Discrete minimum ratio curves and surfaces

Fred Nicolls
University of Cape Town
Cape Town, South Africa
fred.nicolls@uct.ac.za

Philip H. S. Torr
Oxford Brookes University
Oxford, UK
philiptorr@brookes.ac.uk

Abstract

Graph cuts have proven useful for image segmentation and for volumetric reconstruction in multiple view stereo. However, solutions are biased: the cost function tends to favour either a short boundary (in 2D) or a boundary with a small area (in 3D). This bias can be avoided by instead minimising the cut ratio, which normalises the cost by a measure of the boundary size. This paper uses ideas from discrete differential geometry to develop a linear programming formulation for finding a minimum ratio cut in arbitrary dimension, which allows constraints on the solution to be specified in a natural manner, and which admits an efficient and globally optimal solution. Results are shown for 2D segmentation and for 3D volumetric reconstruction.

1. Introduction

Interactive graph-based image segmentation algorithms are useful and powerful. Boykov and Jolly [2] popularised the approach of having a user specify a number of interior and exterior seed points in a segmentation, and of using a graph cut algorithm to find the best solution based on a cost function derived from image data. The cost is defined in terms of regions and boundaries simultaneously: a preference for each region element being inside or outside the object is found using models obtained from user input, while boundary elements are preferred if they coincide with image structures that are likely to be edges.

The method also works in 3D, largely because there is a one-to-one correspondence between cuts in the constructed graph and valid object boundaries. However, costs are required to be positive (or reducible to positive via reparameterisation, which is possible if the objective function is submodular [9]), so there is a shrinking bias in that small objects with short boundaries are favoured. Vogiatzis et al. [18] propose a remedy by including a ballooning term for the case of volumetric reconstruction. However, while modifications of this form exert a general outward pressure on solutions they are poorly directed, and it is difficult to specify an appropriate weight for the normalisation since region and boundary costs are not obviously commensurate.

To mitigate these shortcomings methods have been pursued that instead minimise a ratio cost, where a measure of the size of the solution region is included in the denominator. For example, for the 2D case Cox et al. [4] normalise a boundary cost by the area of the segmented object, and Jermyn and Ishikawa [8] propose a ratio cost of boundary flux per unit length. Tractable algorithms for performing the minimisation are provided. A unified treatment of ratio functionals is provided by Kolmogorov et al. [10], along with a solution method based on parametric maximum flow. In all of these cases the ratio functionals have no apparent size bias.

The approach adopted in this work is to use a ratio of total cost to total length, which in the case of boundaries relates to the objective of minimising cost per unit length of boundary in the solution [19]. It has been shown that the resulting minimum ratio cycle problem can be solved in the 2D case, but the methods used do not apparently extend to 3D. Specifying the boundary conditions that determine the solution and eliminate the trivial null solution can also be problematic.

In addressing a related graph-based method, Grady [6, 7] has highlighted the importance of using discrete differential geometry in formulating segmentation problems. He makes reference to earlier work by Sullivan [15]. Specifically, there is an intimate relationship between regions and boundaries in any valid solution to a segmentation problem, and these can be expressed in terms of incidence matrices of associated geometric entities. Mattiussi [12] elaborates lucidly on the principles and structures involved. In short, the tools of differential geometry can be used to add structure to a higher-dimensional segmentation that is analogous to the structure added by graph planarity in the 2D case. The geometric requirements of a valid segmentation can be added to the problem formulation and make it much easier to solve.

In this work we provide a general formulation for globally minimising a large class of ratio cost functions that effectively subsumes many of those previously suggested. For ease of interpretation we restrict the space of admissible solutions in the discretisation, although this restriction can be removed at the cost of permitting solutions that less obviously correspond to regions with boundary. The method permits ratio terms based on both region and boundary elements in both the numerator and denominator, and there is no restriction on the sign of numerator costs. Region and boundary constraints can be expressed explicitly, and the solution is guaranteed to have the required topology in terms of the interior and exterior of the segmented object. We demonstrate the use of this formulation for 2D and 3D segmentation problems, and show that the absence of a shrinking bias makes it easier to obtain the desired solution.

The main contributions of this work are the following:

- we show how discrete differential geometry can be used to impose consistent interior/exterior constraints on segmentations in a ratio cut framework,
- we present an efficient algorithm for finding the minimum ratio cut that works in dimensions higher than two, and that can be extended to incorporate other Markov random field (MRF) models,
- we demonstrate that the addition of geometric constraints permits tractable solutions to the minimum cut problem with arbitrary (*i.e*. positive or negative) weights.

Section 2 presents the differential geometry principles that are relevant to the problem. Section 3 discusses linear fractional programming, which is important in the ratio cut problem. The optimisation solved in this paper is formulated in Section 4, along with requirements for the discretisation used. A fast minimum cut maximum flow implementation is given in Section 5, along with an application in 2D. In Section 6 we show how the method can be used to solve an image-based cost function for 3D reconstruction, namely squared error per unit projected area, which does not exhibit a shrinking bias and can be globally optimised (admittedly for assumed known visibility). Section 7 discusses results and concludes.

2. Discrete geometry

Graph algorithms generally make no assumptions on topology. However, if there is additional structure in the graph then this can almost certainly be used to develop specialised algorithms with lower computational complexity. For example, the complexity of the maximum flow algorithm for general graphs can be considerably reduced if the graph is planar [1]. In other instances, a restriction in the topology can make a hard optimisation problem tractable.

