Computing Geodesics and Minimal Surfaces via Graph Cuts

Yuri Boykov
Imaging & Visualization
Siemens Corp. Research
Princeton, NJ 08540

Vladimir Kolmogorov
Computer Science
Cornell University
Ithaca, NY 14853

Abstract

Geodesic active contours and graph cuts are two standard image segmentation techniques. We introduce a new segmentation method combining some of their benefits. Our main intuition is that any cut on a graph embedded in some continuous space can be interpreted as a contour (in 2D) or a surface (in 3D). We show how to build a grid graph and set its edge weights so that the cost of cuts is arbitrarily close to the length (area) of the corresponding contours (surfaces) for any anisotropic Riemannian metric.

There are two interesting consequences of this technical result. First, graph cut algorithms can be used to find globally minimum geodesic contours (minimal surfaces in 3D) under arbitrary Riemannian metric for a given set of boundary conditions. Second, we show how to minimize metrication artifacts in existing graph-cut based methods in vision. Theoretically speaking, our work provides an interesting link between several branches of mathematics - differential geometry, integral geometry, and combinatorial optimization. The main technical problem is solved using Cauchy-Crofton formula from integral geometry.

1. Introduction

Our work unifies two standard image segmentation techniques: geodesic active contours and graph cuts. Each of these approaches has its own benefits and drawbacks. Geodesic active contours [6, 29] are based on a continuous formulation (computing geodesics in Riemannian spaces), and produce minimal geometric artifacts. Standard variational techniques for computing geodesic contours (e.g. the level set method) generate local minima of the energy which may be sensitive to initialization. Highly desirable anisotropic formulations tend to be slower due to increased computational burden.

One attractive feature of the graph cut approach is that it can find a global minimum of the energy (e.g. [13, 28, 15, 2]). On the other hand, discrete topology of graphs may produce noticeable geometric artifacts known as metrication errors. For example, 2D grid graphs with a simple 4-neighborhood system impose "Manhattan distance" (L1) metric on the underlying image space. This may create visual artifacts as L1 is not invariant to image rotations and does not treat different directions in the image equally.

In this paper we introduce a notion of *cut metric* on graphs. In fact, cut metrics are (informally speaking) "dual" to well known path-based metrics on graphs. We study geometric properties of cut metrics in case of regular grids. Using powerful Crofton-style formulas from integral geometry we solve the following open problem: how to construct a graph where cut metric approximates any given Riemannian metric with arbitrary precision. Previously, it was not even clear if such a construction was possible.

Our results allow to combine ideas from differential geometry and combinatorial optimization. In particular, we propose a *geocuts* algorithm for image segmentation. Similar to the geodesic active contours technique, we formulate the problem as finding geodesics (in 2D) or minimal surfaces (in 3D). Unlike the level-set method, we use graph cuts to compute *global* geodesics for a given set of boundary conditions. Potentially, this could reduce sensitivity to initialization. Anisotropic metrics present no additional computational cost for our algorithm. Similar to level-set methods, geocuts method is "topologically" free.

The structure of the paper is as follows. