Applications of parametric maxflow in computer vision

Vladimir Kolmogorov
University College London
vnk@adastral.ucl.ac.uk

Yuri Boykov
University of Western Ontario
yuri@csd.uwo.ca

Carsten Rother
Microsoft Research
carrot@microsoft.com

Abstract

The maximum flow algorithm for minimizing energy functions of binary variables has become a standard tool in computer vision. In many cases, unary costs of the energy depend linearly on parameter λ. In this paper we study vision applications for which it is important to solve the maxflow problem for different λ's. An example is a weighting between data and regularization terms in image segmentation or stereo: it is desirable to vary it both during training (to learn λ from ground truth data) and testing (to select best λ using high-knowledge constraints, e.g. user input).

We review algorithmic aspects of this parametric maximum flow *problem previously unknown in vision, such as the ability to compute all breakpoints of λ and corresponding optimal configurations in finite time.*

These results allow, in particular, to minimize the ratio of some geometric functionals, such as flux of a vector field over length (or area). Previously, such functionals were tackled with shortest path techniques applicable only in 2D.

We give theoretical improvements for "PDE cuts" [5]. We present experimental results for image segmentation, 3D reconstruction, and the cosegmentation problem.

1. Introduction

This paper focuses on the following problem:

[Parametric maxflow] *Minimize energy functions of binary variables $E^\lambda(\mathbf{x})$ for values of parameter λ in the set $I \subseteq \mathbb{R}$ where*

$$E^\lambda(\mathbf{x}) = \sum_{u\in\mathcal{V}} (a_u + b_u\lambda)x_u + \sum_{(u,v)\in\mathcal{E}} V_{uv}(x_u, x_v) \quad (*)$$

Here $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is an undirected graph, $x_u \in \{0, 1\}$ is the label of node u. Terms V_{uv} are assumed to be submodular, i.e. $V_{uv}(0,0) + V_{uv}(1,1) \leq V_{uv}(0,1) + V_{uv}(1,0)$. Set I may be given in advance as some interval $[\lambda^{\min}, \lambda^{\max}]$, or it can be provided online (i.e. the next value of λ is selected after the problem is solved for previous λ's).

It is well known in computer vision that for a fixed value λ energy (*) can be minimized via a min cut/max flow algorithm. However, *parametric* maxflow algorithms remained unnoticed in vision, despite the fact that the parametric problems occur very frequently (see section 1.1). In this paper we review algorithmic aspects of solving problem (*). in particular the following facts: (i) there is a finite number of breakpoints of parameter λ and corresponding optimal solutions; (ii) all these solutions can be found in finite time using the ES method [14, 17], which makes at most 2 calls to the maxflow algorithm per breakpoint (on average); (iii) the parametric algorithm can be implemented much more efficiently in the *monotonic* case when coefficients b_u are either all non-negative or all non-positive [14, 32, 15] [1].

To the best of our knowledge, these results have not been used in vision. Thus, one of our contributions is to show the relevance of efficient algorithms for solving the parametric maxflow problem (*) for vision applications. Note, however, that a similar algorithm has been used for image restoration using total variation minimization [18, 37, 11, 8].

1.1. Parametric maxflow problem (*) in vision

Many vision problems, e.g. binary image segmentation and stereo, can be formulated in the MAP-MRF framework: the goal is to minimize an energy function which is a weighted sum of the data term and the regularization term. This naturally leads to the parametric maxflow problem, where λ represents the weighting factor between the two terms. Choosing the correct value of λ is a very challenging task. One standard approach taken e.g. in [2] for image segmentation is cross-validation: learn λ discriminatively on a dataset with ground truth segmentations. This requires solving the parametric maxflow problem, so the learning task will benefit from the algorithms reviewed in this paper[2].

While such learning may provide a reasonable value of λ, it is generally acknowledged that a fixed value of λ cannot work well for all images. (An example is shown in section 4.) This motivates solving the parametric problem during testing. An additional high-knowledge criterion would then be needed to choose the desired λ and corresponding segmentation. In some domains automatic criteria can be designed by studying the problem [12]. In the lack of automatic criteria, one can always resort to user selection.

Similar issues arise when *flux* of some vector field is combined with regularization for image segmentation [19, 34, 21, 23] or 3D reconstruction [6]. (By the divergence the-

[1] Note that the algorithm in [15] for the monotonic case is called "parametric maximum flow", which clashes with the name of problem (*). We use the term "parametric maxflow" for the general problem (*).

[2] Note that this is a very simple form of learning. A large number of more advanced statistical techniques were developed in the literature, e.g. pseudo-likelihood and contrastive divergence. More general learning is outside the scope of this paper; we refer to [25] for more details.

orem, flux can be written as a sum of unary terms, therefore it is a special case of the parametric maxflow problem.)

Ratio optimization Several authors [10, 19, 36, 16] proposed to optimize the ratio of certain functionals for image segmentation. The "ratio region" method [10] minimizes Riemannian length of the boundary over weighted area, while [19] maximizes flux of some vector field over the boundary length. Note that [10, 19] use shortest path based techniques which are applicable to 2D segmentation. We observe that the parametric maxflow allows to extend these techniques to 3D. Indeed, it is known in the fractional programming literature that minimization of the ratio of some discrete functionals can be reduced to the parametric maxflow problem (see section 3 for details). Furthermore, discrete functionals can approximate certain continuous functionals such as Riemannian length/area [3, 23].

Our experimental results confirm the findings of [19] about the usefulness of optimizing flux over length, but also show some shortcomings of this criterion. We discuss ways to overcome these shortcomings.

Incorporating a global constraint into segmentation Parametric maxflow is also useful for optimizing functions which combine standard MRF terms and a global constraint which cannot be written down as a sum of unary and pairwise terms. In [29] a monotonic parametric maxflow was used for obtaining "balanced" binary segmentations. (The criterion in [29] was "ratioCut" which is the ratio of the cost of the cut over the product of areas of the two regions). This approach can potentially be used for optimizing the "normalized cuts" criterion [31]. Non-monotonic parametric maxflow was used in [30] for the cosegmentation problem whose goal is to compute spatially coherent segmentations in two images so that their global histograms match. We show experimentally that the ES method improves on the procedure used in [30].

On a high level, parametric maxflow is useful since it allows to compute a sequence of "reasonable" solutions by solving a parametric problem in which MRF terms are combined with a linear approximation of the global term. The best solution is then selected by evaluating the global energy function for solutions in the sequence.

