Graph Cuts and Efficient ND Image Segmentation

2 authors, including:

Gareth Funka-Lea
Siemens Healthineers
61 PUBLICATIONS   **4,165** CITATIONS

Graph Cuts and Efficient N-D Image Segmentation

YURI BOYKOV

Computer Science, University of Western Ontario, London, ON, Canada

yuri@csd.uwo.ca

GARETH FUNKA-LEA

Imaging & Visualization, Siemens Corp. Research, Princeton, NJ, USA

gareth.funka-lea@siemens.com

Abstract. Combinatorial graph cut algorithms have been successfully applied to a wide range of problems in vision and graphics. This paper focusses on possibly the simplest application of graph-cuts: segmentation of objects in image data. Despite its simplicity, this application epitomizes the best features of combinatorial graph cuts methods in vision: global optima, practical efficiency, numerical robustness, ability to fuse a wide range of visual cues and constraints, unrestricted topological properties of segments, and applicability to N-D problems. Graph cuts based approaches to object extraction have also been shown to have interesting connections with earlier segmentation methods such as snakes, geodesic active contours, and level-sets. The segmentation energies optimized by graph cuts combine boundary regularization with region-based properties in the same fashion as Mumford-Shah style functionals. We present motivation and detailed technical description of the basic combinatorial optimization framework for image segmentation via s/t graph cuts. After the general concept of using binary graph cut algorithms for object segmentation was first proposed and tested in Boykov and Jolly (2001), this idea was widely studied in computer vision and graphics communities. We provide links to a large number of known extensions based on iterative parameter re-estimation and learning, multi-scale or hierarchical approaches, narrow bands, and other techniques for demanding photo, video, and medical applications.

1. Introduction

In the last 20 years the computer vision community has produced a number of useful algorithms for localizing object boundaries in images. There are *snakes* (Kass et al., 1988; Cohen, 1991), *active contours* (Isard and Blake, 1998), *geodesic active contours* (Caselles et al., 1997; Yezzi et al., 1997), "shortest path" techniques (Mortensen and Barrett, 1998; Falcão et al., 1998) and many other examples of methods for partitioning an image into two segments: "object" and "background". Each method comes with its own set of features. This paper describes a novel approach to object/background segmentation based on *s-t* graph cuts.[1] This technique from combinatorial optimization has already demonstrated a great potential for solving many problems in vision and graphics[2] (e.g. image restoration (Greig et al., 1989; Boykov et al., 2001), stereo (Roy and Cox, 1998; Ishikawa and Geiger, 1998; Boykov et al., 1998), multi-view reconstruction (Kolmogorov and Zabih, 2002; Vogiatzis et al., 2005; Lempitsky et al., 2006), texture synthesis (Kwatra et al., 2003)). This paper presents a detailed description of the basic framework for efficient object extraction from N-D image data using s/t graph cuts.

	Variational methods (optimization in $\mathcal{R}^{\infty}$)	Combinatorial methods (optimization in $\mathcal{Z}^{n}$)
explicit boundary representation	*Snakes &Balloons* (variational formulations) e.g. (Kass et al., 1988; Cohen, 1991)	*Dynamic Programming* and *"path-based" graph methods* (2D only) (e.g., Amir et al., 1990; Geiger et al., 1995; Mortensen and Barrett, 1998; Falcão et al., 1998; Jermyn and Ishikawa, 1999)
implicit boundary representation	*Level-sets* (e.g., Sethian, 1999; Osher and Fedkiw, 2002; Sapiro, 2001; Osher and Paragios, 2003)	**Combinatorial Graph Cuts** (as originally outlined in Boykov and Jolly, 2001)

Figure 1.   Energy-based approaches to object extraction. Combinatorial s/t graph cut framework for object segmentation was first proposed in Boykov and Jolly (2001) and recently further developed in (Boykov and Kolmogorov, 2003; Blake et al., 2004; Rother et al., 2004; Li et al., 2004; Kirsanov and Gortler, 2004; Li et al., 2005; Wang et al., 2005; Kolmogorov et al., 2005; Kumar et al., 2005; Kolmogorov and Boykov, 2005; Lombaert et al., 2005; Kohli and Torr, 2005; Li et al., 2006; Bray et al., 2006; Kohli and Torr, 2006; Juan and Boykov, 2006). In particular, it was shown that graph cuts can approximate global (Boykov and Kolmogorov, 2003; Kirsanov and Gortler, 2004; Kolmogorov and Boykov, 2005) and local (Boykov et al., 2006) optima of continuous surface functionals. The majority of current graph cut methods for object segmentation use implicit representation of object boundaries. One notable exception is an explicit technique recently shown in Kirsanov and Gortler (2004).

It should be noted that our graph cut approach to object segmentation was preceded by a number of graph-based methods for image "clustering" that use either combinatorial optimization algorithms (Wu and Leahy, 1993; Ishikawa and Geiger, 1998; Felzenszwalb and Huttenlocher, 2004; Veksler, 2000) or approximate spectral analysis techniques, e.g., *normalized cuts* (Shi and Malik, 2000). The main goal of such methods is a completely automatic high-level grouping of image pixels. Typically, this means that they divide an image into "blobs" or "clusters" using only generic cues of coherence or affinity between pixels.[3] Even though we also use a graph-based framework, our work is directly related to a fairly different group of image segmentation methods that includes *snakes* (Kass et al., 1988; Cohen, 1991), *active contours* (Isard and Blake, 1998; Caselles et al., 1997), *intelligent scissors* (Mortensen and Barrett, 1998), *live-wire* (Falcão et al., 1998), and many techniques based on *level-sets* (Sethian, 1999; Osher and Fedkiw, 2002). These methods integrate model-specific visual cues and contextual information in order to accurately delineate particular object(s) of interest.

The major contribution of our work originally outlined in Boykov and Jolly (2001) is that it first demonstrated how to use binary graph cuts to build efficient object extraction tools for N-D applications based on a wide range of model-specific (boundary and region-based) visual cues, contextual information, and useful

topological constraints. Relationship of our combinatorial graph cuts framework for object extraction to previous methods is illustrated in Fig. 1. Interestingly, graph cuts framework uses implicit representation of object boundaries which makes them a discrete counterpart of *level-sets*. Relationship with level-sets is further studied in Boykov et al. (2006).

The effectiveness of formulating the object segmentation problem via binary graph cuts is also demonstrated by a large number of recent publications in computer vision and graphics that directly build upon the basic concept outlined in Boykov and Jolly (2001). They extended our segmentation technique in a number of interesting directions: geometric cues (Boykov and Kolmogorov, 2003; Kolmogorov and Boykov, 2005) (*geo-cuts*), regional cues based on Gaussian mixture models (Blake et al., 2004) for improved interactivity (Rother et al., 2004) (*grab-cuts*), using super-pixels (Li et al., 2004) (*lazy snapping*), integrating high-level contextual information (Kumar et al., 2005) (*obj-cuts*), multi-level and banded methods (Lombaert et al., 2005; Xu et al., 2003; Juan and Boykov, 2006), binary segmentation using stereo cues (Kolmogorov et al., 2005), efficient algorithms for dynamic applications (Kohli and Torr, 2005; Juan and Boykov, 2006) (*flow-* and *cut-recycling*), extraction of moving or foreground objects from video (Li et al., 2005; Wang et al., 2005), simultaneous segmentation of multiple objects (Li et al., 2006), combining segmentation with 3D pose

estimation (Bray et al., 2006), computing segmentation uncertainty (Kohli and Torr, 2006), and methods for solving surface evolution PDEs (Boykov et al., 2006).

1.1. Previous Object Segmentation Methods

There are many methods for object/background segmentation that predate graph cuts. The simplest techniques, such as *region growing* or *split-and-merge* (see Chapter 10 in Haralick and Shapiro (1992)), do not incorporate a clear cost function and are largely based on ad-hoc ideas. Unfortunately, these simple methods are not robust in practice. They are prone to many problems, most notably, to "leaking" through weak spots in object boundaries. Such weak spots can be found in practically any real image. Despite significant problems, methods like "region growing" and "thresholding" are still the most widely known image segmentation techniques. This can be explained by their simplicity and speed. For example, they could be easily run on personal computers available even 15–20 years ago. Currently available PCs allow much more robust segmentation algorithms, most of which rely on optimizing some form of energy function.

Energy Based Object Segmentation.   Energy-based segmentation methods can be distinguished by the type of energy function they use and by the optimization technique for minimizing it. The majority of standard algorithms can be divided into two large groups:

(A) Optimization of a functional defined on a continuous contour or surface
(B) Optimization of a cost function defined on a discrete set of variables

The standard methods in group A formulate segmentation problem in the domain of continuous functions $\mathcal{R}^\infty$. Most of them rely on a variational approach and gradient descent for optimization. The corresponding numerical techniques are based on *finite differences* or on *finite elements*. The segmentation methods in group B either directly formulate the problem as combinatorial optimization in finite dimensional space $\mathcal{Z}^n$ or optimize some discrete energy function whose minima approximates solution of some continuous problem. Examples of methods in groups A and B are given in Fig. 1.

The methods in group (A) include *snakes* (Kass et al., 1988; Cohen, 1991), *region competition* (Zhu and Yuille, 1996), *geodesic active contours* (Caselles et al.,

1997), and other methods based on *level-sets* (Sethian, 1999; Osher and Fedkiw, 2002; Sapiro, 2001; Osher and Paragios, 2003). Typically, continuous surface functionals incorporate various "regional" and "boundary" properties of segments some of which can be geometrically motivated (Caselles et al., 1997; Vasilevskiy and Siddiqi, 2002). In most cases, methods in group (A) use variational optimization techniques that can guarantee to find only a local minima of the corresponding energy functional.

Our new graph cut approach to object extraction belongs to group B. Most of the discrete optimization methods for object segmentation minimize an energy defined over a finite set of integer-valued variables.[4] Such variables are usually associated with graph nodes corresponding to image pixels or control points. To the best of our knowledge, all previous combinatorial methods for object segmentation use discrete variables whose values encode "direction" of a path along the graph. Many path-based methods use Dynamic Programming (DP) to compute optimal paths. For example, (Mortensen and Barrett, 1998) (*intelligent scissors*) and (Falcão et al., 1998) (*live-wire*) use Dijkstra algorithm while (Amir et al., 1990) (*DP-snakes*) use Viterbi algorithm. Note that all path-based methods can naturally encode boundary-based segmentation cues while the incorporation of region properties in segments is less obvious (Jermyn and Ishikawa, 1999; Reese, 1999). In any case, all path-based methods are limited to 2D applications because object boundary in 3D volumes can not be represented by a path.

Global vs. Local Optimization.   Before the s/t graph cuts approach for object segmentation was first presented in Boykov and Jolly (2001), computing a global optima was possible only for some 2D object segmentation methods. In general, global solutions are attractive because of their potentially better stability. For example, imperfections in a globally optimal solution are guaranteed to directly relate to the cost function rather than to a numerical problem during minimization. Thus, global methods can be more reliable and robust. Some versions of active contours (Cohen and Kimmel, 1997), shortest path algorithms (Mortensen and Barrett, 1998; Falcão et al., 1998), *ratio regions* (Cox et al., 1996), and some other segmentation methods (Jermyn and Ishikawa, 1999) compute a globally optimal solution in 2D applications when a segmentation boundary is a 1D curve. To the best of our

knowledge, none of the previous "global" techniques generalize to 3D segmentation problems leaving that domain for variational techniques and for all kinds of ad-hoc approximation methods (e.g. extrapolating segments from slice to slice).

Our s/t graph cuts framework offers a globally optimal object extraction method for N-dimensional images. We describe a fairly general cost function that can include both region and boundary properties of segments and certain types of topological constrains that naturally fit into our global optimization framework.

Recent developments of the graph cuts methods for image segmentation have somewhat blurred the differences between continuous methods (A) that use variational techniques for local optimization and discrete combinatorial methods (B). For example, graph cut approaches have inspired some global optimization techniques for continuous problems (Boykov and Kolmogorov, 2003; Kirsanov and Gortler, 2004; Kolmogorov and Boykov, 2005; Appleton and Talbot, 2006) and a new class of local optimization techniques (Boykov et al., 2006).

1.2. Why Graph Cuts?

The main technical novelty of the object extraction approach presented in this paper is that we formulate segmentation energy over *binary* variables whose values only indicate whether the pixel is inside or outside the object of interest. In contrast to the earlier "path-based" combinatorial methods (see Sec. 1.1), this can be seen as a "region-based" approach to encoding image segments. In fact, the difference between path-based and region-based representations of segments on a discrete graph is analogous to the difference between explicit contour representation, e.g. snakes, and the implicit level-sets approach (see Fig. 1).