Planar graphs admit tractable algorithms primarily due to the presence of strong geometric duality relationships. A graph drawn in the plane induces a set of faces, with exactly two faces adjacent to any edge. The dual of a planar graph can be constructed procedurally: a dual node is created for every face or region (including one dual node for the exterior face), and dual nodes are connected by a dual edge if the corresponding faces share an edge in the primal graph. The dual graph is also planar, and the dual of the dual is the primal. Figure 1 demonstrates the equivalence between a path in a primal planar graph and a cut in the dual graph.

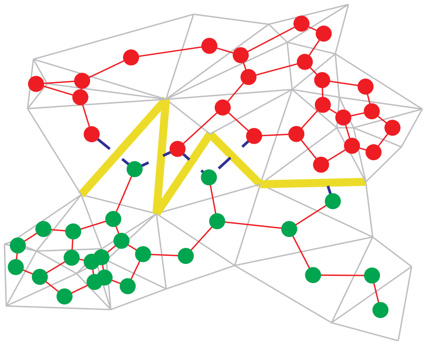


Figure 1. Duality between cut and path for a planar graph. A path (thick solid yellow line) is defined by the dual edges cut (dashed blue lines) between two (dual) node sets. The dual node for the exterior face is not shown.

If edges are directed then the faces can be coherently oriented: one will be to the left of the edge, and one to the right. For any directed cycle in the graph, this makes it possible to differentiate between faces inside the cycle and those outside. We use the convention that the interior of the object is to the left of a directed edge, so boundaries enclose regions with an anticlockwise orientation.

The tools of discrete differential geometry provide a formalism for this duality [14]. The relevant geometric entity is a simplicial complex (also just referred to as a complex), which can be described by a set of incidence relationships on lower-dimensional component simplices. For the 2D case just presented the simplices are the faces, edges, dual edges, and vertices in the planar graph. Vertices are 0-dimensional simplices, edges are 1-dimensional simplices, and faces are 2-dimensional simplices. The incidences specify relationships between these simplices. For example, for the triangular mesh in Figure 1 each face (2D simplex) is incident to three edges (1D simplices), and each edge is incident on two endpoint vertices (0D simplices). Simplices can be assigned an orientation, and the signs of the incidence relationships can be specified consistently [12].

The incidence relationships between simplices in a complex can be expressed algebraically. The vertex-edge incidence can be represented by the node-arc incidence matrix $\mathbf{A}$, where each row corresponds to a vertex and each column to an oriented edge. This matrix has nonzero elements only where a directed edge is incident to a vertex, and a sign convention can be adopted where the value is $+1$ where the edge leaves a vertex and -1 where it enters. This matrix can also be interpreted as a differential operator that takes a set of edges (indicated by a vector $\mathbf{y}$ with elements in $\{0, 1\}$) to a set of nodal boundary vertices (represented by $\mathbf{z}$) through the relation $\mathbf{z} = \mathbf{A}\mathbf{y}$. If the set of edges constitutes a simple directed path then there will be exactly two nonzero elements in $\mathbf{z}$: one with a value $+1$ corresponding to the vertex that the path leaves, and one with -1 where the path enters. In terms of duality, $\mathbf{A}^T$ is the incidence matrix of dual faces to dual edges.

Similarly, for the 2D case, let $\mathbf{C}$ represent the edge-face incidence matrix with a value $+1$ where an edge is incident to a face and is coherently oriented and -1 where incident and anti-coherently oriented. This matrix also represents the differential operator that maps faces to directed boundary edges. Furthermore, $\mathbf{C}^T$ is the incidence matrix of dual nodes to dual edges. In the planar case $\mathbf{C}$ is a cycle matrix, and each column can be considered to relate to an oriented elementary cycle [16]. Since the boundary of a set of faces is a set of closed cycles and cycles have no boundary vertices, the orthogonality relation $\mathbf{AC} = \mathbf{0}$ must always hold. Grady [6, 7] provides a detailed treatment of the subject, with clear examples.

Most importantly, the notion of a simplicial complex can be used to extend planarity and planar duality to higher dimensions. In 3D, the geometric entities in the primal complex are cycles (or cells), faces, edges, and vertices. Cycles correspond to voxels, there are exactly two voxels adjacent to every face (which has an orientation), faces are bounded by edges (also oriented), and each edge is incident on exactly two vertices. Referring to the dual complex, each cell corresponds to a dual node, each face to a dual edge, each edge to a dual face, and each vertex to a dual cell.

The incidence relationships of voxels (cycles) to faces are the same as those of dual nodes to dual edges. Since dual nodes and dual edges constitute a valid graph, their incidence matrix is totally unimodular. The cycle-face incidence matrix is therefore also totally unimodular.

If the primal complex occupies a bounded region, then boundary conditions must be taken into account when constructing the dual. A consistent method of doing this is to consider the outer region to be an additional cycle or face, which becomes a dual node when the dual is constructed.

3. Linear fractional cost functions

Frenk and Schabile [5] give a complete account of linear fractional cost functions in a general setting.

In the ratio cut formulation we consider the cost function

$$t(\mathbf{x}) = \frac{\mathbf{a}^T\mathbf{x}}{\mathbf{b}^T\mathbf{x}} \tag{1}$$

and the requirement is to minimise it over all $\mathbf{x} \in \mathcal{X}$, where $\mathcal{X}$ is the set of feasible solutions.