PDE cuts It was recently shown [5] that graph cuts can perform a gradient descent in the space of contours for certain geometric functionals and metrics. In other words, they can compute the motion of contour/surface under gradient flow, thus providing an alternative numerical scheme to level sets. The method in [5] uses non-monotonic parametric maxflow for computing each step. Parameter λ controls the time step. The largest breakpoint value of λ corresponds to the smallest detectable move. (Experiments in [5] were performed using such smallest move).

Thus, the method in [5] would benefit from the parametric maxflow algorithm. We show that the smallest move can be computed via a *monotonic* parametric algorithm (see a proof in [24]), which in general is significantly faster.

2. Solving parametric maxflow

Consider function $F(\lambda) = \min_{\mathbf{x}} E^{\lambda}(\mathbf{x})$. It is the minimum of $2^{|\mathcal{V}|}$ linear functions, therefore F is a piecewise-linear concave function of λ with a finite number of breakpoints (Fig. 1). Configurations which are optimal for at least one λ are called *dominant* solutions.

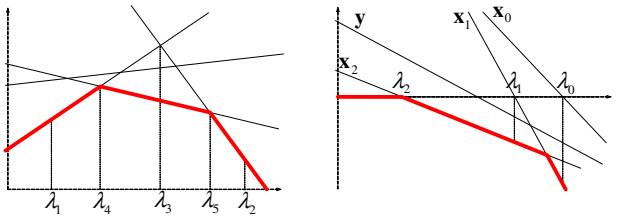
Computing all breakpoints To "solve" problem (*) for interval $I = [\lambda^{\min}, \lambda^{\max}]$ we could compute solutions $\mathbf{x}_1, \ldots, \mathbf{x}_k$ and intervals $I_1 = [\lambda^{\min}, \lambda_1], I_2 = [\lambda_1, \lambda_2], \ldots, I_k = [\lambda_{k-1}, \lambda^{\max}]$ such that $\mathbf{x}_i$ is optimal for $\lambda \in I_i$. This can be done via the technique known as the ES method [14, 17]. Fig. 1(a) illustrates its geometric interpretation. The method maintains a list $(\mathbf{x}_1, I_1), \ldots, (\mathbf{x}_k, I_k)$ such that solution $\mathbf{x}_i$ is optimal for interval $\lambda \in I_i$ and $\sup I_i \leq \inf I_j$ for adjacent intervals I_i, I_j $(j = i + 1)$:

init: compute minimizers $\mathbf{x}^{\min}$, $\mathbf{x}^{\max}$ for $\lambda^{\min}$ and $\lambda^{\max}$
  if $\mathbf{x}^{\min} = \mathbf{x}^{\max}$ set list as $(\mathbf{x}^{\min}, [\lambda^{\min}, \lambda^{\max}])$
  else set list as $(\mathbf{x}^{\min}, \{\lambda^{\min}\}), (\mathbf{x}^{\max}, \{\lambda^{\max}\})$
while there are adjacent items $(\mathbf{x}_i, I_i)$,$(\mathbf{x}_j, I_j)$ with a gap,
    i.e. $(\lambda_i, \lambda_j) = (\sup I_i, \inf I_j)$ is non-empty:
  solve linear equation $E^{\lambda}(\mathbf{x}_i) = E^{\lambda}(\mathbf{x}_j)$
  if λ_i is a solution set $I_j := I_j \cup [\lambda_i, \lambda_j]$
  else if λ_j is a solution set $I_i := I_i \cup [\lambda_i, \lambda_j]$
  else there must exist unique solution $\lambda \in (\lambda_i, \lambda_j)$:
    compute minimizer $\mathbf{x}$ of $E^{\lambda}(\cdot)$
    if $\mathbf{x} = \mathbf{x}_i$ **or** $\mathbf{x} = \mathbf{x}_j$ set
      $I_i := I_i \cup [\lambda_i, \lambda], I_j := I_j \cup [\lambda, \lambda_j]$
    else insert $(\mathbf{x}, \{\lambda\})$ between $(\mathbf{x}_i, I_i)$ and $(\mathbf{x}_j, I_j)$

The ES method makes at most $2B + 2$ calls to the maxflow procedure where B is the number of breakpoints of function F in the given interval [17]. Unfortunately, in the worst case B may be exponential in the size of the problem [7]. (The counter-example originates from the pathological graph of Zadeh which was used to show that the complexity of the simplex method may be exponential). This does not necessarily mean, however, that such pathological cases will occur in practice. In vision applications that we tested the number of breakpoints was manageable.

We implemented the ES method using the maxflow algorithm in [4]. When computing maximum flow for the next value of λ, we reuse flow from the previous computation, as well as search trees as in [22]. (Changing λ results in updating capacities of edges from the source and to the sink.)

Computing one breakpoint In certain cases computing all breakpoints may not be necessary. A binary search style procedure may be more appropriate: after computing solution $\mathbf{x}$ for value $\lambda \in (\lambda_1, \lambda_2)$ we decide whether to proceed with interval $[\lambda_1, \lambda]$ or $[\lambda, \lambda_2]$. Then the ES method can be run only for the appropriate interval. For exam-



(a) (b)

Figure 1. **Parametric problem.** Each configuration $\mathbf{x}$ corresponds to a linear function $E^\lambda(\mathbf{x})$. Their lower envelope gives the function $F(\lambda) = \min_{\mathbf{x}} E^\lambda(\mathbf{x})$ (shown in bold red). **(a)** Computing all breakpoints in the interval $[\lambda_1, \lambda_2]$ for a general parametric problem. The ES method will run maxflow for values $\lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5$. **(b)** Ratio optimization problem; $E^\lambda(\mathbf{x}) = P(\mathbf{x}) - \lambda Q(\mathbf{x})$ (case when $\min_{\mathbf{x}} R(\mathbf{x}) > -\infty$ and there exists $\mathbf{x}$ with $P(\mathbf{x}) = Q(\mathbf{x}) = 0$). Line with a negative slope corresponding to configuration $\mathbf{x}$ intersects the OX axis at point $\lambda = R(\mathbf{x})$. Newton's method starting with configuration $\mathbf{x}_0$ will produce the sequence $(\mathbf{x}_0, \lambda_0), (\mathbf{x}_1, \lambda_1), (\mathbf{x}_2, \lambda_2)$.

ple, to compute the first breakpoint in $[\lambda^{\min}, \lambda^{\max}]$ we can process the smallest index i for which there is a gap (i.e. $\sup I_i < \inf I_{i+1}$) until the first breakpoint is identified.