Robust and Efficient Optimization in N-D. Numerically, our image segmentation framework relies on powerful graph cut algorithms from combinatorial optimization (Ford and Fulkerson, 1962; Goldberg and Tarjan, 1988; Boykov and Kolmogorov, 2004). We start with a discrete energy formulation and directly solve it with an exact graph-based optimization method.

In contrast, variational methods inherently rely on approximating numerical schemes (e.g. finite differences or finite elements) that must be very carefully designed to insure robustness. Convergence of such numerical methods is an important and non-trivial issue that has to be carefully addressed. Segmentation results generated by two variational techniques using the same energy may depend on their implementation details. In contrast, the discrete optimization approach advocated in this paper is very straightforward and robust numerically. It is also repeatable. Assuming the same energy function, one would always get identical segments even though one can choose from a number of different combinatorial min-cut/max-flow algorithms for computing minimum s-t cuts on graphs (Ford and Fulkerson, 1962; Goldberg and Tarjan, 1988; Boykov and Kolmogorov, 2004).

Recently, (Boykov and Kolmogorov, 2004) studied the practical efficiency of combinatorial min-cut/max-flow algorithms on applications in computer vision. It was shown that some max-flow techniques could solve 2D and 3D segmentation problems in close to real-time using regular PCs. Further significant acceleration was demonstrated for dynamic segmentation problems using *flow-recycling* (Kohli and Torr, 2005) and *cut-recycling* (Juan and Boykov, 2006).[5] Some versions of max-flow/min-cut algorithms can be run on parallel processors (Goldberg and Tarjan, 1988). Parallel implementations are also possible on Graphics Processing Units.[6] While straightforward implementation of graph cuts may require a lot of memory for 3D applications, recent results in Lombaert et al. (2005) showed that multi-level and banded techniques can alleviate the problem.

Relation to Previous Segmentation Methods. Figure 2 demonstrates two simple examples of object segmentation via graph cuts and Example (b) shows that graph cuts include *image thresholding* as a trivial special case. Another special case in (a) shows that, in 2D applications, graph cuts are dual to shortest path methods like *intelligent scissors* (Mortensen and Barrett, 1998) and *live-wire* (Falcão et al., 1998). These techniques find the cheapest subset of edges (a path) that connects two seeds placed on the desired object boundary. In contrast, graph cuts find the cheapest subset of edges (a cut) that separates seeds marking the inside of the object and background regions. It is easy to show that on planar graphs the two algorithms solve two equivalent problems formulated on dual graphs. Unlike *intelligent scissors* or *live-wire*, however, graph cuts can naturally integrate any regional bias in addition to boundary cues and they extend to N-D segmentation problems.
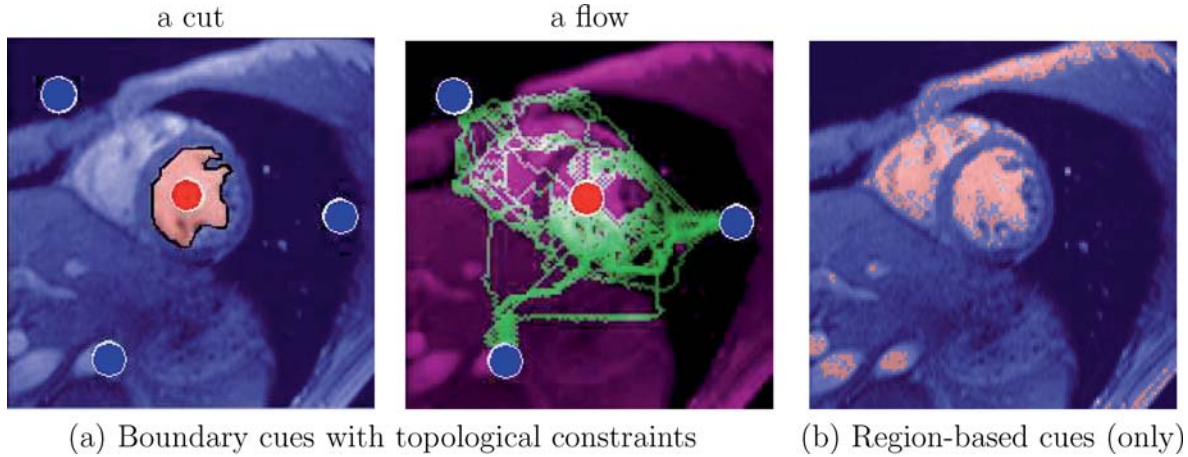
(a) Boundary cues with topological constraints   (b) Region-based cues (only)

Figure 2. Two basic examples of object extraction via s/t graph cuts. Image pixels form a 2D grid graph. In (a) all neighboring pixels are connected by edges, *n-links*, whose capacities depend on intensity differences $|I_p - I_q|$. The seeds, circles, are also connected by edges, *t-links*, to the terminals (source and sink). If t-links are expensive enough, the seeds are hardwired to the terminals thereby providing topological constraints for segmentation. Max-flow algorithms (Ford and Fulkerson, 1962; Boykov and Kolmogorov, 2004) gradually increase the flow of "water" from source s to sink t along the edges (pipes) until enough edges get saturated to form a boundary (cut) separating the terminals. Typically, the bottlenecks are cheap n-links between pixels caused by high intensity differences. While example (a) mostly relies on boundary-based cues encoded by n-links, (b) shows an extreme opposite case of graph cut segmentation using only t-links. In general, t-links encode region-based cues. Assuming that all n-links have zero capacity and each pixel p is connected to both terminals via t-links with capacities $|I_p - A|$ and $|I_p - B|$ then s/t graph cuts will generate a segmentation, shown in red/blue in (b), equivalent to thresholding the image around intensity level $\frac{A+B}{2}$. One of the main strengths of our graph cut approach to segmentation is that it can combine boundary cues, regional cues, and topological constraints in a unified global optimization framework.

Implementations of graph cuts using *push-relabel* (Goldberg and Tarjan, 1988) and algorithms based on pseudo-flows (Hochbaum, 1998; Juan and Boykov, 2006) can show a sequence of cuts as they converge to the global optima. The corresponding image segmentation has dynamics that may resemble methods like *region growing* and *split-and-merge*. Of course, the major difference is that the principles built into combinatorial algorithms for graph cuts guarantee convergence to the exact minima of a clearly defined global energy function.

Recent results in (Boykov and Kolmogorov, 2003; Kirsanov and Gortler, 2004; Kolmogorov and Boykov, 2005) connect graph cuts to a very popular geometric approach to image segmentation, *geodesic active contours* (Caselles et al., 1997). For example, (Boykov and Kolmogorov, 2003) demonstrated that geometric artifacts previously attributed to discrete segmentation methods can be avoided. They show that discrete *cut metrics* on a grid can approximate any continuous Riemannian metric. Then, (Kolmogorov and Boykov, 2005) obtained a complete geometric characterization of continuous surface functionals that graph cut methods can approximate. The result is derived from *submodularity* (Murota, 2003; Kolmogorov and

Zabih, 2004) of discrete energies that network flow algorithms can minimize exactly.

More recently, (Boykov et al., 2006) showed a very strong connection between graph cuts and *level-sets* (Sethian, 1999; Osher and Fedkiw, 2002; Sapiro, 2001; Osher and Paragios, 2003). In particular, (Boykov et al., 2006) developed a novel integral approach to solving surface propagation PDEs based on combinatorial graph cuts algorithms. Such PDEs arise when computing gradient flow evolution of *active contours* which are very widely used in computer vision and medical image analysis. The results in Boykov et al. (2006) suggest that combinatorial graph cuts algorithms can be used as a robust numerical method for an important class of variational problems that was previously addressed mainly with *level-sets*.

Integrating Regions, Boundary, and Shape. Graph cuts optimize discrete energies that combine boundary regularization with regularization of regional properties of segments in the same style as continuous Mumford-Shah functionals (Mumford and Shah, 1989).[7] This work describes in details how to incorporate the most basic types of visual cues into the binary graph cut framework for object extraction. Such

model-specific visual cues allow "fine" localization of the desired object boundaries.

We show that the regional bias can encode any desirable intensity distributions of the object and/or background while the most typical boundary cue is based on the intensity differences between neighboring pixels. Boundary cues based on intensity differences are fairly standard for many applications of graph cuts. For example, this is how *static cues* were encoded for stereo in Boykov et al. (2001). This is also a typical generic cue widely used in image "clustering" (Wu and Leahy, 1993; Ishikawa and Geiger, 1998; Felzenszwalb and Huttenlocher, 2004; Shi and Malik, 2000; Veksler, 2000). In general, graph cuts allow more general "directed" n-links. We demonstrate that model-specific directed n-links avoid some segmentation artifacts.

Note that a number of papers that followed our original publication in Boykov and Jolly (2001) further developed object segmentation cues that graph cuts can encode. For example, (Boykov and Kolmogorov, 2003; Kolmogorov and Boykov, 2005) showed that n-links can be used to encode geometric functionals such as *length/area* and *flux*. Consequently, graph cut segmentation methods can use geometrically motivated cues first introduced by *geodesic active contour* models (Caselles et al., 1997). For example, incorporating *flux* can improve edge alignment (Vasilevskiy and Siddiqi, 2002; Kimmel and Bruckstein, 2003; Kolmogorov and Boykov, 2005; Funka-Lea et al., 2006) and greatly helps to segment thin objects.

A number of recent publications also further developed the use of regional cues. Blake et al. (2004) suggested to use a Gaussian Mixture (GM) model to approximate the regional properties of segments. As demonstrated by *grab-cuts* (Rother et al., 2004), iterative re-estimation of GM intensity models of the object and background help to significantly improve segmentation of colored photographs. A very innovative use of regional cues from stereo was demonstrated for foreground-background segmentation in Kolmogorov et al. (2005).

Binary graph cuts can also integrate some global shape priors. In the simplest case, one can incorporate a regional bias based on values of a signed *distance map* function representing some shape. The same basic idea is widely used in *level-sets*, e.g. (Cremers et al., 2006; Cremers, 2006). Another comparable technique for incorporating shape priors into graph cuts uses *flux* (Kolmogorov and Boykov, 2005). The results in

Boykov et al. (2006) show that a regional bias based on a signed distance map penalizes deviation between the segment and the prior shape according to L_2 distance in the space of contours. Other metrics can be approximated as well. Boykov et al. (2006) use graph cuts to compute a *gradient flow* evolution of a surface by adding an L_2 shape-penalty for a drift from its current position. This additional "current-shape" penalty works as a "breaker" that does not allow the graph cuts to jump too far from the current solution. This iterative graph cut technique can generate gradual motion of a surface consistent with continuous *gradient flow* PDEs (Boykov et al., 2006).

Integrating Topological Constraints.   Our graph cut framework for segmentation is based on an implicit representation of object boundaries. Thus, it allows the same flexible topological properties of segments as in *level-sets* techniques. Our segmentation results may have isolated "objects" which may also contain holes. However, it may be useful to impose some topological constraints reflecting certain high-level contextual information about the object of interest.

Graph cuts can incorporate some types of topological constraints. For example, the hard constraints can indicate some image pixels *a priori* known to be a part of the object or background. We show that topological constraints can be used to reduce the search space of feasible segmentations. Our algorithm also allows very efficient editing of segmentation results, if necessary. The optimal segmentation can be efficiently recomputed if some hard constraints (seeds) are added or removed. Our algorithm efficiently adjusts the current segmentation without recomputing the whole solution from scratch.

Region-based topological constraints naturally fit into graph cuts.[8] These constraints correspond to an infinite cost regional bias that guarantees that some given subsets of pixels should be inside or outside of the object, as shown in Fig. 2(a). Such constraints can restrict the search space for the object of interest in the image. Boykov and Veksler (2006) discusses some non-regional topological constraints that potentially could be used in the graph cut algorithms for object extraction.

Note that *intelligent scissors* (Mortensen and Barrett, 1998) and *live wire* (Falcão et al., 1998) use a boundary-based hard constraints where the user can indicate certain pixels where the segmentation boundary should

pass. The segmentation boundary is then computed as the "shortest path" between the marked pixels according to some energy function based on image gradient. One difficulty with such hard constraints is that the user inputs have to be very accurately positioned at the desired boundary. In contrast, regional hard constraints for graph cuts do not have to be precisely positioned. Moving the seeds around the object of interest (within some limits) does not typically change the segmentation results.