Lawler [11, p. 94] describes a method of solving the problem, which is elaborated upon by Meggiddo [13]. The algorithm involves minimising

$$v(t, \mathbf{x}) = (\mathbf{a} - t\mathbf{b})^T\mathbf{x} \tag{2}$$

over admissible $\mathbf{x}$ for a sequence of chosen values of t. The method requires that $\mathbf{b} \geq \mathbf{0}$ and $\mathbf{b}^T\mathbf{x} \neq 0$ for all $\mathbf{x}$, and an initial finite interval $[t_l, t_u]$ bounding the minimum ratio value must be known. The initial upper bound t_u can be provided by any feasible solution $\mathbf{x}$: an obvious choice is to use the solution that minimises $\mathbf{a}^T\mathbf{x}$. If $\mathbf{a} \geq \mathbf{0}$ then the lower bound can be taken as $t_l = 0$.

Suppose that we select a value of $t_0 \in [t_l, t_u]$. Let

$$\mathbf{x}^* = \arg\min_{\mathbf{x}} v(t_0, \mathbf{x}) \tag{3}$$

with minimum attained value $v(t_0, \mathbf{x}^*)$. Three cases can occur.

If $v(t_0, \mathbf{x}^*) = 0$ then $\mathbf{x}^*$ satisfies $\mathbf{a}^T\mathbf{x}^*/\mathbf{b}^T\mathbf{x}^* = t_0$. However, since $\min_{\mathbf{x}}(\mathbf{a} - t_0\mathbf{b})^T\mathbf{x} = 0$ it must be true that $\mathbf{a}^T\mathbf{x}/\mathbf{b}^T\mathbf{x} \geq t_0$ for all $\mathbf{x}$, so t_0 is a lower bound on the minimum ratio. Since $\mathbf{x}^*$ attains this minimum it is a valid solution, the minimum ratio is t_0, and the algorithm can terminate.

If $v(t_0, \mathbf{x}^*) < 0$ then $\mathbf{a}^T\mathbf{x}^*/\mathbf{b}^T\mathbf{x}^* < t_0$. The value t_0 is larger than the minimum ratio value, since $\mathbf{x}^*$ yields a ratio that is strictly smaller than t_0. Thus $\mathbf{a}^T\mathbf{x}^*/\mathbf{b}^T\mathbf{x}^*$ is a new upper bound on the ratio, and we can apply the update $t_u \leftarrow \mathbf{a}^T\mathbf{x}^*/\mathbf{b}^T\mathbf{x}^*$. Similarly, if $v(t_0, \mathbf{x}^*) > 0$ then

$$\mathbf{a}^T\mathbf{x}/\mathbf{b}^T\mathbf{x} \geq t_0 + v(t_0, \mathbf{x}^*)/\mathbf{b}^T\mathbf{x} \tag{4}$$

for all valid $\mathbf{x}$. Thus t_0 is smaller than the minimum ratio value and the lower bound can be updated: $t_l \leftarrow t_0$.

The approach for obtaining a solution is to select a value $t_0 \in (t_l, t_u)$, minimise Equation 3 over $\mathbf{x} \in \mathcal{X}$, and either terminate with the required solution or update the bounds and repeat with a value of t_0 in the new interval. Since the length of the interval $[t_l, t_u]$ is strictly decreasing and there are finitely many admissible solutions the algorithm will terminate.

All that remains is to specify the rule for the selection of $t_0 \in [t_l, t_u]$ at each iteration. In practice it makes little difference: the upper bound value, the arithmetic mean, and the geometric mean all work in the applications presented and result in convergence after a small number of iterations.

The elements of the weight vector in the minimisation of $v(t_0, \mathbf{x})$ in Equation 3 can have arbitrary sign. It is this factor that makes standard maximum flow algorithms inappropriate for the task, since flow capacities are required to be nonnegative. It is the structure of the mesh and the duality provided by differential geometry that makes the problem tractable.

4. Problem formulation

Following Sullivan [15] and Grady [7, 6], the minimum weight directed path problem with endpoint constraints can be written as

$$\min \ \mathbf{w}_y^T \mathbf{y} \ \text{ subject to } \ \mathbf{A}\mathbf{y} = \mathbf{p}, \tag{5}$$

where $\mathbf{y}$ is an indicator vector for primal edges in a complex. The signed node-edge incidence matrix $\mathbf{A}$ is also the boundary operator for edges: for any valid path, indicated by $\mathbf{y}$ with elements in $\{0, 1\}$, the quantity $\mathbf{A}\mathbf{y}$ has one element per primal node and is only nonzero at the path endpoints. A value of $+1$ or -1 indicates that the directed path either originates from or terminates at the corresponding node.

The vector $\mathbf{p}$ is used to enforce the desired endpoint constraints on the solution path. In most cases of interest it will be a vector of all zeros, with a single $+1$ element at the required start node of the path and a -1 element at the end node. Since the constraint matrix $\mathbf{A}$ is the node-arc incidence matrix of a directed graph it is totally unimodular, so for $\mathbf{p}$ as specified the solutions will be integral with $y_i \in \{0, 1\}$ for each i, where y_i is the ith element of $\mathbf{y}$. Furthermore, the method extends naturally to the problem of finding a minimal surface in 3D if $\mathbf{A}$ is taken to be the edge-face incidence matrix of a 3D complex, $\mathbf{y}$ is an indicator vector for faces, and the constraint vector $\mathbf{p}$ is assigned accordingly.