In our experiments such a procedure always took just a few iterations to converge. In general, however, we are not aware of any polynomial bounds on the number of steps. It should be noted that polynomial-time algorithms for computing a particular breakpoint do exist (see e.g. [28, 17]). It's not clear, though, whether they would be faster than the simple method described above, given that experimentally this simple method makes just a few calls to maxflow.

Monotonic case Finally, let us discuss the *monotonic* case when coefficients b_u in (*) are either all non-negative or all non-positive. For concreteness, let us assume that $b_u \geq 0$ for all nodes u (the other case is similar). In that case the ES algorithm can be implemented much more efficiently [14, 15]. The key property is the *nestedness* of optimal solutions [14, 32, 15]:

Proposition 2.1. *(a) Suppose that $\mathbf{x}_1$ is optimal for λ_1 and $\lambda_1 < \lambda$. Then there exists an optimal solution $\mathbf{x}$ for λ such that $\mathbf{x}_1 \geq \mathbf{x}$.*

(b) Suppose that $\mathbf{x}_1$, $\mathbf{x}_2$ with $\mathbf{x}_1 \geq \mathbf{x}_2$ are optimal respectively for λ_1, λ_2 and $\lambda_1 < \lambda < \lambda_2$. Then there exists an optimal solution $\mathbf{x}$ for λ such that $\mathbf{x}_1 \geq \mathbf{x} \geq \mathbf{x}_2$.

(c) Function $F(\cdot)$ has at most $|\mathcal{V}| + 1$ breakpoints.

(Here $\mathbf{x} \geq \mathbf{y}$ means that $x_u \geq y_u$ for all nodes u.) Thus, the ES method can be modified so that the list of solutions satisfies $\mathbf{x}_1 > \ldots > \mathbf{x}_k$. For example, when computing optimal solution $\mathbf{x}$ for $\lambda \in (\lambda_i, \lambda_j)$ we can enforce constraint $\mathbf{x}_i \geq \mathbf{x} \geq \mathbf{x}_j$ by fixing nodes in the set $\bar{\mathcal{V}}_i = \{u \in \mathcal{V} \mid (x_i)_u = 0\}$ to 0, and nodes in the set $\mathcal{V}_j = \{u \in \mathcal{V} \mid (x_j)_u = 1\}$ to 1. Thus, the maxflow algorithm can be run only for nodes in $\mathcal{V}_{ij} = \mathcal{V} - \bar{\mathcal{V}}_i - \mathcal{V}_j$, which may yield a significant speed-up.

We implemented this modified scheme as follows. We explicitly maintain regions $\mathcal{V}_{12}, \ldots, \mathcal{V}_{k-1,k}$ (note that together with $\bar{\mathcal{V}}_1, \mathcal{V}_k$ they form a partition of the set of nodes $\mathcal{V}$). Similar to the general case, we also maintain a graph with the residual flow from previous computations. Consider the maxflow algorithm for $\lambda \in (\lambda_i, \lambda_j) = (\sup I_i, \inf I_j)$ which involves nodes in $\mathcal{V}_{ij}$. If these nodes are all labeled as 0 or all labeled as 1, then $\mathbf{x} = \mathbf{x}_i$ or $\mathbf{x} = \mathbf{x}_j$, so the intervals I_i, I_j are updated but set $\mathcal{V}_{ij}$ is unchanged. Otherwise $\mathcal{V}_{ij}$ is split into into two regions $\mathcal{V}_{ir}$, $\mathcal{V}_{rj}$ containing nodes in $\mathcal{V}_{ij}$ with label 0 and 1, respectively. (Index r corresponds to the new interval $\{\lambda\}$ inserted between i and j.) After splitting the region, we remove all edges between them. This is justified since all arcs from $\mathcal{V}_{ir}$ to $\mathcal{V}_{rj}$ are saturated by the Ford-Fulkerson theorem (we assume that the source corresponds to label 0), and it is easy to see that they will remain saturated afterwards.

The worst-case complexity of this technique proposed in [14] is $O(n)$ maximum flow computations on a graph with $O(|\mathcal{V}|)$ nodes and $O(|\mathcal{E}|)$ edges. The worst case may occur if, e.g., each time $\mathcal{V}_{ij}$ is split into regions of size 1 and $|\mathcal{V}_{ij}| - 1$. In our experiments, however, the splits were rather balanced, and so the algorithm was quite efficient[3].

3. Ratio minimization

Parametric maxflow problem often arises in the context of *ratio optimization*. Consider two functions $P, Q : \mathcal{X} \to \mathbb{R}$ where $\mathcal{X} = 2^{\mathcal{V}}$ is the set of configurations and $Q(\mathbf{x}) \geq 0$ for all $\mathbf{x} \in \mathcal{X}$. The goal is to minimize function

$$R(\mathbf{x}) = \begin{cases} \frac{P(\mathbf{x})}{Q(\mathbf{x})} & \text{if } Q(\mathbf{x}) > 0 \\ -\infty & \text{if } Q(\mathbf{x}) = 0, P(\mathbf{x}) < 0 \\ +\infty & \text{if } Q(\mathbf{x}) = 0, P(\mathbf{x}) \geq 0 \end{cases}$$

Let $\lambda^* = \min_{\mathbf{x} \in \mathcal{X}} R(\mathbf{x})$ be the optimal value of the ratio (we assume that $\lambda^* < +\infty$), and consider function $E^\lambda(\mathbf{x}) = P(\mathbf{x}) - \lambda Q(\mathbf{x})$. It is easy to see that $\lambda^* \leq \lambda$ if and only if $\min_{\mathbf{x} \in \mathcal{X}} E^\lambda(\mathbf{x}) \leq 0$. Thus, value λ^* can be computed with an arbitrary precision using binary search (if $\lambda^* > -\infty$), if we are able to minimize function $E^\lambda(\cdot)$.

Alternatively, the problem can be solved via the Newton's method for fractional optimization, also known as Dinkelbach's method [13]. Starting with configuration $\mathbf{x}_0$ and value $\lambda_0 = R(\mathbf{x}_0)$, it produces a decreasing finite sequence $\lambda_0, \lambda_1, \ldots, \lambda^*$; upon termination, $\mathbf{x}$ is an optimal solution and $\lambda = R(\mathbf{x})$ is the minimum of the ratio:

[3] Note that the worst-case complexity can be improved to that of a single maxflow computation [15]. The method in [15] uses the push-relabel algorithm of Goldberg and Tarjan. For given $\lambda \in (\sup I_i, \inf I_j)$ and region $\mathcal{V}_{ij}$ two flow computations are performed in parallel, until the first one terminates. While this scheme improves the worst-case complexity, it is not necessarily faster in practice than the simpler technique [1].