Relation to Markov Random Fields.   Our segmentation method is based on *s-t* graph cut algorithm. Greig et al. (1989) were the first to discover that powerful graph cut algorithms from combinatorial optimization can be useful for computer vision problems. In particular, they showed that graph cuts can be used for restoration of binary images.[9] The problem was formulated as Maximum *A Posterior* estimation of a Markov Random Field (MAP-MRF) that required minimization of posterior energy

$$E(I) = -\lambda \sum_{p \in \mathcal{P}} \ln \Pr(I_p | I^o) + \sum_{\{p,q\} \in \mathcal{N}} \delta_{I_p \neq I_q}, \quad (1)$$

where

$$\delta_{I_p \neq I_q} = \begin{cases} 1 & \text{if } I_p \neq I_q \\ 0 & \text{if } I_p = I_q. \end{cases}$$

is a Kronecker delta representing interaction potential, $I = \{I_p | p \in \mathcal{P}\}$ is an unknown vector of original binary intensities $I_p \in \{0, 1\}$ of image pixels $\mathcal{P}$, vector I^o represents observed binary intensities corrupted by noise, and $\mathcal{N}$ is a set of all pairs of neighboring pixels.

Greig et al. constructed a two terminal graph such that the minimum cost cut of the graph gives a globally optimal binary vector I. Previously, exact minimization of energies like (1) was not possible and such energies were approached mainly with iterative algorithms like simulated annealing. In fact, Greig et al. used their results mainly to show that in practice simulated annealing reaches solutions very far from the global minimum even in very simple binary image restoration examples.

Unfortunately, the graph cut technique in Greig et al. remained unnoticed for almost 10 years mainly because binary image restoration looked very limited as an application. In the late 90's new computer vision techniques appeared that showed how to use *s-t* cut algorithm on graphs for more interesting non-binary problems. Roy and Cox (1998) was the first to use graph cuts to compute multi-camera stereo. Later, (Ishikawa and Geiger, 1998; Boykov et al., 1998) showed that with the right edge weights on a graph similar to (Roy and Cox, 1998) one can minimize a non-binary case of (1) with linear interaction penalties. This graph construction was further generalized to handle arbitrary convex interactions in Ishikawa (2003). Another general case of multi-label energies where the interaction penalty is a *metric* (on the space of labels) was studied in (Boykov et al., 1998, 2001). Their α-expansion algorithm finds provably good approximate solutions by iteratively running min-cut/max-flow algorithms on appropriate graphs. The case of *metric* interactions includes many kinds of "robust" cliques that are frequently used in practice. Later it was shown that α-expansion technique can be also used for non-metric interactions, often loosing optimality guarantees (e.g., Rother et al., 2005).

In this paper we consider a binary segmentation problem where a given object has to be accurately separated from its background. One of our insights is that such a problem can be formulated as a binary (object/background) labeling problem with energy similar to (1). Indeed, binary image restoration energy (1) contains two terms representing "regional" and "boundary" properties. Such a combination looks very appropriate for an object segmentation method. Moreover, a binary energy like (1) can be minimized exactly even in N-dimensional cases using standard *s-t* graph cut algorithms. Technically, our basic method uses a graph construction similar to (Greig et al., 1989). Our main contribution is that our work first demonstrated that global optimization of discrete energy like (1) can be effectively used for accurate object extraction from N-D images and showed how (1) can integrate segmentation cues previously used in snakes and in implict active contour models (e.g. level-sets).

2.   Optimal Object Segmentation via Graph Cuts

In this section we describe our object segmentation framework in detail. Section 2.1 presents our basic ideas relating graph cuts and binary segmentation. Minimal background on *s-t* cuts from graph theory is provided. Section 2.2 formulates our segmentation technique in terms of energy minimization. Our cost function serves as a soft constraint on regional and boundary properties of segments. The synergy of regional and boundary properties is discussed in Section 2.3. In Section 2.4 we introduce hard
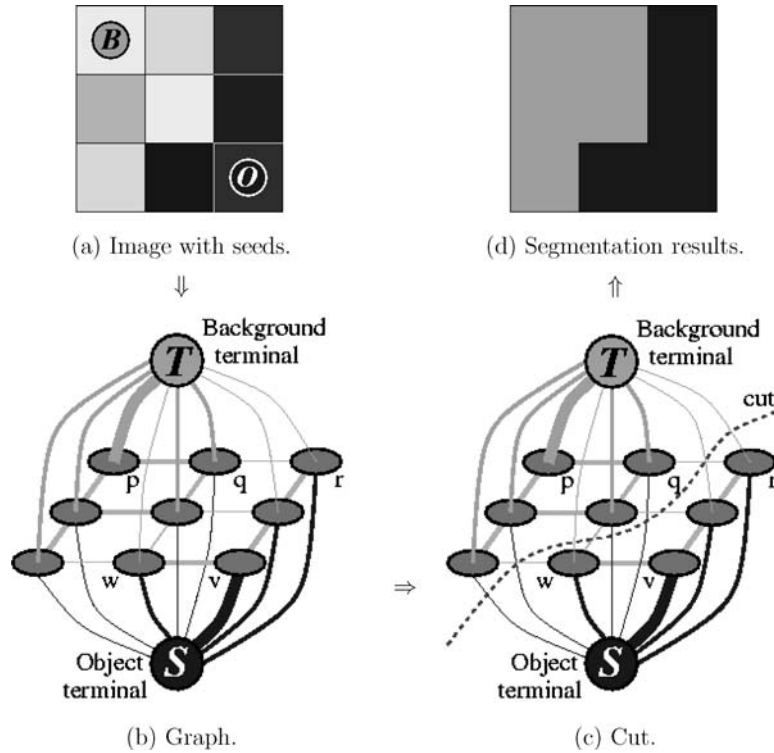
(a) Image with seeds.   (d) Segmentation results.

$\Downarrow$ $\Uparrow$

(b) Graph.   (c) Cut.

Figure 3.   A simple 2D segmentation example for a 3 × 3 image. The seeds are $\mathcal{O} = \{v\}$ and $\mathcal{B} = \{p\}$. The cost of each edge is reflected by the edge's thickness. The boundary term (4) defines the costs of n-links while the regional term (3) defines the costs of t-links. Inexpensive edges are attractive choices for the minimum cost cut. Hard constraints (seeds) (8,9) are implemented via infinity cost t-links. A globally optimal segmentation satisfying hard constraints can be computed efficiently in low-order polynomial time using max-flow/min-cut algorithms on graphs (Ford and Fulkerson, 1962; Goldberg and Tarjan, 1988; Cook et al., 1998).

constraints and show how they can be used to restrict the search space of feasible solutions. Section 2.5 provides implementational details and formally shows that a minimum cost *s-t* cut on an appropriately constructed graph corresponds to a globally optimal solution among all binary segmentations satisfying a given set of hard constraints. Section 2.6 shows an efficient solution for recomputing optimal segments when hard constraints are changed. This feature of our method is very useful for fast object editing, especially in 3D applications. Generalization to directed graphs is discussed in Section 2.7. In some cases directed edges can significantly improve segmentation results.

2.1.   Basic Ideas and Background Information

First, we will introduce some terminology. A graph $\mathcal{G} = \langle \mathcal{V}, \mathcal{E} \rangle$ is defined as a set of nodes or vertices $\mathcal{V}$ and a set of edges $\mathcal{E}$ connecting "neighboring" nodes. For simplicity, we mainly concentrate on undirected

graphs where each pair of connected nodes is described by a single edge $e = \{p, q\} \in \mathcal{E}$.[10] A simple 2D example of an undirected graph that can be used for image segmentation is shown in Fig. 3(b).

The nodes of our graphs represent image pixels or voxels. There are also two specially designated terminal nodes S (*source*) and T (*sink*) that represent "object" and "background" labels. Typically, neighboring pixels are interconnected by edges in a regular grid-like fashion. Edges between pixels are called *n-links* where n stands for "neighbor". Note that a neighborhood system can be arbitrary and may include diagonal or any other kind of n-links. Another type of edges, called *t-links*, are used to connect pixels to terminals. All graph edges $e \in \mathcal{E}$ including n-links and t-links are assigned some nonnegative weight (cost) w_e. In Fig. 3(b) edge costs are shown by the thickness of edges.

An *s-t* cut is a subset of edges $C \subset \mathcal{E}$ such that the terminals S and T become completely separated on the induced graph $\mathcal{G}(C) = \langle \mathcal{V}, \mathcal{E} \backslash C \rangle$. Note that a cut

(see Fig. 3(c)) divides the nodes between the terminals. As illustrated in Fig. 3 (c-d), any cut corresponds to some binary partitioning of an underlying image into "object" and "background" segments. Note that in the simplistic example of Fig. 3 the image is divided into one "object" and one "background" regions. In general, cuts can generate binary segmentation with arbitrary topological properties. Examples in Section 3 illustrate that object and background segments may consist of several isolated connected blobs that also may have holes.

Our goal is to compute the best cut that would give an "optimal" segmentation. In combinatorial optimization the cost of a cut is defined as the sum of the costs of edges that it severs

$$|C| = \sum_{e \in C} w_e$$

Note that severed n-links are located at the segmentation boundary. Thus, their total cost represents the cost of segmentation boundary. On the other hand, severed t-links can represent the regional properties of segments. Thus, a minimum cost cut may correspond to a segmentation with a desirable balance of boundary and regional properties. In Section 2.2 we formulate a precise segmentation energy function that can be encoded via n-links and t-links. Note that infinity cost t-links make it possible to impose hard constraints on segments.

Numerically, our technique is based on a well-known combinatorial optimization fact that a globally minimum *s-t* cut can be computed efficiently in low-order polynomial time (Ford and Fulkerson, 1962; Goldberg and Tarjan, 1988; Cook et al., 1998). The corresponding algorithms work on any graphs. Therefore, our graph cut segmentation method is not restricted to 2D images and computes globally optimal segmentation on volumes of any dimensions. In Section 3 we show a number of 3D examples.

Note that a fast implementation of graph cut algorithms can be an issue in practice. The most straightforward implementations of the standard graph cut algorithms, e.g. max-flow (Ford and Fulkerson, 1962) or push-relabel (Goldberg and Tarjan, 1988), can be slow. The experiments in Boykov and Kolmogorov (2004) compare several well-known "tuned" versions of these standard algorithms in the context of graph based methods in vision. The same paper also describes a new version of the max-flow algorithm that (on typical vision examples) significantly outperformed the standard techniques. Our implementation of the segmentation method of this paper uses the new graph cut algorithm from (Boykov and Kolmogorov, 2004).

2.2. Segmentation Energy

Consider an arbitrary set of data elements (pixels or voxels) $\mathcal{P}$ and some neighborhood system represented by a set $\mathcal{N}$ of all (unordered) pairs $\{p, q\}$ of neighboring elements in $\mathcal{P}$. For example, $\mathcal{P}$ can contain pixels (or voxels) in a 2D (or 3D) grid and $\mathcal{N}$ can contain all unordered pairs of neighboring pixels (voxels) under a standard 8- (or 26-) neighborhood system. Let $A = (A_1, \ldots, A_p, \ldots, A_{|\mathcal{P}|})$ be a binary vector whose components A_p specify *assignments* to pixels p in $\mathcal{P}$. Each A_p can be either "obj" or "bkg" (abbreviations of "object" and "background"). Vector A defines a segmentation. Then, the soft constraints that we impose on boundary and region properties of A are described by the cost function

$$E(A) = \lambda \cdot R(A) + B(A) \qquad (2)$$

where

$$R(A) = \sum_{p \in \mathcal{P}} R_p(A_p) \quad (\textit{regional term}) \qquad (3)$$

$$B(A) = \sum_{\{p,q\} \in \mathcal{N}} B_{p,q} \cdot \delta_{A_p \neq A_q} \quad (\textit{boundary term})$$

$$\qquad (4)$$

and

$$\delta_{A_p \neq A_q} = \begin{cases} 1 & \text{if } A_p \neq A_q \\ 0 & \text{if } A_p = A_q. \end{cases}$$

The coefficient $\lambda \geq 0$ in (2) specifies a relative importance of the region properties term $R(A)$ versus the boundary properties term $B(A)$. The regional term $R(A)$ assumes that the individual penalties for assigning pixel p to "object" and "background", correspondingly $R_p(\text{"obj"})$ and $R_p(\text{"bkg"})$, are given. For example, $R_p(\cdot)$ may reflect on how the intensity of pixel p fits into given intensity models (e.g. histograms) of the object and background

$$R_p(\text{"obj"}) = -\ln \Pr(I_p | \text{"obj"}) \qquad (5)$$

$$R_p(\text{"bkg"}) = -\ln \Pr(I_p | \text{"bkg"}) \qquad (6)$$

This use of negative log-likelihoods is motivated by the MAP-MRF formulations in (Greig et al., 1989; Boykov et al., 2001).
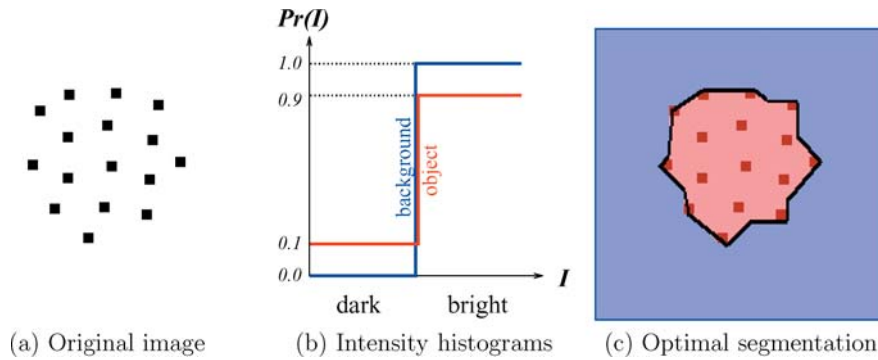
Figure 4. Synthetic Gestalt example. The optimal object segment (red tinted area in (c)) finds a balance between "region" and "boundary" terms in (2). The solution is computed using graph cuts as explained in Section 2.5. Some ruggedness of the segmentation boundary is due to metrication artifacts that can be realized by graph cuts in textureless regions. Such artifacts can be eliminated using results in Boykov and Kolmogorov (2003).