The problem we consider in this paper involves closed directed cycles rather than paths. Since a closed cycle has no endpoints, the appropriate constraint is $\mathbf{A}\mathbf{y} = \mathbf{0}$ and in the 2D case specifies that the indegree and outdegree of directed edges in the solution must be equal for every node. Algebraically, $\mathbf{y}$ must lie in the nullspace of $\mathbf{A}$. However, since this constraint is invariant to scaling of $\mathbf{y}$, additional restrictions are required to ensure that for each i we have $y_i \in \{0, 1\}$ at the solution.

Let $\mathbf{C}$ be the edge-face incidence matrix of the primal complex, and for clarity assume that the problem being addressed is segmentation in 2D. Define an indicator vector $\mathbf{x}$ with one element per region, where $x_j = +1$ denotes that region j is in the interior of the object being semented, or $x_j = 0$ otherwise. Since $\mathbf{C}$ is the boundary operator for regions, the oriented edges on the region boundary, represented by $\mathbf{y}$, can be found using the relation $\mathbf{y} = \mathbf{C}\mathbf{x}$. Note that the orthogonality between paths and cycles implies that $\mathbf{A}\mathbf{y} = \mathbf{A}\mathbf{C}\mathbf{x} = \mathbf{0}$, as required. For any consistent region-boundary pair $(\mathbf{x}, \mathbf{y})$, with $x_j \in \{0, 1\}$, it must be true that $\mathbf{y}_i \in \{0, 1\}$.

The overall optimisation problem addressed in this paper can be written as

$$\min \ \frac{\mathbf{n}_x^T \mathbf{x} + \mathbf{n}_y^T \mathbf{y}}{\mathbf{d}_x^T \mathbf{x} + \mathbf{d}_y^T \mathbf{y}} \ \text{ subject to } \ \mathbf{y} = \mathbf{C}\mathbf{x},$$
$$\mathbf{x}_1 \le \mathbf{x} \le \mathbf{1} - \mathbf{x}_0, \quad \mathbf{y}_1 \le \mathbf{y} \le \mathbf{1} - \mathbf{y}_0. \tag{6}$$

Here $\mathbf{x}_1$ is a zero-one indicator vector for region variables forced to value 1 (interior) and $\mathbf{x}_0$ indicates regions forced to zero (exterior). The indicators $\mathbf{y}_1$ and $\mathbf{y}_0$ are defined similarly but for edges being forced to be either in or not in the solution respectively. The elements of $\mathbf{n}_x$, $\mathbf{n}_y$, $\mathbf{d}_x$, and $\mathbf{d}_y$ can be positive or negative, but the denominator must be positive for all feasible solutions.

Defining

$$\mathbf{w}_n = \mathbf{n}_x + \mathbf{C}^T \mathbf{n}_y \quad \text{and} \quad \mathbf{w}_d = \mathbf{d}_x + \mathbf{C}^T \mathbf{d}_y$$

the explicit dependence on $\mathbf{y}$ can be removed, yielding the problem

$$\min \ \frac{\mathbf{w}_n^T \mathbf{x}}{\mathbf{w}_d^T \mathbf{x}} \ \text{ subject to}$$
$$\mathbf{x}_1 \le \mathbf{x} \le \mathbf{1} - \mathbf{x}_0, \quad \mathbf{y}_1 \le \mathbf{C}\mathbf{x} \le \mathbf{1} - \mathbf{y}_0. \tag{7}$$

This is in the form of Equation 1, where the inequality constraints describe the feasible set $\mathcal{X}$. It is the fact that edge costs can be transferred to region costs that makes the optimisation tractable. The constraints in Equation 7 can be written as

$$\begin{pmatrix} -\mathbf{I} \\ \mathbf{I} \\ -\mathbf{C} \\ \mathbf{C} \end{pmatrix} \mathbf{x} \le \begin{pmatrix} -\mathbf{x}_1 \\ \mathbf{1} - \mathbf{x}_0 \\ -\mathbf{y}_1 \\ \mathbf{1} - \mathbf{y}_0 \end{pmatrix}. \tag{8}$$

The right hand side of this inequality is integral. Also, since the matrix $\mathbf{C}$ is the transpose of the node-arc incidence matrix of the dual complex it is totally unimodular. It is quite simple to show that total unimodularity of a matrix is retained over duplicating a row, appending a unit row, and changing the sign of a row, so the matrix on the left of Equation 8 is totally unimodular [11]. The system therefore has integral extrema, and there must be an optimal solution $\mathbf{x}$ with $x_i \in \{0, 1\}$.

Figure 2 shows two possible meshes in 2D. Each of them supports the topology of a closed cycle, in the sense that they can discretely approximate any directed cycle in terms of position and orientation. (Note that not all meshes have this property: if all the directed edges in a mesh of Figure 2 were pointing east and south, it would not be possible to represent the directed portion of a cycle in the east-to-west direction.) The orientation of the edges in terms of the outward normal from the segmentation interior is indicated by the red crossing arrows, which also indicate the edges in the dual complex. Large red nodes correspond to cells or cycles in the primal mesh, or nodes in the dual.