Finally, we mention that there are iterative algorithms for solving the monotonic parametric maxflow problem converging in the limit [8, 38]. Unlike the methods described above, they are not guaranteed to terminate in finite time.

[0] Initialize: pick $\mathbf{x}$ with $R(\mathbf{x}) < +\infty$, set $\lambda = R(\mathbf{x})$.
[1] Compute minimizer $\mathbf{x}^*$ of function $E^\lambda(\cdot)$.
[2] If $Q(\mathbf{x}^*) = 0, P(\mathbf{x}^*) \geq 0$ stop.
[3] Set $\mathbf{x} := \mathbf{x}^*$, $\lambda := R(\mathbf{x}^*)$. If λ has decreased go to step 1, otherwise stop.

This algorithm is illustrated in Fig. 1(b). In step [0] configuration $\mathbf{x}$ can be set, for example, as a minimum of $E^\lambda(\cdot)$ for $\lambda > \lambda^*$. Note, in the case when $\min_{\mathbf{x}} R(\mathbf{x}) > -\infty$ and there exists $\mathbf{x}$ with $P(\mathbf{x}) = Q(\mathbf{x}) = 0$ the algorithm is equivalent to the method for computing the first breakpoint described in section 2.

In order to use Newton's method (or binary search), we should be able to minimize efficiently function $E^\lambda(\cdot)$. This can be done in the following special cases:

C1 P, Q are submodular, $P(\mathbf{x}) < 0$ for some $\mathbf{x}$.
C2 P is submodular, Q is supermodular, $P(\mathbf{x}) \geq 0$ for all $\mathbf{x}$.
C3 P is submodular, Q is modular.

Recall the the modular function is both submodular and supermodular; it can be written as $Q(\mathbf{x}) = b_0 + \sum_{u \in \mathcal{V}} b_u x_u$.

In this paper we focus on case **C3** and on the following special case of **C1**:

C1′ P is modular, Q are submodular, $P(\mathbf{x}) < 0$ for some $\mathbf{x}$.

As follows from the discussion above, both **C3** and **C1′** can be reduced to the parametric maxflow problem (*).

3.1. Optimizing Ratios of Geometric Functionals

We propose to combine the standard combinatorial optimization techniques for ratios of submodular or supermodular energies with the results in [3, 23] which show that the following continuous surface functionals in $\mathcal{R}^N$

$$\int_{\partial S} g(s)\, ds, \quad g(x) \geq 0 \qquad \textit{(length (2D) or area (3D))}$$
$$\int_{\partial S} \mathbf{v}_s \cdot \mathbf{n}_s\, ds \qquad \textit{(flux)}$$
$$\int_S f(p)\, dp \qquad \textit{(volumetric potential)}$$

can be arbitrarily closely approximated by submodular energies of binary variables on an N-dimensional grid. The results in [3, 23] apply to a wide class of metrics g (including general anisotropic Riemannian case), to any vector field $\{\mathbf{v}\}$, and to any volumetric potential function f. $\mathbf{n}_s$ denotes the outward normal to the surface at point s.

In particular, **C3** and **C1′** imply that we can minimize the following ratios of geometric surface functionals

$$\inf_{S \subset \Omega} \frac{\int_{\partial S} g(s) ds}{\int_S 1 \cdot dp} \qquad \textit{(area/volume - C3)} \qquad (1)$$

$$\inf_{S \subset \Omega} \frac{\int_S f(p) dp}{\int_{\partial S} g(s) ds} \qquad \textit{(potential/area - C1′)} \qquad (2)$$

$$\inf_{S \subset \Omega} \frac{\int_{\partial S} \mathbf{v}_s \cdot \mathbf{n}_s\, ds}{\int_{\partial S} g(s) ds} \qquad \textit{(flux/area - C1′)} \qquad (3)$$

Our combinatorial optimization approach allows to minimize these functionals over shapes S in a closed bounded (compact) subset $\Omega \subset \mathcal{R}^N$ since in practice we can use only finite grids to represent points in Ω. In applications Ω corresponds to a rectangular box representing an image or a bounded volume of interest in 3D.

Note that the minimal ratio of (1) is a (small) non-negative number $\lambda_0 \geq 0$ since both numerator and denominator are non-negative functionals. In contrast, the minimum ratio for (2) and (3) is a (large) negative $\lambda_0 \leq 0$. We will assume that volumetric potential function $f(p)$ has negative values at least at one point (pixel) $p \in \Omega$. The flux functional in (3) is guaranteed to be strictly negative for some shapes S unless the divergence of vector field $\{\mathbf{v}\}$ is null everywhere.

Ratio functionals (1), (2), and (3) have already been proposed for image segmentation by a number of authors [10, 19]. However, they used various forms of shortest path techniques restricting their methods to 2D applications. Our work extends ratio optimization to 3D problems in computer vision. In particular, besides segmentation of 3D medical and video data our method can be used for volumetric multiview reconstruction problems (see Sec. 4). Note that our approach can not minimize another ratio $\frac{\int_{\partial S} g1(s) ds}{\int_{\partial S} g2(s) ds}$ proposed for 2D image segmentation by [36].

Since our method allows to optimize different ratios of continuous surface functionals, it is helpful to compare geometric properties of the corresponding optimal shapes. Note that problem (1) is related to a well known in geometry *constrained isoperimetric problem* or *Cheeger problem* [9]. The most popular form of the isoperimetric problem is to find a shape in $\mathcal{R}^N$ of minimum surface area when the volume is fixed. Eventhough the solution was known to ancient Greeks (circle or sphere), a discrete version of this problem is NP-hard. One form of constrained isoperimetric problem is to find the shape with the lowest perimeter-to-volume ratio among all shapes inside some given closed bounded subset of $\mathcal{R}^N$. The optimal value of this ratio is called *Cheeger number*. [33, 20] show some interesting examples of the corresponding optimal *Cheeger sets*.