The term $B(A)$ comprises the "boundary" properties of segmentation A. Coefficient $B_{p,q} \geq 0$ should be interpreted as a penalty for a discontinuity between p and q. Normally, $B_{p,q}$ is large when pixels p and q are similar (e.g. in their intensity) and $B_{p,q}$ is close to zero when the two are very different. The penalty $B_{p,q}$ can also decrease as a function of distance between p and q. Costs $B_{p,q}$ may be based on local intensity gradient, Laplacian zero-crossing, gradient direction, geometric (Boykov and Kolmogorov, 2003; Kolmogorov and Boykov, 2005) or other criteria. Often, it is sufficient to set the boundary penalties from a simple function like

$$B_{p,q} \propto \exp\left(-\frac{(I_p - I_q)^2}{2\sigma^2}\right) \cdot \frac{1}{dist(p, q)}. \quad (7)$$

This function penalizes a lot for discontinuities between pixels of similar intensities when $|I_p - I_q| < \sigma$. However, if pixels are very different, $|I_p - I_q| > \sigma$, then the penalty is small. Intuitively, this function corresponds to the distribution of noise among neighboring pixels of an image. Thus, σ can be estimated as "camera noise".

2.3.   "Region" vs. "Boundary"

A simple example of Fig. 4 illustrates some interesting properties of our cost function (2). The object of interest is a cluster of black dots in Fig. 4(a) that we would like to segment as one blob. We combine boundary and region terms (3,4) taking $\lambda > 0$ in (2). The penalty for

discontinuity in the boundary cost is

$$B_{p,q} = \begin{cases} 1 & \text{if } I_p = I_q \\ 0.2 & \text{if } I_p \neq I_q \end{cases}$$

To describe regional properties of segments we use *a priori* known intensity histograms (Fig. 4(b)). Note that the background histogram concentrates exclusively on bright values while the object allows dark intensities observed in the dots. If these histograms are used in (5,6) then we get the following regional penalties $R_p(A_p)$ for pixels of different intensities.

I_p	R_p("obj")	R_p("bkg")
dark	2.3	$+\infty$
bright	0.1	0

The optimal segmentation in Fig. 4(c) finds a balance between the regional and the boundary term of energy (2). Individually, bright pixels slightly prefer to stay with the background (see table above). However, spatial coherence term (4) forces some of them to agree with nearby dark dots which have a strong bias towards the object label (see table).

2.4.   Hard Constraints

In the simple example of Fig. 4 the regional properties of the object of interest are distinct enough to segment it from the background. In real examples, however,
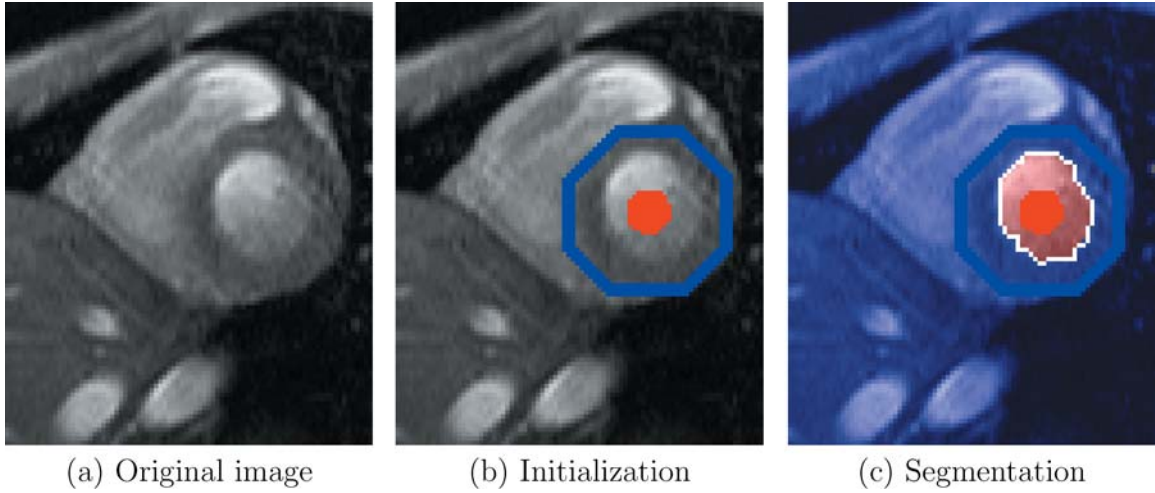
(a) Original image   (b) Initialization   (c) Segmentation

Figure 5. Automatic segmentation of cardiac MR data. Initialization in (b) is based on hard constraints that can be placed automatically using simple template matching. Then, graph cuts accurately localize object boundaries in (c).

objects may not have sufficiently distinct regional properties. In such cases it becomes necessary to further constraint the search space of possible solutions before computing an optimal one. We propose topological (hard) constraints as an important source of "high level" contextual information about the object of interest in real images.

Assume that $\mathcal{O}$ and $\mathcal{B}$ denote the subsets of pixels *a priori* known to be a part of "object" and "background", correspondingly. Naturally, the subsets $\mathcal{O} \subset \mathcal{P}$ and $\mathcal{B} \subset \mathcal{P}$ are such that $\mathcal{O} \cap \mathcal{B} = \emptyset$. For example, consider sets $\mathcal{O}$ (red pixels) and $\mathcal{B}$ (blue pixels) in Fig. 5(b). Our goal is to compute the global minimum of (2) among all segmentations A satisfying hard constraints

$$\forall p \in \mathcal{O}: \quad A_p = \text{"obj"} \tag{8}$$

$$\forall p \in \mathcal{B}: \quad A_p = \text{"bkg"}. \tag{9}$$

Figure 5(c) shows an example of an optimal segmentation satisfying the hard constraints in (b). Throughout this paper we use red tint to display object segments and blue tint for background.

Ability to incorporate hard constraints (8,9) is one of the most interesting features of our segmentation method. There is a lot of flexibility in how these hard constraints can be used to adjust the algorithm for different tasks. The hard constraints can be used to initialize the algorithm and to edit the results. The hard constraints can be set either automatically or manually depending on an application. Manually controlled

seeds is the most likely way to enter hard constraints in many generic applications, e.g. photo-editing shop. Manual seeds are also useful for editing segments (see Section 2.6) when initial segmentation results require corrections. On the other hand, automatically set hard constraints can be used to initialize the algorithm in highly specialized applications such as organ segmentation from medical images or volumes.

For example, consider a medical application where one should segment the blood pool of a left ventricle (one of the heart chambers) captured in an MR image of Fig. 5(a). A simple template matching can roughly localize the left ventricle in the image, e.g. using its known circular shape. The hard constraints in Fig. 5(b) can be set automatically as soon as a rough position of the blood pool is known. Then, our graph cut technique can accurately localize the boundary of the blood pool in Fig. 5(c).

Note that the hard constraints in Fig. 5 restrict the set of feasible cuts to closed contours in a band between the blue and red seeds. It is possible to show that the minimum cost cut in that band can be interpreted as the shortest length path on a "dual" graph. In fact, our graph cut approach can be seen as a generalization of path-based segmentation techniques in (Geiger et al., 1995; Mortensen and Barrett, 1998; Falcão et al., 1998). These methods are intrinsically 2D while our graph cut approach can compute optimal segmentation boundary for N-dimensional cases as well.

The band in Fig. 5 also restricts the area where the actual computation takes place. It is enough to build a
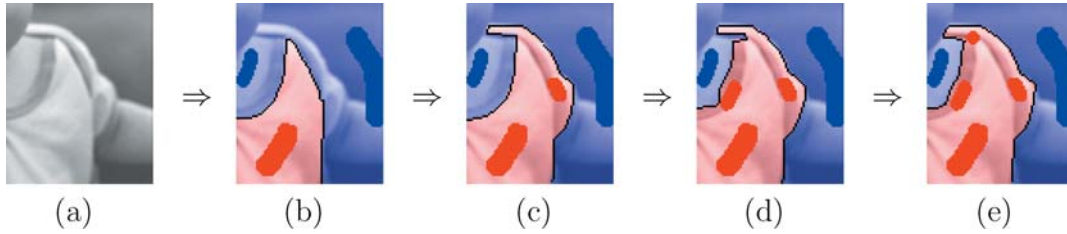
Figure 6. Editing segments by adding hard constraints (seeds). A fragment of an original photo is shown in (a). Initial seeds and segmentation are shown in (b). The results in (c,d,e) illustrate changes in optimal segmentation as new hard constraint are successively added. The computation time for consecutive corrections in (c,d,e) is marginal compared to time for initial results in (b).

graph only in the area of the band since max-flow/min-cut algorithms would not access any other nodes.

An example in Fig. 6 further illustrates how hard constraints (8,9) affect the search space of feasible solutions. Naturally, adding hard constraints helps to segment desirable object as unacceptable solutions are discarded and the search space reduces. Hard constraints placed in one 2D slice may be enough to properly constrain the search space of segmentations in a 3D volume, e.g. in Fig. 10.

Note that in a practical implementation of our method it is possible to make a double use of the seeds. First of all, they can constrain the search space as discussed above. In addition, we can use intensities of pixels (voxels) marked as seeds to learn the histograms for "object" and "background" intensity distributions: Pr(I|"obj") and Pr(I|"bkg") in (5,6). Other ideas for initializing intensity distributions are studies in (Blake et al., 2004; Rother et al., 2004).

2.5. Optimal Solution Via Graph Cuts

In this section we provide algorithmic details about our segmentation technique. The general work-flow is shown in Fig. 3. We describe the details of the corresponding graph construction and prove that the minimum cost cut gives an optimal segmentation for energy (2) and hard constraints (8,9).

To segment a given image we create a graph $\mathcal{G} = \langle \mathcal{V}, \mathcal{E} \rangle$ with nodes corresponding to pixels $p \in \mathcal{P}$ of the image. There are two additional nodes: an "object" terminal (a source S) and a "background" terminal (a sink T). Therefore,

$$\mathcal{V} = \mathcal{P} \cup \{S, T\}.$$

The set of edges $\mathcal{E}$ consists of two types of undirected edges: *n-links* (neighborhood links) and *t-links*

(terminal links). Each pixel p has two t-links $\{p, S\}$ and $\{p, T\}$ connecting it to each terminal. Each pair of neighboring pixels $\{p, q\}$ in $\mathcal{N}$ is connected by an n-link. Without any ambiguity, an n-link connecting a pair of neighbors p and q is also denoted by $\{p, q\}$. Therefore,

$$\mathcal{E} = \mathcal{N} \bigcup_{p \in \mathcal{P}} \{\{p, S\}, \{p, T\}\}.$$

The following table gives weights of edges in $\mathcal{E}$

edge	weight (cost)	for
$\{p, q\}$	$B_{p,q}$	$\{p, q\} \in \mathcal{N}$
$\{p, S\}$	$\lambda \cdot R_p(\text{"bkg"})$	$p \in \mathcal{P}, \ p \notin \mathcal{O} \cup \mathcal{B}$
	K	$p \in \mathcal{O}$
	0	$p \in \mathcal{B}$
$\{p, T\}$	$\lambda \cdot R_p(\text{"obj"})$	$p \in \mathcal{P}, \ p \notin \mathcal{O} \cup \mathcal{B}$
	0	$p \in \mathcal{O}$
	K	$p \in \mathcal{B}$

where

$$K = 1 + \max_{p \in \mathcal{P}} \sum_{q: \{p,q\} \in \mathcal{N}} B_{p,q}.$$

The graph $\mathcal{G}$ is now completely defined. We draw the segmentation boundary between the object and the background by finding the minimum cost cut on the graph $\mathcal{G}$. The minimum cost cut $\hat{C}$ on $\mathcal{G}$ can be computed exactly in polynomial time via algorithms for two terminal graph cuts (see Section 2.1) assuming that the edge weights specified in the table above are non-negative.[11]

Below we state exactly how the minimum cut $\hat{C}$ defines a segmentation $\hat{A}$ and prove this segmentation is optimal. We need one technical lemma. Assume that $\mathcal{F}$ denotes a set of all *feasible* cuts C on graph $\mathcal{G}$ such that

- C severs exactly one t-link at each p
- $\{p, q\} \in C$ iff p and q are t-linked to different terminals
- if $p \in \mathcal{O}$ then $\{p, T\} \in C$
- if $p \in \mathcal{B}$ then $\{p, S\} \in C$.