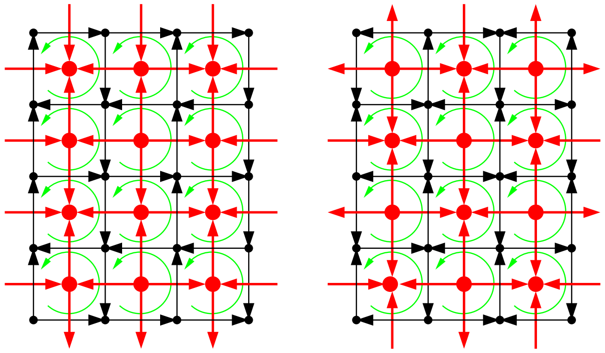


Figure 2. Possible oriented 2D meshes for use in minimum ratio cut. Oriented boundaries are composed of directed edge elements (black arrows), which are in one-to-one correspondence with dual edges (red arrows).

The mesh construction limits the space of admissible solutions by requiring that edges have the specified orientation. This was done to ease interpretation of the segmented region boundary, and adds the condition that the dual graph be bipartite. However, it is also possible to construct meshes with pairs of double edges (forward and reverse) that do not have this requirement [7]. In this case auxiliary or degenerate cycles must be inserted ”between” the edge pairs, and the corresponding auxiliary variables serve to reverse the orientation if the surrounding surface topology supports it. An arbitrary triangular or tetrahedral (in 3D) mesh can therefore be used instead. In this case, however, it is possible for both a forward and a reverse edge to simultaneously be in the solution, which requires some additional interpretation.

5. Implementation

The optimisation problem presented in Equation 7 can be formulated as a minimum s-t cut problem, and using maxflow/mincut duality can be solved using a maximum flow solver. Each element of $\mathbf{x}$ represents a cycle, and relates to a node in the graph construction. Since $x_i \in \{0,1\}$ and a cut must separate the node from either the source or the sink, the mapping from cuts to values of $\mathbf{x}$ is well defined: if the cut separates node i from the source then $x_i = 1$, while $x_i = 0$ if the cut separates node i from the sink. It remains to specify the arcs in the graph and their corresponding costs.

The $\mathbf{x}_1 \leq \mathbf{x} \leq \mathbf{1} - \mathbf{x}_0$ constraints on the elements of $\mathbf{x}$ are simple. Considering the first case, $\mathbf{x}_1 \leq \mathbf{x}$, this constraint is only limiting where elements of $\mathbf{x}_1$ are unity. The corresponding elements x_i must then be 1, so the related nodes must be in the sink set. This can be enforced using infinite t-weights from these nodes to the sink node. Similarly, the constraint $\mathbf{x} \leq \mathbf{1} - \mathbf{x}_0$ is only limiting where $\mathbf{x}_0$ is equal to 1, forcing the corresponding elements of $\mathbf{x}$ to 0, and can be enforced using arcs with infinite weight from the source node.

Since $\mathbf{C}$ is a cycle matrix, it has one $+1$ and one -1 element per row. The constraints on $\mathbf{y} = \mathbf{C}\mathbf{x}$ therefore involve pairs of nodes. Consider $\mathbf{C}\mathbf{x} \leq \mathbf{1} - \mathbf{y}_0$: since $\mathbf{x}$ is a zero-one vector this constraint is only limiting where the right hand side is zero, or where $\mathbf{y}_0$ is 1. The constraints then take the form $x_i - x_j \leq 0$, so the only disallowed configuration is when $x_i = 1$ and $x_j = 0$. This can be enforced by an arc with infinite capacity from node j to node i.

The constraint $\mathbf{y}_1 \leq \mathbf{C}\mathbf{x}$ is slightly more complicated. Each row can be written as either $1 \leq x_i - x_j$ or $0 \leq x_i - x_j$, depending on whether the corresponding element of $\mathbf{y}_0$ is 1 or 0. In the first case the only valid configuration is $x_i = 1$ and $x_j = 0$, which can be enforced with infinite weight arcs from the source node to x_j and from x_j to the sink. In the second case the only disallowed configuration is $x_i = 0$ and $x_j = 1$, which can be eliminated using an infinite weight arc from node i to j.

Disallowed configurations that violate the inequality constraints can therefore be eliminated in the minimum cut formulation using appropriate infinite arcs. The elements of the cut value $C(\mathbf{x}) = \mathbf{w}^T\mathbf{x} = \sum w_i x_i$ are represented by finite arcs between nodes and the source and sink terminals. Two cases occur. If w_i is positive, the weight of the cut should increase by w_i if $x_i = 1$. This is encoded by an arc from the source to node i with weight $w_i \geq 0$. If w_i is negative the total weight should decrease by $|w_i|$ if $x_i = 1$, which can be achieved by reparameterisation: w_i is added to the overall cost and an arc of weight $-w_i \geq 0$ is included between node i and the sink node [9].

A representation of the graph construction for the planar 2D case is shown in Figure 3. Red arrows in this graph