(1) states the constrained isoperimetric problem for non-Euclidean metric. The second row in Figure 2 shows (in red) Cheeger sets that we computed for several synthetic examples using standard metric g based on image gradients[4]. As pointed out in [10], solving (1) in the context of image segmentation has a bias to larger volumes. In order to segment a desired object one may need to crop a subset Ω close around its border. At the same time, interesting motivation to use Cheeger sets for image clustering is presented in [16] which also provides a heuristic-based approximation algorithm. Note that related Cheeger numbers and sets in graph

[4] In our experiments the boundary of the image was constrained to be background (see [24] for more details).

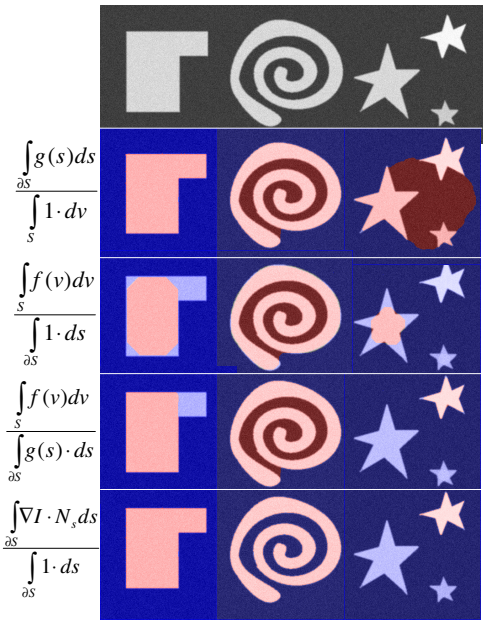


Figure 2. **Optimizing ratio functionals** for three simple images (top row). Other rows show segmentation results minimizing (from top to bottom): a) image weighted length of the boundary over volume, b) (regional) color model over Euclidean length, c) color model over image weighted length, d) flux of image gradients over Euclidean length of the boundary.

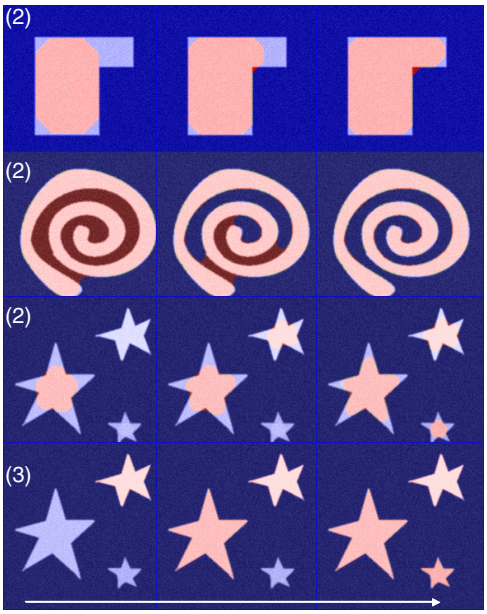


Figure 3. **Larger breakpoints.** The first column shows the best ratio solution to problems (2) and (3) (in both cases, for Euclidean length). The second and the third columns show representative dominant solutions obtained for $\lambda \geq \lambda_0$. Such solutions optimize the ratio among larger objects, see Proposition 3.1. Some real 3D examples are shown in Fig. 5.

theory are used to describe graph's "bottleneckness".

Problem (2) presents a certain generalization of the constrained isoperimetric (Cheeger) problem. If volumetric potential function $f(p) = -1$ for all $p \in \Omega$ then (2) is equivalent to (1). For example, if pixel's potentials $f(p)$ describe negative log-likelihoods of desired object intensities then the bias to larger volume is replaced with the bias to the right intensity model. For example, in row 3 and 4 of Fig. 2 we show the optimal ratio sets for the case of Euclidean and image-based metric, respectively. In these cases, the optimal sets try to maximize the number of white pixels (object color model) with the minimum possible length. The case of Euclidean length (row 3) reveals a strong bias to circular shapes which might be explained by a strong connection to the standard isoperimetric problem of ancient Greeks. Tendency to undersegment is also apparent. The case of image-weighted (Riemannian) metric (row 4) significantly reduces these problems but it still demonstrates a trend to undersegment details and to smooth out sharp corners. These problems are widely known as the "shrinking problem".

Arguably, optimizing the ratio of flux to length (3) presents no bias to any particular shape present in other ratios in this paper. This ratio functional was first proposed for image analysis by Jermyn and Ishikawa in [19] who also presented an efficient combinatorial optimization algorithm applicable to 2D problems. As they pointed out, this ratio functional is scale independent. If vector field $\mathbf{v}$ represents image gradients then the optimal ratio (3) gives the shape of the highest average contrast (assuming no contrast reversals on the boundary). Row 5 in Fig. 2 shows optimal segments for this ratio. The single star segmented in the image on the right has a slightly brighter color. Unlike the results above, the tips of the star are not cut. The divergence theorem implies that (3) is a special case of (2) for $f = div\ \mathbf{v}$.

3.2. Larger breakpoints and ratio optimization

As discussed earlier, the optimal ratio is obtained for solution $\mathbf{x}^*$ corresponding to the smallest breakpoint value λ of function $F(\lambda) = \min_{\mathbf{x}}[P(\mathbf{x}) - \lambda Q(\mathbf{x})]$ (see Fig. 1(b)). In addition to $\mathbf{x}^*$, the ES algorithm can compute consecutive larger breakpoints λ_i and obtain a sequence $\mathbf{x}_0, \mathbf{x}_1, \ldots$ such that $\mathbf{x}_i$ is an optimal solution for $\lambda_{i+1} \leq \lambda \leq \lambda_{i+2}$. Interestingly, these consecutive solutions are also related to the ratio optimization problem and they can be useful for image segmentation and multiview reconstruction (see Sec. 4). It is not difficult to show that (see Fig. 1(b)):

Proposition 3.1. *(a) The sequences* $Q(\mathbf{x}_0), Q(\mathbf{x}_1), \ldots$ *and* $R(\mathbf{x}_0), R(\mathbf{x}_1), \ldots$ *are strictly increasing. (b) Configuration* $\mathbf{x}_i$ *is an optimal solution of* $\min_{\mathbf{x}\,:\,Q(\mathbf{x}) \geq C} R(\mathbf{x})$ *where* $C = Q(\mathbf{x}_i)$.

While solving the constrained problem in (b) for arbitrary values of C appears to be a difficult problem, it can be solved for specific values $Q(\mathbf{x}_0), Q(\mathbf{x}_1), \ldots$.