Lemma 1. *The minimum cut on $\mathcal{G}$ is feasible, i.e. $\hat{C} \in \mathcal{F}$.*

Proof: $\hat{C}$ severs at least one t-link at each pixel since it is a cut that separates the terminals. On the other hand, it can not sever both t-links. In such a case it would not be minimal since one of the t-links could be returned. Similarly, a minimum cut should sever an n-link $\{p, q\}$ if p and q are connected to the opposite terminals just because any cut must separate the terminals. If p and q are connected to the same terminal then $\hat{C}$ should not sever unnecessary n-link $\{p, q\}$ due to its minimality. The last two properties are true for $\hat{C}$ because the constant K is larger than the sum of all n-links costs for any given pixel p. For example, if $p \in \mathcal{O}$ and $\hat{C}$ severs $\{p, S\}$ (costs K) then we would construct a smaller cost cut by restoring $\{p, S\}$ and severing all n-links from p (costs less then K) as well as the opposite t-link $\{p, T\}$ (zero cost). $\square$

For any feasible cut $C \in \mathcal{F}$ we can define a unique corresponding segmentation $A(C)$ such that

$$A_p(C) = \begin{cases} \text{``obj''}, & \text{if } \{p, T\} \in C \\ \text{``bkg''}, & \text{if } \{p, S\} \in C. \end{cases} \qquad (10)$$

The definition above is coherent since any feasible cut severs exactly one of the two t-links at each pixel p. The lemma showed that a minimum cut $\hat{C}$ is feasible. Thus, we can define a corresponding segmentation $\hat{A} = A(\hat{C})$. The next theorem completes the description of our algorithm.

Theorem 1. *The segmentation $\hat{A} = A(\hat{C})$ defined by the minimum cut $\hat{C}$ as in* (10) *minimizes* (2) *among all segmentations satisfying constraints* (8, 9).

Proof: Using the table of edge weights, definition of feasible cuts $\mathcal{F}$, and Eq. (10) one can show that a cost of any $C \in \mathcal{F}$ is

$$|C| = \sum_{p \notin \mathcal{O} \cup \mathcal{B}} \lambda \cdot R_p(A_p(C)) + \sum_{\{p,q\} \in \mathcal{N}} B_{p,q} \cdot \delta_{A_p(C) \neq A_q(C)}$$

$$= E(A(C)) - \sum_{p \in \mathcal{O}} \lambda \cdot R_p(\text{``obj''}) - \sum_{p \in \mathcal{B}} \lambda \cdot R_p(\text{``bkg''}).$$

Therefore, $|C| = E(A(C)) - const$. Note that for any $C \in \mathcal{F}$ assignment $A(C)$ satisfies constraints (8,9). In fact, Eq. (10) gives a one-to-one correspondence between the set of all feasible cuts in $\mathcal{F}$ and the set $\mathcal{H}$ of all assignments A that satisfy hard constraints (8,9). Then,

$$E(\hat{A}) = |\hat{C}| + const = \min_{C \in \mathcal{F}} |C| + const$$

$$= \min_{C \in \mathcal{F}} E(A(C)) = \min_{A \in \mathcal{H}} E(A)$$

and the theorem is proved. $\square$

2.6. Fast Editing of Segments

In practice, no segmentation algorithm can guarantee 100% accuracy. Thus, it is convenient to have a simple way to correct segments if necessary. Within our framework segment editing can be done by placing additional hard constraints (seeds) in incorrectly segmented image areas.[12] Figure 6 shows one example of editing in a photo-shop context.

In fact, our technique can efficiently recompute a new globally optimal solution that satisfies additional constraints by adjusting a current segmentation. This "fast editing" feature is very useful in practical applications. In particular, this is very important for fast editing of objects in 3D cases. Indeed, initial segmentation may take 5-30 seconds or more depending on volume size. Corrections, however, can be computed within a second.

Below we describe an efficient method to recompute an optimal solution when hard constraints are changed. We assume that a max-flow algorithm (see (Ford and Fulkerson, 1962; Cook et al., 1998)) is used to determine the minimum cut on $\mathcal{G}$. The max-flow algorithm gradually increases the flow sent from the source S to the sink T along the edges in $\mathcal{G}$ given their capacities (weights). Upon termination the maximum flow saturates the graph. The saturated edges correspond to the minimum cost cut on $\mathcal{G}$ giving us an optimal segmentation.
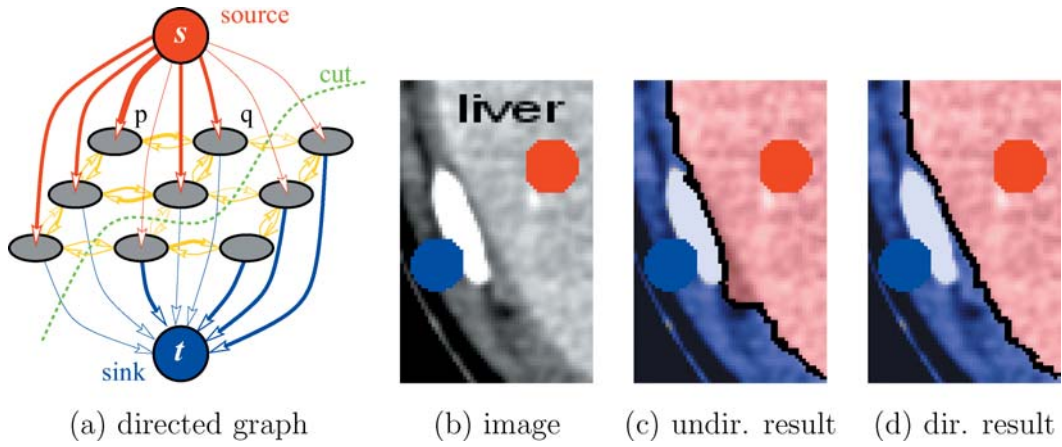
Figure 7. Segmentation via cuts on a directed graph. Compare the results on an undirected graph (c) with the results on a directed graph in (d).

Assume now that an optimal segmentation is already computed for some initial set of seeds. A user adds a new "object" seed to pixel p that was not previously assigned any seed. We need to change the costs for two t-links at p

t-link	initial cost	new cost
$\{p, S\}$	$\lambda R_p(\text{"bkg"})$	K
$\{p, T\}$	$\lambda R_p(\text{"obj"})$	0

and then compute the maximum flow (minimum cut) on the new graph. In fact, we can start from the flow found at the end of initial computation. The only problem is that reassignment of edge weights as above reduces capacities of some edges. If there is a flow through such an edge then we may break the flow consistency. Increasing an edge capacity, on the other hand, is never a problem. Then, we can solve the problem as follows.

To accommodate the new "object" seed at pixel p we increase the t-links weights according to the table

t-link	initial cost	add	new cost
$\{p, S\}$	$\lambda R_p(\text{"bkg"})$	$K + \lambda R_p(\text{"obj"})$	$K + c_p$
$\{p, T\}$	$\lambda R_p(\text{"obj"})$	$\lambda R_p(\text{"bkg"})$	c_p

These new costs are consistent with the edge weight table for pixels in $\mathcal{O}$ since the extra constant c_p at both t-links of a pixel does not change the optimal cut.[13]

Then, a maximum flow (minimum cut) on a new graph can be efficiently obtained starting from the previous flow without recomputing the whole solution from scratch.

Note that the same trick can be done to adjust the segmentation when a new "background" seed is added or when a seed is deleted. One has to figure the right amounts that have to be added to the costs of two t-links at the corresponding pixel. The new costs should be consistent with the edge weight table plus or minus the same constant.

2.7. Using Directed Edges

For simplicity, we previously concentrated on the case of *undirected* graphs as in Fig. 3. In fact, the majority of *s-t* cut algorithms from combinatorial optimization can be applied to *directed* graphs as well. Figure 7(a) gives one example of such a graph where each pair of neighboring nodes is connected by two directed edges (p, q) and (q, p) with distinct weights $w_{(p,q)}$ and $w_{(q,p)}$. If a cut separates two neighboring nodes p and q so that p is connected to the source while q is connected to the sink then the cost of the cut includes $w_{(p,q)}$ while $w_{(q,p)}$ is ignored. Vise versa, if q is connected to the source and p to the sink then the cost of the cut includes only $w_{(q,p)}$.

In certain cases one can take advantage of such directed costs to obtain more accurate object boundaries. For example, compare two segmentations in Fig. 7(c,d) obtained on a medical image in (b) using the same set of constraints. A relatively bright object of interest on the right (liver) is separated from a small bright blob on

the left (bone) by a thin layer of a relatively dark muscle tissue. The contrast between the bone and the muscle is much better then the contrast between the muscle and the liver. Thus, according to (7) the standard "undirected" cost of edges between the bone and the muscle is much cheaper than the cost of edges between the muscle and the liver. Consequently, an optimal cut on an undirected graph produces segmentation in Fig. 7(c) that sticks to the bone instead of following the actual liver boundary. Correcting this error may require segment editing as in Section 2.6.

Alternatively, directed graphs can automatically distinguish between the incorrect boundary in Fig. 7(c) and the desirable one in (d). The key observation is that the weights of directed edges (p, q) can depend on a sign of intensity difference $(I_p - I_q)$. In contrast, weights of undirected edges should be symmetric with respect to its end points and could depend only on the absolute value $|I_p - I_q|$ as in (7). Note that the object boundary that stuck to the bone in (c) separates darker tissue (muscle) in the object segment from brighter tissue (bone) in the background. On the other hand, the correct object boundary in (d) goes from brighter tissue (liver) in the object to darker tissue (muscle) in the background. Note that directed edge weights

$$
w_{(p,q)} = \begin{cases} 1 & \text{if } I_p \leq I_q \\ \exp\left(-\dfrac{(I_p - I_q)^2}{2\sigma^2}\right) & \text{if } I_p > I_q \end{cases}
$$

would specifically encourage cuts from brighter tissue in the object to darker tissue in the background. The results in Fig. 7(d) show optimal segmentation on a directed graph using such edge weights.

Formally, a "directed" version of segmentation energy (2) can be presented as follows. The neighborhood system $\mathcal{N}$ should include all *ordered* pairs (p, q) of neighboring pixels. Then the boundary term of energy (2) is

$$
B(A) = \sum_{(p,q)\in\mathcal{N}} B_{(p,q)} \cdot \delta_{A_p=\text{"obj"}, A_q=\text{"bkg"}} \quad (11)
$$

Note that $B_{(p,q)}$ and $B_{(q,p)}$ are two distinct (directed) discontinuity penalties for cases when $p \in$ "object", $q \in$ "background" and when $p \in$ "background", $q \in$ "object". Straightforward generalization of the technical results in Section 2.5 shows that energy (2) with boundary term (11) can be minimized via s-t graph cuts using directed n-links $w_{(p,q)} = B_{(p,q)} \geq 0$.

In fact, *non-negativity* of directed penalties $B_{(p,q)} \geq 0$ is not an essential limitation of our general s/t graph cut framework for object extraction. Vladimir Kolmogorov has pointed out to us that it generalizes to *submodular* penalties

$$
B_{(p,q)} + B_{(q,p)} \geq 0
$$

In the context of object segmentation, such directed *submodular* penalties were recently studied in Kolmogorov and Boykov (2005) where "directedness" of the boundary cost (11) is geometrically characterized in terms of *flux* of a vector field. Consistently with some variational methods (Vasilevskiy and Siddiqi, 2002; Kimmel and Bruckstein, 2003) from group A in Section 1.1, (Kolmogorov and Boykov, 2005) demonstrates that optimization of *flux* helps graph cuts to better align with object boundaries and to segment thin structures.