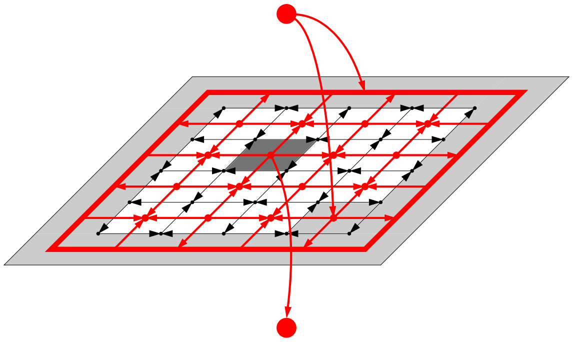
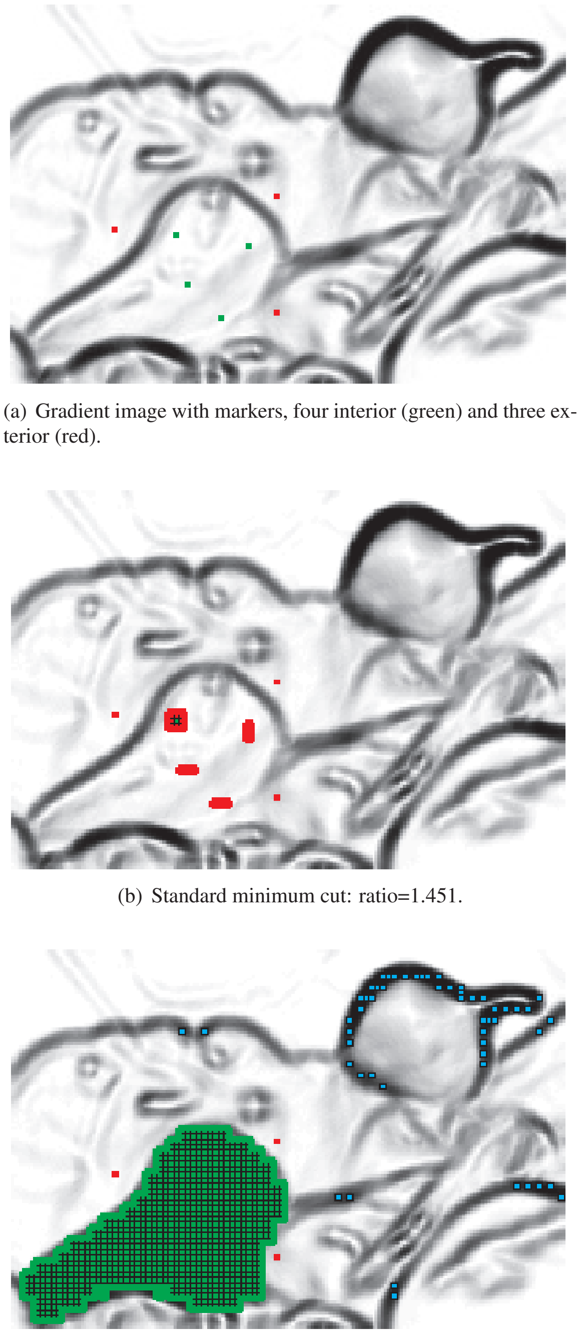


Figure 3. Graph construction for 2D segmentation. Dual arcs linking to displayed (dual) nodes all have infinite weight. The dark region is constrained to the interior via an infinite arc to the sink. Light regions (including the exterior face region) are constrained to the exterior via an infinite arc from the source. Other arcs between dual nodes (red) and terminal nodes are not shown.

all represent infinite weight arcs. The minimum cut can be found by maximising the flow in the constructed graph. We use the maxflow algorithm of [3], which is efficient for vision problems where interior arcs occur between spatially adjacent nodes.

Figure 4 shows an example of interactive segmentation using the ratio cut algorithm. The data is the gradient magnitude g of an image of vegetables, mapped through an exponential e^{-g/σ^2} (with a fairly arbitrary bandwidth parameter σ related to the size of image structures), and forms the background in the displayed images. The cost function only uses boundary weights (so $\mathbf{n}_x = \mathbf{d}_x = \mathbf{0}$ in Equation 6), although including region weights would obviously improve performance. The numerator in the ratio cost measures the total weight of the cycles in the solution, where weights are proportional to the grey level displayed, and interior and exterior constraint seeds are shown in Figure 4(a). The denominator measures the total Euclidean length of all the boundaries in the solution. The ratio is therefore a geodesic measurement of boundary cost per unit length, which has no shrinking bias. While numerator costs are all positive in this formulation it is not required. Figure 4(b) shows the result for a standard minimum cut: the interior and exterior constraints are topologically consistent with the solution, but the minimum cost objective favours short Euclidean paths over low average cost paths. Figure 4(c) shows the ratio cut solution, with the interior hatched: the desired solution is obtained with a small amount of annotation, and the algorithm terminated after 7 iterations.

The formulation does have some shortcomings. A minimum cost per unit length solution tends to make paths as long as possible in areas that have low cost. This results in spurious cycles visible in the top right of Figure 4(c) (at the position of what was an onion in the original image): still topologically consistent, they have ratio costs lower than that of the desired cycle shown in green. It has not been determined whether the cycles that enclose interior seeds still have some useful optimality properties. To extract the desired object from the segmentation we simply detect cycles not connected to seed regions, and remove them from the solution. For the results shown in Figure 4 the ratio cost of the desired solution is 0.707 before removing the disconnected cycles and 0.83 after, while the minimum cut solution has ratio cost of 1.451. Interestingly, Vicente et al. [17] consider connectivity constraints in a segmentation problem and indicate that they are NP-hard to optimise over even in the planar 2D case.



(a) Gradient image with markers, four interior (green) and three exterior (red).

(b) Standard minimum cut: ratio=1.451.

(c) Minimum ratio cut: ratio=0.707 (post-processed=0.83).

Figure 4. Example of interactive image segmentation using standard minimum cut and minimum ratio cut.