Specifying a lower bound on the denominator $Q(x)$ is a useful additional tool in finding optimal ratio segments under extra constraint on segment size (see Fig. 3). For exam-

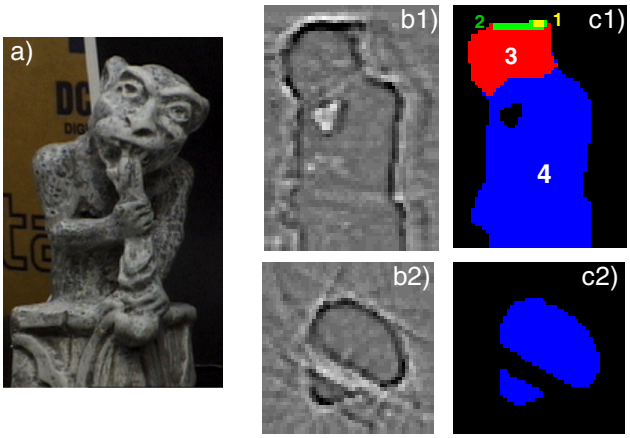


Figure 4. **Volumetric Multiview Stereo.** Optimizing the ratio of flux (of *photoconsistency gradients* [6]) over Euclidean surface area for "gargoyle" data set (courtesy of K. Kutulakos and S. Seitz). a) Sample camera view. b) Divergence of photoconsistency gradients (volumetric potential). c) Reconstruction results. Representative dominant solutions grow from the highest contrast region 1 (yellow). However, the growth is not always monotonic, e.g. there is an intermediate solution between 3 and 4 with no hole in the object.

ple, when we minimize flux over length/area, often the first segmentation corresponds to the highest "contrast" object of small size. As λ increases, we find optimal solutions of the highest contrast among all segments of larger size. Our segmentation and reconstruction experiments (Sec. 4) show that the optimal shape grows in size incorporating parts of smaller contrast as the boundary length increases.

4. Experimental results

4.1. Ratio optimization in 3D

Fig. 4 shows **volumetric multiview stereo** reconstruction obtained by optimizing the ratio of *flux of photoconsistency gradients* [6] over the surface area. Thus, the optimal result has the highest density of photoconsistency gradients. However, the highest "contrast" solution just selects one "ear" where the gradients are the highest due to visibility from the largest number cameras. We reconstruct the head and then the whole "gargoyle" by adding larger size constraint (see Proposition 3.1). Similarly to the synthetic examples at the bottom of Figure 3, the consecutive breakpoint solutions for ratio (3) grew monotonically (from yellow to blue) as the surface area increased. Note that the right breakpoint can be automatically found if a reasonable lower bound on the model's size is known. One may also select the last breakpoint just before monotonicity breaks. Alternatively, the parametric maxflow technique allows a user to efficiently select a good breakpoint manually.

We also ran experiments for **surface fitting** to a cloud of laser scanned points based on the minimum ratio of *flux of the estimated surface normals* [26] over surface area. Results can be found in [24].

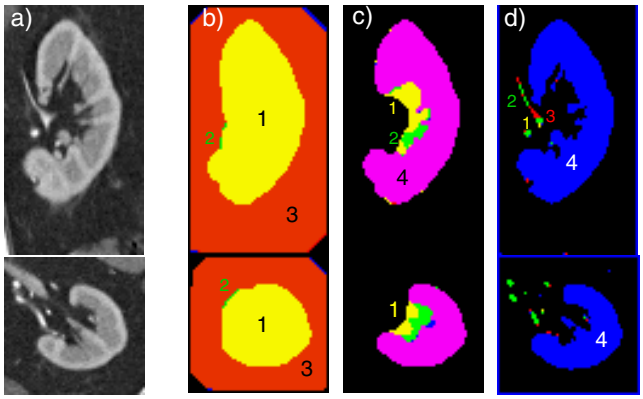


Figure 5. **Image Segmentation in 3D.** Optimizing various ratios for 3D medical image data ("kidney" CT). a) Two orthogonal image slices. b) Ratio of Riemannian area (based on image gradients) over volume. c) Ratio of data likelihood (intensity model) over Riemannian area. d) Ratio of flux (of image gradients) over Euclidean surface area.

Fig. 5 (b,c,d) shows **volumetric image segmentation** results optimizing ratios (1,2,3), correspondingly. Optimizing flux of image gradients over surface area (d) is equivalent to finding the segment with the largest average contrast on its boundary. The colors (from yellow to blue) indicate representative larger optimal segments as the lower bound on surface area increases at consecutive breakpoints (see Prop. 3.1). As in earlier examples, ratio (3) has no bias to any particular shape and shows good alignment to details. In contrast, optimizing the ratio of image weighted (Riemannian) surface area over volume (b) shows a strong bias to circles coming from its close relationship with the isoperimetric problem. The smallest ratio is achieved by the smooth yellow blob around the kidney. Larger breakpoint solutions monotonically switch to larger segments approximating Cheeger sets for rectangles [33]. Replacing volume by color model likelihood as in (c) improves the oversmoothing bias but many details are still undersegmented.

Image segmentation results in Fig. 5(b) also illustrate typical dominant solutions for a closely related multiview reconstruction technique using volumetric ballooning. Surface functionals combining photoconsistency-weighted area with volumetric ballooning are common in 3D reconstruction (e.g. [35, 27]). Choosing λ to balance these two terms is equivalent to finding some breakpoint for the constrained isoperimetric (Cheeger) problem (1). As noted in [10], the ratio of surface area to volume (1) has bias to large segments[5]. Prop. 3.1 shows that the smallest size breakpoint solution (e.g. yellow segment in Figure 5(b)) is the one that minimizes ratio (1). This solution is likely to be the most appropriate for multiview reconstruction methods using volumetric ballooning.

[5]Since surface area grows quadratically w.r.t. object diameter and volume grows cubically, the ratio of the two is smaller for larger segments.

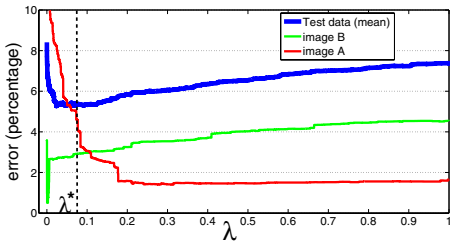


Figure 6. **Image segmentation - error statistics.** The number of misclassified pixels (error) varies with respect to λ. The optimal λ does also vary considerably among test images (images A and B are shown in [24]).

4.2. Choosing an optimal λ

We consider now the problem of segmenting color images using a regularization energy $E(\mathbf{x}) = \lambda \sum_p D_p(x_p) + \sum_{pq} V_{pq}(x_p, x_q)$.