3. Experimental Results

We demonstrate our general-purpose segmentation method on several generic examples including photo/video editing and medical data processing. The main goal is to prove the concept of object extraction via s/t graph cuts proposed in our work. We show original data and segments generated by our technique for a given set of hard constraints. Our actual implementation allows a user to enter hard constraints (seeds) via mouse operated brush of red (for object) or blue (for background) color. We present segmentation results in different formats depending on what is more appropriate in each case.

Note that we used simple 4-neighborhood systems in 2D examples and 26-neighborhood system in 3D examples. All running times are given for 1.4 GHz Pentium III. Our implementation is based on a new "max-flow" algorithm from (Boykov and Kolmogorov, 2004).

3.1. Photo and Video Editing

The results in this Section are obtained with $\lambda = 0$, that is without the regional term (3). There are examples where some objects in real images have distinct intensity distributions that may help to segment them. However, a segmentation algorithm in a general-purpose photo/video editor can not count on
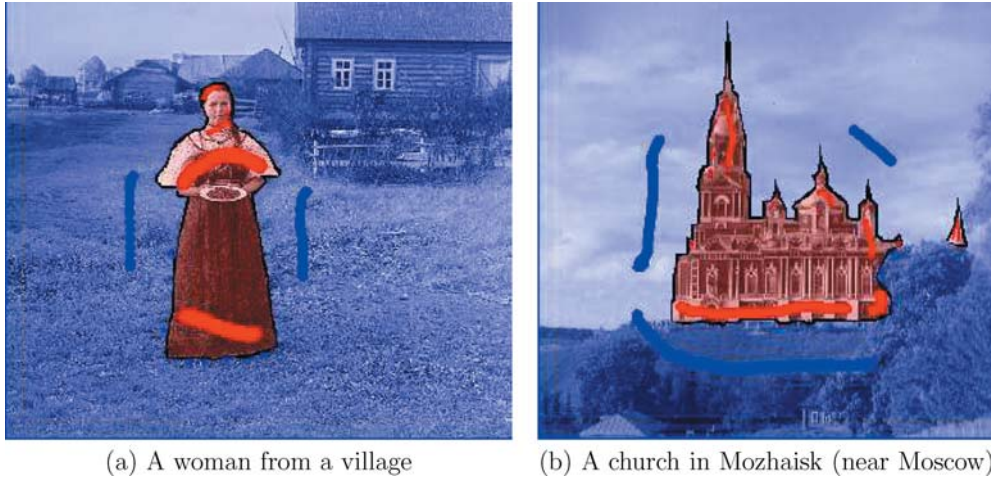
(a) A woman from a village   (b) A church in Mozhaisk (near Moscow)

Figure 8.   Segmentation of photographs (early 20th century). Initial segmentation for a given set of hard constraints (seeds) takes less than a second for most 2D images (up to 1000 × 1000). Correcting seeds are incorporated in the blink of an eye. Thus, the speed of our method for photo editing mainly depends on time for placing seeds. An average user will not need much time to enter seeds in (a) and (b).

any specific regional properties. Indeed, objects of interest are determined by random users and can not be predicted. On the other hand, we can still use the boundary term (4) with the discontinuity penalty (7). Most objects of interest will have borders along high contrast boundaries inside images. Hard constraints entered as seeds can bring "high level" contextual information.

In Fig. 8 we show segmentation results for some photographs.[14] The user can start with a few "object" and "background" seeds loosely positioned inside and, correspondingly, outside the object(s) of interest. By reviewing the results of initial segmentation the user may observe that some areas are segmented incorrectly. Then, one can edit segments as described in Section 2.6.

Naturally, the hope is that the method can quickly identify the right object boundaries with minimum user interaction. The algorithm would not be practical if the user has to add correcting seeds until the whole image is basically manually segmented. The performance of the algorithm can be judged by the amount of constraints/seeds that had to be placed. For that reason, the results in Fig. 8 are shown with the corresponding seeds.

In Fig. 9 we segmented moving cars in a video sequence. The sequence of 21 video frames (255 × 189) was treated as a single 3D volume. The necessary seeds were entered in a simple 3D interface where we could browse through individual 2D slices (frames) of the volume. The whole volume is segmented based on hard constraints placed in just a few frames. Note that cor-

recting seeds in one frame can fix imperfections in many adjacent frames. Each car was segmented in an independent experiment. Segmentation of the whole volume of 21 frames can be obtained by placing hard constraints in only a few of the frames.

The initial segmentation might take from 2–3 seconds on smaller volumes (200 × 200 × 10) to a few minutes on bigger ones (512 × 512 × 50). Thus, efficient editing described in Section 2.6 is crucial for 3D applications. Usually, the new seeds are incorporated in a few seconds even on bigger volumes.

3.2.   Medical Images and Volumes

Figures 10, 11, and 12 show segmentation results that we obtained on a 3D medical volumes. Each object was segmented in 10 to 30 seconds. In the examples of Figs. 10 and 12 the objects were extracted from 3D volumes after entering seeds in only one slice shown in (a). In Fig. 11 some correcting seeds were added in a couple of slices in addition to initial hard constraints shown in (a). Note that the object in Fig. 10 consist of two isolated blobs one of which has a hole. This clearly demonstrates that our method does not restrict topological properties of segments.

As in Section 3.1 we did not use regional term (3) for the experiments in Figs. 10, 11 and 12 as it was not useful. In some applications, however, the regional term may significantly simplify, if not completrely automate (Kolmogorov et al., 2005), the segmentation process.

Figure 9. Segmentation of a video sequence (21 frames). Initial segmentation takes 3–5 seconds while correcting seeds are incorporated within a second. The car in the center is the simplest to segment. It is enough to place hard constraints (seeds) in one frame. The car on the left required some editing due to low contrast and seeds were placed in 3 frames. The bus on the right moves behind a tree and its segmentation required seeds in 4-5 frames.
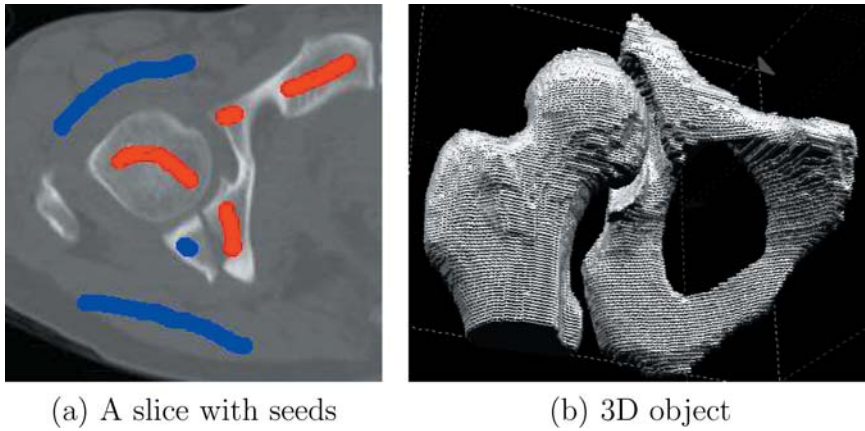

(a) A slice with seeds   (b) 3D object

Figure 10. Segmentation of bones in a CT volume [256x256x119].

In Fig. 13 we demonstrate segmentation on 3D kidney MR data that benefited from regional term (3). We segmented out cortex, medulla, and collecting system of a kidney in three consecutive steps. First, the whole kidney is separated from the background and the latter is cropped. The remaining pixels belong to three distinct types of kidney tissue (cortex, medulla, or collecting system) with identifiable regional properties. At this point it becomes useful to engage the regional term (3) of energy.

The results in Fig. 13 are shown without seeds since the process involved three different segmentations. Using regional bias allows to get 3D segmentation results by entering only a few seeds in one slice. Initial optimal segments are computed in 1–10 seconds and minor correction can be incorporated in less then a second. This example also demonstrates unrestricted topological properties of our segments. Fully automatic segmentation of kidney might be possible with more sophisticated models for regional.
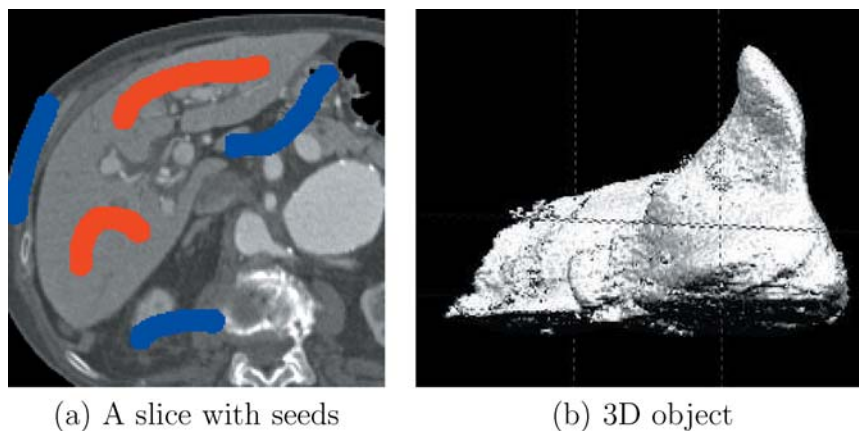
(a) A slice with seeds   (b) 3D object

Figure 11.   Segmentation of liver in a CT volume [170x170x144].

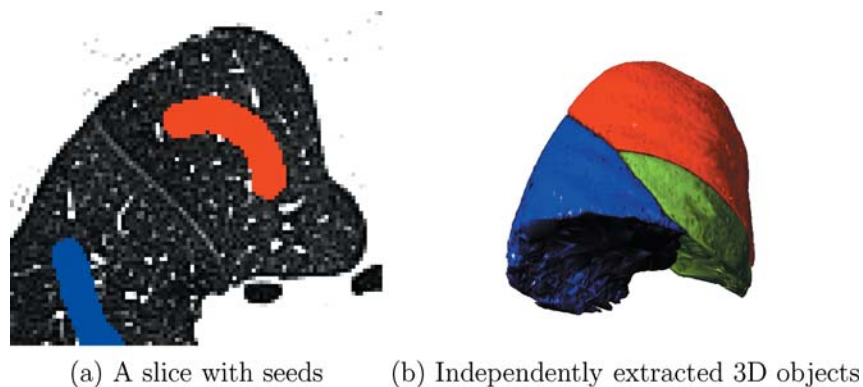(a) A slice with seeds   (b) Independently extracted 3D objects

Figure 12.   Segmentation of lung lobes in a CT volume [205x165x253].

4. Discussion

We presented a novel framework for extracting objects from images/volumes. In many ways our graph-cut approach to object extraction can be seen as a unifying framework for segmentation that combines many good features of the previous methods like snakes, active contours, and level sets while providing efficient and robust global optimization applicable to N-D problems.

The graph cuts framework is very flexible with initialization. It can be based either on topological constraints reflecting high-level contextual information (Boykov and Jolly, 2001), or on various visual cues like object/background color distributions (Boykov and Jolly, 2001; Blake et al., 2004; Rother et al., 2004), texture, or even stereo cues (Kolmogorov et al., 2005). The method does not have to have an initial contour/surface, but it may take advantage of some shape prior (Kolmogorov and Boykov, 2005; Boykov et al., 2006) or of some initial approximate guess (Juan and Boykov, 2006), if available. The algorithm can also be accelerated in dynamic applications where previously segmented image frame is similar to the new frame (Kohli and Torr, 2005). Similar to level-set based methods, graph cuts use implicit boundary representation and the topological properties of the recovered segments are unrestricted, unless additional constraints are imposed (Boykov and Veksler, 2006). In addition, our algorithm allows effective editing of segments if necessary.