These spurious cycles do not occur without region constraints. Consider two cycles with costs c_1 and c_2 and lengths l_1 and l_2. If cycle 1 is in the solution, then cycle 2 will enter if $(c_1 + c_2)/(l_1 + l_2) \leq c_1/l_1$, or if $c_2/l_2 \leq c_1/l_1$ (so the ratio cost of cycle 2 is less than that of cycle 1). By identical reasoning cycle 1 will only be included in the solution with cycle 2 if $c_1/l_1 \leq c_2/l_2$. The only way that cycle 1 and cycle 2 can be in the solution simultaneously is if they each have a ratio cost the same as their combined ratio cost. However, if one of the cycles is constrained then this symmetry is broken, and cycles not related to the main cycle can appear.

The problem as described does not utilise the full power of a minimum cut algorithm: interior arcs (*i.e.* those not connected to the source or sink terminals) all have weight ∞, and are there purely to enforce consistency between regions and boundaries. However, the solution as formulated permits including finite weight arcs between nonadjacent cells, which could be used to penalise configurations based on other information. In this sense the infinite weight arcs represent a (submodular) structural constraint, and any other submodular MRF can be superimposed on the graph construction.

6. Application in 3D

Given a collection of candidate faces $f_i, i = 1, \dots, N$ and J camera views, each view j allocates a cost c_{ij} for including that face into the solution. We also assume that the visibility of faces is known for each view: $v_{ij} = 1$ if face i is visible in view j, zero otherwise, yielding a set of visibility values by $\mathbf{v}$. We use an estimate of $\mathbf{v}$ obtained from an initial visual hull reconstruction from the object silhouettes: any face in the set of candidate faces is assumed to have the same visibility as the nearest face on the visual hull.

A principled way of assigning costs c_{ij} is to consider how a face projects into the images. Since we know from which cameras a face is visible, we can backproject the pixels from the associated views and form an estimate of the face appearance based on all the image data. A robust measure can be used to mitigate visibility errors. For now we use a coarse measurement model, where only the projection of the face centroid in the images is used for calculating appearance. Thus the model assumes that the faces are uniformly coloured, the surface Lambertian, and the lighting homogeneous. The squared difference between this average reconstructed colour over the equivalent region in the image is used as a match measure.

An indicator variable x_i is assigned to each patch in the model. The total error of a configuration of $\mathbf{x}$ is the sum of the squared errors for all the included visible faces over all the views:

$$E_{\mathbf{v}}(\mathbf{x}) = \sum_j \sum_i x_i v_{ij} c_{ij}. \tag{9}$$

Minimising $E_{\mathbf{v}}(\mathbf{x})$ over $\mathbf{x}$ will not work for the same reason that most geodesic formulations fail: the total squared error will tend to be minimised for small reconstructed objects, which project to a lower area in the images and therefore contribute less total cost.

The ratio cost can be used to eliminate this problem. We create a projected area functional

$$A_{\mathbf{v}}(\mathbf{x}) = \sum_j \sum_i x_i v_{ij} a_{ij}, \tag{10}$$

where a_{ij} measures the area in image j covered by face f_i. The ratio functional

$$R_{\mathbf{v}}(\mathbf{x}) = \frac{E_{\mathbf{v}}(\mathbf{x})}{A_{\mathbf{v}}(\mathbf{x})} \tag{11}$$

therefore measures the total squared error per unit of projected area, which is equivalent to mean squared reprojection error. Note that this formulation also assumes $\mathbf{n}_x = \mathbf{d}_x = \mathbf{0}$ in Equation 6, although ballooning could be incorporated via an appropriate choice of nonzero $\mathbf{n}_x$.

Figure 5 shows results when optimising this functional for a calibrated turntable dataset of 8 images with the camera slightly overhead[1]. The reference visibility estimate is obtained by silhouette backprojection of careful segmentations. Interior and exterior constraint regions are specified according to the distance of voxel centers from the visual hull surface: active voxels are chosen to lie within a distance of 50 in the interior and 20 in the exterior, where the height of the figurine in world units is approximately 400, and at the resolution used yields about 1 million faces.

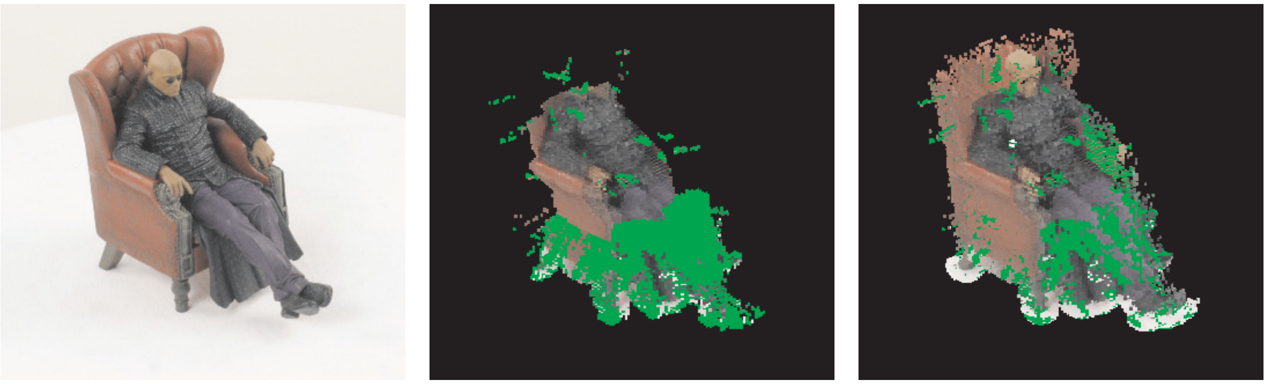
Figure 5(a) shows one of the views used in the reconstruction. Figure 5(b) shows a reprojection of the standard minimal surface solution onto the same viewpoint, using the estimated face colours for the rendering. Green faces are ones where the visual hull visibility estimate does not provide any cameras that are visible in the forward direction. These have no projected costs (since they have no associated error), and are therefore inclined to appear in the solution. No ballooning term is used, so the minimum cut suffers from bias in shrinking the solution towards the interior constraint surface. Figure 5(c) shows the ratio cut solution: the bias is absent and the reconstruction is good, apart from the errors caused by poor visibility estimates on some faces. The optimisation takes about one second after all the required quantities are calculated, and 6 iterations were needed to arrive at the minimum ratio solution.