The goal is to demonstrate that (i) discriminative learning of λ on a training set with ground truth segmentation yields a reasonable value for many test images, and (ii) the learned value λ^* cannot work well for all images, so the problem may need to be solved for different λ's during testing. These points have been raised in previous work, e.g. [2]. The paper [2] uses a discriminative procedure for learning λ, however they do not specify how the parametric problem is solved.

The data likelihoods D_p of the energy come from learned gaussian mixture models of foreground and background, as in [2]. (We used 10 Gaussians with full covariance for each region.) We used the dataset of 50 images with ground truth segmentations introduced in [2][6]. We split it randomly into 25 training and 25 test images. We used hard constraints obtained by dilating ground truth segmentation by a fixed amount which gives a "trimap" (see example in Fig. 2 in [2]). The segmentation error is defined as the percentage of misclassified pixels within the trimap.

The value of λ^* that we learned on the training data using parametric maxflow algorithm was $\lambda^* = 0.074$. Fig. 6 shows the error on the test data set. We see that there is a range of $\lambda \in [0.023, 0.12]$ where the error is equally low, i.e. between 5.27% and 5.36%. The error on the test data set is 5.34% for the learned λ^* and if we were to choose the optimal fixed λ of 0.023 we would get a test error of 5.27%. This shows that the training set was probably sufficiently large and there was no overfitting.

At the same time, segmentation results can be considerably improved by choosing for each image a different, optimal λ. This gives a test error of 3.79%, which is considerably lower than 5.27%. For 50% of the test images choosing image-specific λ improved the error over learned value λ^* by more than 0.5%. Two of these examples are shown in [24], where the optimal λ lies even outside the range of $[0.023, 0.12]$; the corresponding error curves are in Fig. 6.

Thus, it is desirable to run parametric maxflow during testing and let, for example, the user choose best λ. Unfortunately, our implementation was not fast enough for interactive segmentation[7]. However, potentially the speed can be improved significantly by solving the parametric problem via shortest path techniques. (The general scheme of the ES method would still be applicable). In order to do this, we could first compute segmentation for λ^* and fix the topology. This is left as a future work.

4.3. Cosegmentation using TRGC

We tested the parametric maxflow algorithm for the *trust region graph cuts* (TRGC) method in [30] for the problem of cosegmenting two images. The key optimization problem of the cosegmentation task is as follows: given an image where each pixel is assigned to a certain bin and a target histogram over bins is given, compute segmentation $\mathbf{x}$ which minimizes function $E(\mathbf{x}) = E_{MRF}(\mathbf{x}) + E_{hist}(\mathbf{x})$. The first term is a contrast-dependent discontinuity cost, and the second term is the $L1$ norm of the difference between the histogram of the region $\{u \mid x_u = 1\}$ and the target histogram.

The TRGC method maintains vector $\mathbf{y} \in \mathbb{R}^{\mathcal{V}}$ and configuration $\mathbf{x}$ such that linear function $\mathbf{y}^T\mathbf{z} + \text{const}$ is an approximation of the global term $E_{hist}(\mathbf{z})$ and $\mathbf{x}$ is a global minimum of $E_{MRF}(\mathbf{z}) + \mathbf{y}^T\mathbf{z}$. In each iteration, new approximation vector $\tilde{\mathbf{y}}$ is chosen, based on current configuration $\mathbf{x}$. (This approximation is exact for all $\mathbf{z}$ that differ from $\mathbf{x}$ by at most 1 pixel). Let $\mathbf{y}^\lambda = \lambda\tilde{\mathbf{y}} + (1-\lambda)\mathbf{y}$ be the interpolation between $\mathbf{y}$ and $\tilde{\mathbf{y}}$. The TRGC method computes minimum $\mathbf{x}^\lambda$ of approximation $E_{MRF}(\mathbf{z}) + (\mathbf{y}^\lambda)^T\mathbf{z}$ for different λ's in $[0, 1]$, and selects pair $(\mathbf{y}^\lambda, \mathbf{x}^\lambda)$ corresponding to the smallest true energy $E(\mathbf{x}^\lambda)$. Note, the energy will not go up since $\lambda = 0$ corresponds to current pair $(\mathbf{y}, \mathbf{x})$.

We tested three strategies of searching interval $[0, 1]$:

A: [Same as in [30]] Start with $\lambda = 1$, keep halving it until one of the following happens: (a) $\mathbf{x}^\lambda = \mathbf{x}$; (b) $\lambda < 10^{-3}$; or (c) energy $E(\mathbf{x}^\lambda)$ is larger compared to previous λ, and the energy for the previous λ was smaller than $E(\mathbf{x})$.

B: Similar to **A**, only next λ in $[0, \lambda]$ is chosen as in the ES method. (The last λ will be the smallest breakpoint, unless the search is stopped earlier).

C: Compute all breakpoints and solutions in $[0, 1]$.

The average results on a data-set with 20 images (part of the ground truth data-set introduced in [2]) were as follows (for an example see [24]):

[6] http://research.microsoft.com/vision/cambridge/i3l/segmentation/GrabCut.htm

[7] For the range of $\lambda \in [0, 1]$ computing all cuts took $24.5 \in [3.3, 61]$ secs (average over 50 images; the interval shows minimum and maximum values), compared to 0.9secs on average for λ^*. The number of cuts was $425 \in [81, 1422]$. For a more reasonable range $\lambda \in [0.023, 0.12]$ (see Fig. 6) the running time was $7 \in [0.64, 16.3]$ secs, and the number of cuts was $110 \in [3, 382]$.

strategy	final energy	misclass. error	time
A	1431.7	4.42%	0.37 secs
B	1111.6	3.21%	0.77 secs
C	1087.3	3.05%	10.5 secs

Here the target histogram is given by the ground truth segmentation. We refer to [24] for details of initialization and further results.

5. Conclusions

We showed that parametric maxflow algorithm from combinatorial optimization is useful for regularization-based N-D applications in computer vision (segmentation, cosegmentation, multi-view stereo, surface fitting, etc). The algorithm can efficiently find a sequence of solutions for all values of the trade-off parameter λ. The best solution from this sequence of *dominant solutions* can be selected based on any global criteria which otherwise would be practically infeasible. Examples of such global criteria include minimization of normalized (ratio) functionals, minimum required size, surface total curvature, proximity to certain prior shape, or user selection. The algorithm can be also used to accelerate parameter learning.