To the best of our knowledge, our graph cut approach is the first global optimization object extraction technique that extends to N-dimensional images. The underlying numerical optimization scheme is straightforward and robust. It uses exact min-cut/max-flow
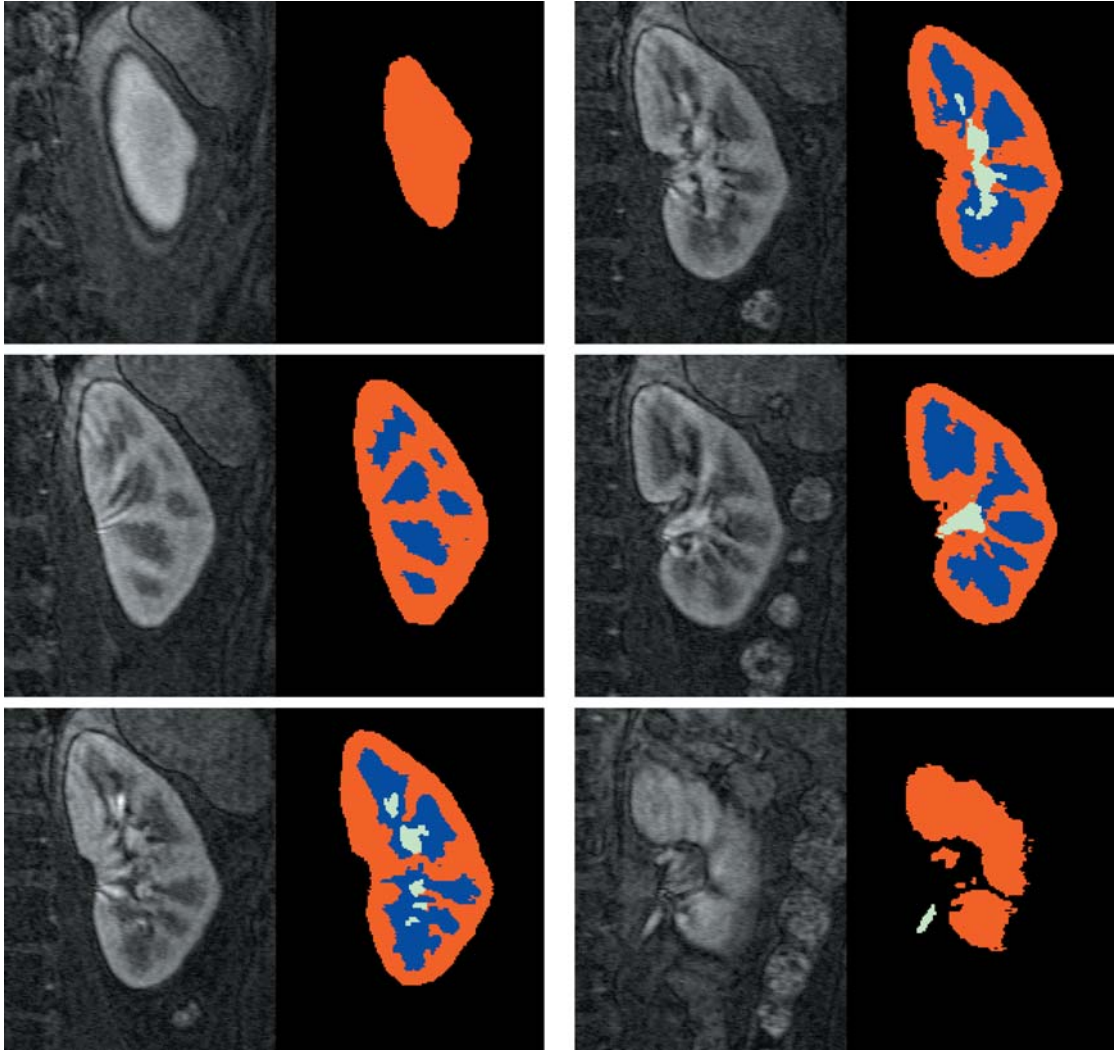
Figure 13.   Kidney in a 3D MRI angio data [55 × 80 × 32] segmented into cortex, medulla, and collecting system.

algorithms from combinatorial optimization and there are no numerical convergence issues.

The downside of our global optimization approach is a restricted set of constraints that we have available to guide the segmentation process (Kolmogorov and Zabih, 2004; Kolmogorov and Boykov, 2005). At the same time, the range of available constraints is sufficiently wide to make graph cuts very useful in practice. Surprisingly, graph cuts can implement the same geometrically-motivated cues (Kolmogorov and Boykov, 2005) that are widely used in continuous level-sets methods.

Our main technical tools are weights and topology of n-links and t-links of image-embedded graphs.

Typically, n-links encode the segmentation boundary cost that is closely related to its geometric length in a non-Euclidean image-based metric (Boykov and Kolmogorov, 2003). At the moment we can not optimize second order properties of the boundary (e.g. curvature) which may be helpful in some applications. It is plausible that combinatorial representation of such properties would require an energy function (4) with triple-cliques interactions in the boundary term[15] while our n-links describe only pairwise relations. Recent results in Kolmogorov and Zabih (2004) describe a class of triple-clique interactions that can be optimized with graph cuts. It is not clear, though, if this class is good enough to model any particular second-order boundary

properties of segments. In case this open question has a positive answer our object extraction framework should be able to benefit from the corresponding construction.

Our examples with directed n-links (see Section 2.7) show that our method can model certain boundary properties which might be hard to handle with other object extraction methods. In fact, there are many other creative ways to use specific features of our graph-based approach. For example, directed n-links can be used to optimize flux (Kolmogorov and Boykov, 2005) of a given vector field through the segmentation boundary. A field of such vectors for each image pixel can be *a priori* defined from data (e.g. intensity gradients) or other contextual information (e.g. gradients of a distance map for a prior object/shape model) (Kolmogorov and Boykov, 2005).

Note that (Kolmogorov and Boykov, 2005) characterized the boundary component of the energy that can be minimized with graph cuts through geometric terms like *length/area* and *flux*. Note that these two geometric concepts have somewhat dual effect on the segmentation boundary. While optimization of length causes "shrinking" of the boundary, optimization of flux may have the opposite "ballooning" effect on the boundary. The discovery of flux component in Kolmogorov and Boykov (2005) allows to counteract the "shrinking" bias that graph cut techniques were previously criticized for. The "shrinking" bias can be also addressed by integrating an appropriate regional bias.

The t-links are a tool for implementing region-based topological constraints and regional bias in the first term of energy formulation (2). Constraining the search space of cuts is a critical feature of our approach and the regional bias is an option that should be used carefully designed depending on a specific domain. For example, intensity-based regional models in (5,6) may not help to segment an object if its intensity properties are very similar to the background as in Fig. 12(a) (also known as "camouflage" problem). In this case, intensity distributions (5,6) are almost identical and relative weight for that type of regional bias in the energy should be set close to zero.

Our artificial example in Fig. 4 shows that a regional bias may work well even when there is a significant overlap in the intensity properties of the object and background. In real images, however, it may be hard to set the right value of λ and the balance between the region and boundary terms of energy (2) might be elusive.

Robustness of the regional term can be improved if simple gray-scale intensity histograms in (5,6) are re-placed with more elaborate regional bias models based on color, local texture, etc. For example, in Fig. 13 we used

$$R_p(A_p) = -\ln \Pr(\bar{I}_p | A_p)$$

based on multi-dimensional vector $\bar{I}_p$ of intensities obtained at each pixel in a sequence of timely acquisitions as a contrast agent propagated through different tissues. Distinct functional properties of these tissues gave informative regional models based on multi-dimensional intensity distributions. Importance of the careful estimation of regional intensity properties of segments is also emphasized in (Blake et al., 2004; Rother et al., 2004) where object/background color distribution is estimated using Gaussian Mixture Models with fairly impressive results.

One additional option studied in (Blake et al., 2004; Rother et al., 2004) is that the parameters of the energy (e.g. regional bias) can be iteratively re-estimated. It is also possible to use new correcting seeds during the editing stage (see Section 2.6) to learn the energy parameters. A new global optima with "updated" regional term can be efficiently recomputed from a previous solution using algorithmic ideas similar to those in Section 2.6. Efficient updating of n-links was studied in Kohli and Torr (2005) in the context of dynamic applications.

Finally, there is an interesting way to generalize our segmentation framework to multi-object extraction problem. Figures 12 (b) and 13(b) show examples of multi-label segmentations obtained in a sequence of consecutive independent binary steps based on constrained *s-t* graph cuts described in Section 2. In fact, the multi-way graph-cut algorithms in Boykov et al. (2001) can be used to extract multiple objects at once. In practice, extracting multiple objects simultaneously should be more robust, convenient, and faster when compared to iterative binary approach.

Our graph-cuts approach can be extended to simultaneous multi-object extraction as follows. A graph should have multiple terminals representing each object of interest.[16] A user can appropriately place seeds (hard constraints) of different "colors" to provide high-level contextual information and to constrain the search space of feasible multi-way cuts. In fact, seeds of each color would represent infinity-cost t-links to the corresponding graph terminal (object label). N-links between image pixels can still represent a segmentation boundary cost. The multi-way cut approach in

Boykov et al. (2001) allows to set different discontinuity penalties $B_{p,q}(A_p, A_q)$ between a given pair of neighboring pixels depending on a specific pair of labels assigned to them. A minimum cost multi-way cut is a graph partitioning that separates all terminals into isolated segments by severing edges of the smallest possible accumulative cost. Infinity cost t-links guarantee that multi-way partitioning of a constrained graph described above infers an optimal multi-object image segmentation. The corresponding segmentation energy is a direct generalization of (2) to a multi-label setting

$$E(A) = \lambda \cdot \sum_{p \in \mathcal{P}} R_p(A_p)$$
$$+ \sum_{\{p,q\} \in \mathcal{N}} B_{p,q}(A_p, A_q) \cdot \delta_{A_p \neq A_q} \quad (12)$$

where $A \in \{obj_1, obj_2, \ldots, obj_N\}^{|\mathcal{P}|}$ is a multi-object image labeling.

Unfortunately, multi-way cut problem is NP-hard. Thus, we loose guaranteed global optimality that we have in the binary object/background setting of Section 2. At the same time, α-*expansion* algorithm in Boykov et al. (2001) can efficiently find provably good approximating solutions to the multi-way cut problem above. In practice, it has been shown that these approximating solutions have sufficiently high quality for many problems in vision (e.g. for stereo (Scharstein and Szeliski, 2002; Szeliski and Zabih, 1999)). Constrained multi-way graph cuts is a promising natural extension of our object/background segmentation approach to a more general multi-object extraction problem.

Acknowledgements

Notes

1. Preliminary version of this work appeared in Boykov and Jolly (2001).
2. One survey can be found in Boykov and Veksler (2006).
3. *Impossibility theorem* in Kleinberg (2002) shows that no clustering algorithm can simultaneously satisfy three basic axioms on *scale-invariance*, *richness*, and *consistency*. Thus, any clustering method has some bias.
4. One interesting exception is a recent *random walker* approach to image segmentation (Grady, 2005) based on optimization problem over a finite set of real-valued variables.
5. It seems feasible that *flow-recycling* (Kohli and Torr, 2005) and *cut recycling* (Juan and Boykov, 2006) can be combined for a multiplied effect.
6. A patent application jointly with G. Paladini (Siemens Corp. Research, Princeton, NJ) is pending.
7. The relationship of binary graph cuts with piece-wise constant Mumford-Shah model is obvious from the characterization of the segmentation boundary cost in terms of geometric length (Boykov and Kolmogorov, 2003). The general form of Mumford-Shah functional can be seen as a continuous counterpart for the piece-wise smooth discrete MRF models which, for example, can be optimized with non-binary graph cuts methods like α-expansion (Boykov et al., 2001).
8. Some versions of *region growing* method (Griffin et al., 1994; Reese, 1999) use region-based hard constraints similar to ours.
9. A typed or hand-written letter is an example of a binary image. Restoration of such an image may involve removal of a salt and pepper noise.
10. In contrast, each pair of connected nodes on a directed graph is linked by two distinct (directed) edges (p, q) and (q, p). Such directed edges can be very useful in some applications (see Section 2.7).
11. In fact, non-negativity assumption is not essential. For example, at any pixel p we can always increase the values of $R_p(\text{"obj"})$ and $R_p(\text{"bkg"})$ by the same amount without changing the minimum of energy (2). The actual limitations for discontinuity penalties B are discussed in Section 2.7.
12. Note that adding correcting "object" seeds to pixels that are already segmented as "object" can not change the optimal segmentation. Indeed, such hard constraints are already satisfied by the current optimal solution. The same argument applies to correcting "background" seeds.
13. Note that the minimum cut severs exactly one of two t-links at pixel p.
14. Courtesy of the Library of Congress collection of slides by S.M. Prokudin-Gorskii made in Russia in the early 20th century. The whole collection is available at www.loc.gov/exhibits/empire/

15. This point was raised by Davi Geiger (NY University) in a private communication.
16. In Section 2 we use only two graph terminals *s* and *t* representing object and background labels.

References

Amini, A.A., Weymouth, T.E., and Jain, R.C. 1990. Using dynamic programming for solving variational problems in vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(9):855–867.

Appleton, B. and Talbot, H. 2006. Globally minimal surfaces by continuous maximal flows. *IEEE transactions on Pattern Analysis and Pattern Recognition (PAMI)*, 28(1):106–118.

