, L. Staib: Boundary Finding with Correspondence Using Statistical Shape Models. In CVPR 1998.

Appendix: Proof of the monotonicity of the lower bound

By definition (see (2) in section 3.2), our bound $L(\Omega)$ equals

$$L(\Omega) = \min_{\mathbf{x}\in 2^{\mathcal{V}}} \left[\min_{\omega\in\Omega} C(\omega) + \sum_{p\in\mathcal{V}} \min_{\omega\in\Omega} F^p(\omega)\cdot x_p + \right.$$

$$\left. \sum_{p\in\mathcal{V}} \min_{\omega\in\Omega} B^p(\omega)\cdot(1-x_p) + \sum_{p,q\in\mathcal{E}} \min_{\omega\in\Omega} P^{pq}(\omega)\cdot|x_p - x_q| \right] , \qquad (9)$$

where $\mathbf{x} \in 2^{\mathcal{V}}$ is the segmentation vector, $\mathcal{V}$ is the set of pixels, ω is the non-local parameter, C, F^p, B^p, P^{pq} are real-valued functions of ω.

We need to prove that if $\Omega_1 \subset \Omega_2$ then $L(\Omega_1) \geq L(\Omega_2)$ (monotonicity).

Proof. Let us denote with $A(\mathbf{x}, \Omega)$ the expression within the outer minimum of (9):

$$A(\mathbf{x}, \Omega) = \left[\min_{\omega \in \Omega} C(\omega) + \sum_{p \in \mathcal{V}} \min_{\omega \in \Omega} F^p(\omega) \cdot x_p + \right.$$

$$\left. \sum_{p \in \mathcal{V}} \min_{\omega \in \Omega} B^p(\omega) \cdot (1 - x_p) + \sum_{p,q \in \mathcal{E}} \min_{\omega \in \Omega} P^{pq}(\omega) \cdot |x_p - x_q| \right] . \tag{10}$$

Then, (9) reformulates as:

$$L(\Omega) = \min_{\mathbf{x} \in 2^{\mathcal{V}}} A(\mathbf{x}, \Omega) . \tag{11}$$

Assume $\Omega_1 \subset \Omega_2$.

Let us prove that for any fixed $\mathbf{x}$, for all pixels p and edges pq, the following holds:

$$\min_{\omega \in \Omega_1} C(\omega) \geq \min_{\omega \in \Omega_2} C(\omega) \tag{12}$$

$$\min_{\omega \in \Omega_1} F^p(\omega) \cdot x_p \geq \min_{\omega \in \Omega_2} F^p(\omega) \cdot x_p \tag{13}$$

$$\min_{\omega \in \Omega_1} B^p(\omega) \cdot (1 - x_p) \geq \min_{\omega \in \Omega_2} B^p(\omega) \cdot (1 - x_p) \tag{14}$$

$$\min_{\omega \in \Omega_1} P^{pq}(\omega) \cdot |x_p - x_q| \geq \min_{\omega \in \Omega_2} P^{pq}(\omega) \cdot |x_p - x_q|. \tag{15}$$

This is because, firstly, all values x_p, $1 - x_p$, and $|x_p - x_q|$ are non-negative (recall that all x_p takes the value of 0 or 1) and, secondly, all minima on the left sides are taken over a subset of the domain of the same minima on the left sides.

Consider for instance (15) for some edge pq. If $|x_p - x_q| = 0$ then (15) is trivial ($0 \geq 0$). Otherwise (if $|x_p - x_q| = 1$) (15) is eqivalent to $\min_{\omega \in \Omega_1} P^{pq}(\omega) \geq \min_{\omega \in \Omega_2} P^{pq}(\omega)$, which is true because the domain on the left lies inside the domain on the right ($\Omega_1 \subset \Omega_2$ by assumption). Same argument holds for all inequalities (12)–(15).

Summing up inequalities (12)–(15) over all pixels p and edges pq and taking into account the definition (10), we get:

$$\forall \mathbf{x} \qquad A(\mathbf{x}, \Omega_1) \geq A(\mathbf{x}, \Omega_2) . \tag{16}$$

I.e. monotonicity holds for any fixed $\mathbf{x}$.

Let $\mathbf{x}_1$ be the segmentation delivering the global optimum of $A(\mathbf{x}, \Omega_1)$: $\mathbf{x}_1 = argmin_{\mathbf{x} \in 2^{\mathcal{V}}} A(\mathbf{x}, \Omega_1)$. Let $\mathbf{x}_2$ be the segmentation delivering the global optimum of $A(\mathbf{x}, \Omega_2)$: $\mathbf{x}_2 = argmin_{\mathbf{x} \in 2^{\mathcal{V}}} A(\mathbf{x}, \Omega_2)$.

Then,

$$L(\Omega_1) = A(\mathbf{x}_1, \Omega_1) \geq A(\mathbf{x}_1, \Omega_2) \geq A(\mathbf{x}_2, \Omega_2) = L(\Omega_2) . \tag{17}$$

Here, the first equality is by the definition of L, A, $\mathbf{x}_1$ (see (11)); the following inequality follows from (16); the next inequality is by the definition of $\mathbf{x}_2$; the last equality is by the definition of L, A, $\mathbf{x}_2$ (see (11)).

Therefore:

$$L(\Omega_1) \geq L(\Omega_2) \ .$$