Acknowledgements We thank Victor Lempitsky for providing divergence data for the multiview reconstruction and surface fitting experiments. Olga Veksler pointed out several important related papers on ratio optimization, and noticed that the method in [12] can be reduced to monotonic parametric maxflow. We also thank Leo Grady for useful discussions regarding the work of Gilbert Strang.

References

[1] M. Babenko and A. Goldberg. Experimental evaluation of a parametric flow algorithm. Technical Report MSR-TR-2006-77, Microsoft Research, June 2006. 3

[2] A. Blake, C. Rother, M. Brown, P. Perez, and P. Torr. Interactive image segmentation using an adaptive GMMRF model. In *ECCV*, 2004. 1, 7

[3] Y. Boykov and V. Kolmogorov. Computing geodesics and minimal surfaces via graph cuts. In *ICCV*, Oct. 2003. 2, 4

[4] Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *PAMI*, 26(9), September 2004. 2

[5] Y. Boykov, V. Kolmogorov, D. Cremers, and A. Delong. An integral solution to surface evolution PDEs via geo-cuts. In *ECCV*, 2006. 1, 2

[6] Y. Boykov and V. Lempitsky. From photohulls to photoflux optimization. In *BMVC*, 2006. 1, 6

[7] P. J. Carstensen. *The complexity of some problems in parametric linear and combinatorial programming*. PhD thesis, Dept. of Math., The University of Michigan, 1983. 2

[8] A. Chambolle. Total variation minimization and a class of binary MRF models. In *EMMCVPR*, November 2005. 1, 3

[9] J. Cheeger. A lower bound for the smallest eigenvalue of the Laplacian. *Problems in Analysis*, pages 195–199, 1970. 4

[10] I. J. Cox, S. B. Rao, and Y. Zhong. Ratio regions: a technique for image segmentation. In *ICPR*, 1996. 2, 4, 6

[11] J. Darbon and M. Sigelle. A fast and exact algorithm for total variation minimization. In *Pattern Recognition and Image Analysis: 2nd Iberian Conf.*, LCNS. Springer, June 2005. 1

[12] P. Das, O. Veksler, V. Zavadsky, and Y. Boykov. Semiautomatic segmentation with compact shape prior. In *Canadian Conference on Computer and Robot Vision*, 2006. 1, 8

[13] W. Dinkelbach. On nonlinear fractional programming. *Manage. Sci.*, 13:492–498, 1967. 3

[14] M. J. Eisner and D. G. Severence. Mathematical techniques for efficient record segmentation in large shared databases. *J. ACM*, 23(4):619–635, Oct. 1976. 1, 2, 3

[15] G. Gallo, M. D. Grigoriadis, and R. E. Tarjan. A fast parametric maximum flow algorithm and applications. *SIAM J. Computing*, 18:30–55, 1989. 1, 3

[16] L. Grady and E. Schwartz. Isoperimetric graph partitioning for image segmentation. *PAMI*, 28(3):469–475, 2006. 2, 4

[17] D. Gusfield. *Sensitivity analysis for combinatorial optimization*. PhD thesis, UC Berkeley, 1980. 1, 2, 3

[18] D. S. Hochbaum. An efficient algorithm for image segmentation, Markov Random Fields and related problems. *J. ACM*, 48:2:686–701, July 2001. 1

[19] I. Jermyn and H. Ishikawa. Globally optimal regions and boundaries as minimum ratio weight cycles. *PAMI*, 23(10), Oct. 2001. 1, 2, 4, 5

[20] B. Kawohl and T. Lachand-Robert. Characterization of Cheeger sets for convex subsets of the plane. *Pacific J. Math*, 225:103–118, 2006. 4

[21] R. Kimmel and A. M. Bruckstein. On regularized Laplacian zero crossings and other optimal edge integrators. *IJCV*, 53(3):225–243, 2003. 1

[22] P. Kohli and P. H. S. Torr. Efficiently solving dynamic Markov random fields using graph cuts. In *ICCV*, 2005. 2

[23] V. Kolmogorov and Y. Boykov. What metrics can be approximated by geo-cuts, or global optimization of length/area and flux. In *ICCV*, Oct. 2005. 1, 2, 4

[24] V. Kolmogorov, Y. Boykov, and C. Rother. Applications of parametric maxflow in computer vision. Technical report, Microsoft Research, 2007. 2, 4, 6, 7, 8

[25] S. Kumar, J. August, and M. Hebert. Exploiting inference for approximate parameter learning in discriminative fields: An empirical study. In *EMMCVPR*, 2005. 1

[26] V. Lempitsky and Y. Boykov. Global optimization for shape fitting. In *CVPR*, 2007. 6

[27] V. Lempitsky, Y. Boykov, and D. Ivanov. Oriented visibility for multiview reconstruction. In *ECCV*, 2006. 6

[28] N. Megiddo. Combinatorial optimization with rational objective functions. *Math. of OR*, 4(4):414–424, 1979. 3

[29] S. B. Patkar and H. Narayanan. Improving graph partitions using submodular functions. *Discrete Applied Mathematics*, 131:535–553, 2003. 2

[30] C.Rother, V.Kolmogorov, T.Minka, and A.Blake. Cosegmentation of image pairs by histogram matching - incorporating a global constraint into MRFs. In *CVPR*, 2006. 2, 7

[31] J. Shi and J. Malik. Normalized cuts and image segmentation. *PAMI*, 22(8):888–905, 2000. 2

[32] H. S. Stone. Critical load factors in two-processor distributed systems. *T. Soft. Eng.*, SE-4(3):254–258, May 1978. 1, 3

[33] G. Strang. Maximum flows and minimum cuts in the plane. *Journal of Global Optimization*, 2007, to appear. 4, 6

[34] A. Vasilevskiy and K. Siddiqi. Flux-maximizing geometric flows. *PAMI*, 24(12):1565–1578, 2002. 1

[35] G. Vogiatzis, P. H. S. Torr, and R. Cipolla. Multi-view stereo via volumetric graph-cuts. In *CVPR '05(2)*, 2005. 6

[36] S. Wang and J. M. Siskind. Image segmentation with ratio cut. *PAMI*, 25(6):675–690, June 2003. 2, 4

[37] B. A. Zalesky. Network flow optimization for restoration of images. *J. Appl. Math.*, 2(4):199–218, 2002. 1

[38] B. Zhang, J. Ward, and Q. Feng. A simultaneous maximum flow algorithm with vertex balancing. Technical Report HPL-2004-121, HP Labs, 2004. 3