Blake, A., Rother, C., Brown, M., Perez, P., and Torr, P. 2004. Interactive image segmentation using an adaptive gmmrf model. In *European Conference on Computer Vision (ECCV)*, Prague, Chech Republic.

Boykov, Y. and Kolmogorov, V. 2003. Computing geodesics and minimal surfaces via graph cuts. In *International Conference on Computer Vision*, vol. I, pp. 26–33.

Boykov, Y., Veksler, O., and Zabih, R. 1998. Markov random fields with efficient approximations. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 648–655.

Boykov, Y. and Jolly, M.-P. 2001. *Interactive graph cuts* for optimal boundary & region segmentation of objects in N-D images. In *International Conference on Computer Vision*, vol. I, pp. 105–112, July 2001.

Boykov, Y. and Kolmogorov, V. 2004. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(9):1124–1137.

Boykov, Y., Kolmogorov, V., Cremers, D., and Delong, A. 2006. An integral solution to surface evolution PDEs via geo-cuts. In *European Conference on Computer Vision*, LNCS 3953, Graz, Austria, vol. III, pp. 409–422.

Boykov, Y. and Veksler, O. 2006. Graph cuts in vision and graphics: Theories and applications. In: N. Paragios, Y. Chen, and O. Faugeras, (Eds.), Handbook of Mathematical Models in Computer Vision, Springer-Verlag, pp. 79–96.

Boykov, Y., Veksler, O., and Zabih, R. 2001. Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(11):1222–1239.

Bray, M., Kohli, P., and Torr, P.H.S. 2006. Posecut: Simultaneous segmentation and 3D pose estimation of humans using dynamic graph-cuts. In *European Conference on Computer Vision*, Graz, Austria, May 2006, (to appear).

Caselles, V., Kimmel, R., and Sapiro, G. 1997. Geodesic active contours. *International Journal of Computer Vision*, 22(1):61–79.

Cohen, L.D. 1991. On active contour models and ballons. *Computer Vision, Graphics, and Image Processing: Image Understanding*, 53(2):211–218.

Cohen, L.D. and Kimmel, R. 1997. Global minimum for active contour models: A minimal path approach. *International Journal of Computer Vision*, 24(1):57–78.

Cook, W.J., Cunningham, W.H., Pulleyblank, W.R., and Schrijver, A. 1998. *Combinatorial Optimization*. John Wiley & Sons.

Cox, I.J., Rao, S.B., and Zhong, Y. 1996. "Ratio regions": a technique for image segmentation. In *International Conference on Pattern Recognition*, vol. II, pp. 557–564.

Cremers, D. 2006. Dynamical statistical shape priors for level set based tracking. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, (to appear).

Cremers, D., Osher, S.J., and Soatto, S. 2006. Kernel density estimation and intrinsic alignment for shape priors in level set segmentation. *International Journal of Computer Vision*, (to appear).

Falcão, A.X., Udupa, J.K., Samarasekera, S., and Sharma, S. 1998. User-steered image segmentation paradigms: Live wire and live lane. *Graphical Models and Image Processing*, 60:233–260.

Felzenszwalb, P. and Huttenlocher, D. 2004. Efficient graph-based image segmentation. *International Journal of Computer Vision*, 59(2):167–181.

Ford, L. and Fulkerson, D. 1962. *Flows in Networks*. Princeton University Press.

Funka-Lea, G., Boykov, Y., Florin, C., Jolly, M.-P., Moreau-Gobard, R., Ramaraj, R., and Rinck, D. 2006. Automatic heart isolation for CT coronary visualization using graph-cuts. In *IEEE International Symposium on Biomedical Imaging*, Arlington, VA, April 2006.

Geiger, D., Gupta, A., Costa, L.A., and Vlontzos, J. 1995. Dynamic programming for detecting, tracking, and matching deformable contours. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(3):294–402.

Goldberg, A.V. and Tarjan, R.E. 1988. A new approach to the maximum-flow problem. *Journal of the Association for Computing Machinery*, 35(4):921–940.

Grady, L. 2005. Multilabel random walker segmentation using prior models. In *IEEE Conference of Computer Vision and Pattern Recognition*, San Diego, CA, June 2005, vol. 1, pp. 763–770.

Greig, D., Porteous, B., and Seheult, A. 1989. Exact maximum a posteriori estimation for binary images. *Journal of the Royal Statistical Society, Series B*, 51(2):271–279.

Griffin, L.D., Colchester, A.C.F., Röll, S.A., and Studholme, C.S. 1994. Hierarchical segmentation satisfying constraints. In *British Machine Vision Conference*, pp. 135–144.

Haralick, R.M. and Shapiro, L.G. 1992. *Computer and Robot Vision*. Addison-Wesley Publishing Company.

Hochbaum, D.S. 1998. The pseudoflow algorithm for the maximum flow problem. *Manuscript, UC Berkeley*, revised 2003, Extended abstract in: The pseudoflow algorithm and the pseudoflow-based simplex for the maximum flow problem. Proceedings of IPCO98, June 1998. *Lecture Notes in Computer Science*, Bixby, Boyd and Rios-Mercado (Eds.) 1412, Springer, pp. 325–337.

Isard, M. and Blake, A. 1998. *Active contours*. Springer-Verlag.

Ishikawa, H. and Geiger, D. 1998. Occlusions, discontinuities, and epipolar lines in stereo. In *5th European Conference on Computer Vision*, pp. 232–248.

Ishikawa, H. and Geiger, D. 1998. Segmentation by grouping junctions. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 125–131.

Ishikawa, H. 2003. Exact optimization for Markov Random Fields with convex priors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(10):1333–1336.

Jermyn, I.H. and Ishikawa, H. 1999. Globally optimal regions and boundaries. In *International Conference on Computer Vision*, vol. II, pp. 904–910.

Juan, O. and Boykov, Y. 2006. Active Graph Cuts. In *IEEE Conference of Computer Vision and Pattern Recognition*, 2006 (to appear).

Kass, M., Witkin, A., and Terzolpoulos, D. 1988. Snakes: Active contour models. *International Journal of Computer Vision*, 1(4):321–331.

Kimmel, R. and Bruckstein, A.M. 2003. Regularized Laplacian zero crossings as optimal edge integrators. *International Journal of Computer Vision*, 53(3):225–243.

Kirsanov, D. and Gortler, S.J. 2004. A discrete global minimization algorithm for continuous variational problems. *Harvard Computer Science Technical Report*, TR-14-04, July 2004, (also submitted to a journal).

Kleinberg, J. 2002. An impossibility theorem for clustering. In *The 16th conference on Neural Information Processing Systems (NIPS)*.

Kohli, P. and Torr, P.H.S. 2005. Efficiently solving dynamic markov random fields using graph cuts. In *International Conference on Computer Vision*.

Kohli, P. and Torr, P.H.S. 2006. Measuring uncertainty in graph cut solutions—efficiently computing min-marginal energies using dynamic graph cuts. In *European Conference on Computer Vision*, Graz, Austria, May 2006 (to appear).

Kolmogorov, V. and Boykov, Y. 2005. What metrics can be approximated by geo-cuts, or global optimization of length/area and flux. In *International Conference on Computer Vision*, Beijing, China, vol. I, pp. 564–571.

Kolmogorov, V., Criminisi, A., Blake, A., Cross, G., and Rother, C. 2005. Bi-layer segmentation of binocular stereo video. In *IEEE Conference of Computer Vision and Pattern Recognition*, San Diego, CA.

Kolmogorov, V. and Zabih, R. 2002. Multi-camera scene reconstruction via graph cuts. In *7th European Conference on Computer Vision*, volume III of *LNCS 2352*, pp. 82–96, Copenhagen, Denmark, May 2002. Springer-Verlag.

Kolmogorov, V. and Zabih, R. 2004. What energy functions can be minimized via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(2):147–159.

Kumar, M.P., Torr, P.H.S., and Objcut, A.Z. 2005. In *IEEE Conference of Computer Vision and Pattern Recognition*, pp. 18–25.

Kwatra, V., Schodl, A., Essa, I., and Bobick, A. 2003. GraphCut textures: image and video synthesis using graph cuts. In *ACM Transactions on Graphics (SIGGRAPH)*, vol. 22, July 2003.

Lempitsky, V., Boykov, Y., and Ivanov, D. 2006. Oriented visibility for multiview reconstruction. In *European Conference on Computer Vision*, Graz, Austria, May 2006 (to appear).

Li, K., Wu, X., Chen, D.Z., and Sonka, M. 2006. Optimal surface segmentation in volumetric images-a graph-theoretic approach. *IEEE transactions on Pattern Analysis and Pattern Recognition (PAMI)*, 28(1):119–134.

Li, Y., Sun, J., and Shum, H.-Y. 2005. Video object cut and paste. In *SIGGRAPH (ACM Transaction on Graphics)*,

Li, Y., Sun, J., Tang, C.-K., and Shum, H.-Y. 2004. Lazy snapping. In *SIGGRAPH (ACM Transaction on Graphics)*.

Lombaert, H., Sun, Y., Grady, L., and Xu, C. 2005. A multilevel banded graph cuts method for fast image segmentation. In *International Conference on Computer Vision*, October 2005.

Mortensen, E.N. and Barrett, W.A. 1998. Interactive segmentation with intelligent scissors. *Graphical Models and Image Processing*, 60:349–384.

Mumford, D. and Shah, J. 1989. Optimal approximations by piecewise smooth functions and associated variational problems.

Comm. Pure Appl. Math., 42:577–685.

Murota, K. 2003. *Discrete Convex Analysis*. SIAM Monographs on Discrete Mathematics and Applications.

Osher, S. and Paragios, N. 2003. *Geometric Level Set Methods in Imaging, Vision, and Graphics*. Springer Verlag.

Osher, S.J. and Fedkiw, R.P. 2002. *Level Set Methods and Dynamic Implicit Surfaces*. Springer Verlag.

Reese, L.J. 1999. Intelligent paint: Region-based interactive image segmentation. Master's thesis, Brigham Young University.

Rother, C., Kolmogorov, V., and Blake, A. 2004. Grabcut—interactive foreground extraction using iterated graph cuts. In *ACM Transactions on Graphics (SIGGRAPH)*.

Rother, C., Kumar, S., Kolmogorov, V., and Blake, A. 2005. Digital tapestry. In *IEEE Conference of Computer Vision and Pattern Recognition*, San Diego, CA.

Roy, S. and Cox, I. 1998. A maximum-flow formulation of the n-camera stereo correspondence problem. In *IEEE Proc. of Int. Conference on Computer Vision*, pp. 492–499.

Sapiro, G. 2001. *Geometric Partial Differential Equations and Image Analysis*. Cambridge University Press.

Scharstein, D. and Szeliski, R. 2002. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision*, 47(1/3):7–42.

Sethian, J.A. 1999. *Level Set Methods and Fast Marching Methods*. Cambridge University Press.

Shi, J. and Malik, J. 2000. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905.

Szeliski, R. and Zabih, R. 1999. An experimental comparison of stereo algorithms. In *Vision Algorithms: Theory and Practice*, number 1883 in LNCS, pp. 1–19, Springer-Verlag, Corfu, Greece, September 1999.

Vasilevskiy, A. and Siddiqi, K. 2002. Flux maximizing geometric flows. *PAMI*, 24(12):1565–1578.

Veksler, O. 2000. Image segmentation by nested cuts. In *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 339–344.

Vogiatzis, G., Torr, P.H.S., and Cipolla, R. 2005. Multi-view stereo via volumetric graph-cuts. In *IEEE Conference of Computer Vision and Pattern Recognition*, pp. 391–398.

Wang, J., Bhat, P., Colburn, R.A., Agrawala, M., and Cohen, M.F. 2005. Interactive video cutout. In *SIGGRAPH (ACM Transaction on Graphics)*.

Wu, Z. and Leahy, R. 1993. An optimal graph theoretic approach to data clustering: Theory and its application to image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(11):1101–1113.

Xu, N., Bansal, R., and Ahuja, N. 2003. Object segmentation using graph cuts based active contours. In *IEEE Conference on Computer Vision and Pattern Recognition*, vol. II, pp. 46–53.

Yezzi, A., Jr., Kichenassamy, S., Kumar, A., Olver, P., and Tannenbaum, A. 1997. A geometric snake model for segmentation of medical imagery. *IEEE Transactions on Medical Imaging*, 16(2):199–209.

Zhu, S.C. and Yuille, A. 1996. Region competition: Unifying snakes, region growing, and Bayes/MDL for multiband image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(9):884–900.