The principle applied here can also be used for other measures of match, which can instead be measured in units of projected area rather than in absolute units. For example, if image data is used to calculate a consistency measure for a face, then this consistency can be scaled by the projected area and will therefore be properly normalised. Oriented visibility or photometric models can be included as a matter of course. An advantage of this explicit surface formulation is that it allows errors to be referred to the image domain, where maps of the reprojection error can be made and model validity assessed.

7. Discussion and conclusion

A method has been presented for segmenting 2D images based on a very general minimum ratio cost function. Constraints on the solution can be easily and conveniently imposed, and an efficient algorithm exists for performing the

[1] http://www-cvr.ai.uiuc.edu/ponce_grp/data/mview.



(a) One of 8 images used in reconstruction.

(b) Minimum surface (no ballooning): ratio=647.4.

(c) Ratio surface: ratio=258.6.

Figure 5. Three-dimensional reconstruction using minimum cut and ratio cut.

optimisation. It works because negative costs on boundary elements can be referred to regions via the region-boundary operator, and region costs can take arbitrary sign without affecting the tractability of the problem.

The method uses the principles of differential geometry, and the formulation extends naturally to three (or more) dimensions. It is shown that 3D reconstruction without a shrinking bias is possible, and the solution can be obtained efficiently.

The formulation does have some complications. When region constrains are imposed, the solution can exhibit spurious foreground regions which make it difficult to interpret. These are a consequence of the cost function rather than of the algorithm, and it remains to be seen whether there is a systematic way of eliminating them.

References

[1] G. Borradaile. *Exploiting Planarity for Network Flow and Connectivity Problems*. PhD thesis, Brown University, 2007. 2

[2] Y. Boykov and M.-P. Jolly. Interactive graph cuts for optimal boundary and region segmentation of objects in N-D images. In *ICCV*, volume I, pages 105–112, 2001. 1

[3] Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *PAMI*, 26(9):1124–1137, September 2004. 6

[4] I. J. Cox, S. B. Rao, and Y. Zhong. ”Ratio regions”: A technique for image segmentation. In *ICPR*, volume 2, pages 557–564, Vienna, Austria, 1996. 1

[5] J. B. G. Frenk and S. Schaible. Fractional programming. In N. Hadjisavvas, S. Komlósi, and S. Schaible, editors, *Handbook of Generalized Convexity and Generalized Monotonicity*, chapter 8. Springer, 2005. 3

[6] L. Grady. Computing exact discrete minimal surfaces: Extending and solving the shortest path problem in 3D with application to segmentation. In *CVPR*, pages 69–78, New York, NY, 2006. 1, 3, 4

[7] L. Grady. Minimal surfaces extend shortest path segmentation methods to 3D. *PAMI*, 32(2):321–334, February 2010. 1, 3, 4, 5

[8] I. H. Jermyn and H. Ishikawa. Globally optimal regions and boundaries as minimum ratio weight cycles. *PAMI*, 23(10):1075–1088, October 2001. 1

[9] P. Kohli and P. H. S. Torr. Dynamic graph cuts for efficient inference in markov random fields. *PAMI*, 29(12):2079–2088, 2007. 1, 5

[10] V. Kolmogorov, Y. Boykov, and C. Rother. Applications of parametric maxflow in computer vision. In *ICCV*, pages 1–8, Rio de Janeiro, 2007. 1

[11] E. L. Lawler. *Combinatorial Optimization: Networks and Matroids*. Hold, Rinehart and Winston, 1976. 3, 4

[12] C. Mattiussi. The Finite Volume, Finite Difference, and Finite Elements Methods as Numerical Methods for Physical Field Problems. *Advances in Imaging and Electron Physics*, 113:1–146, 2000. 1, 2

[13] N. Meggiddo. Combinatorial optimization with rational objective functions. *Mathematics of Operations Research*, 4(4):414–424, November 1979. 3

[14] J. R. Munkres. *Elements of algebraic topology*. Addison-Wesley, 1984. 2

[15] J. M. Sullivan. *A Crystalline Approximation Theorem for Hypersurfaces*. PhD thesis, Princeton University, 1990. 1, 4

[16] M. N. S. Swamy and K. Thulasiraman. *Graphs, Networks, and Algorithms*. Wiley, 1981. 3

[17] S. Vicente, V. Kolmogorov, and C. Rother. Graph cut based image segmentation with connectivity priors. In *CVPR*, pages 1–8, Anchorage, Alaska, 2008. 6

[18] G. Vogiatzis, P. H. S. Torr, and R. Cipolla. Multi-view stereo via volumetric graph-cuts. In *CVPR*, pages 391–398, San Diego, CA, 2005. 1

[19] S. Wang and J. M. Siskind. Image segmentation with ratio cut. *PAMI*, 25(6):675–690, June 2003. 1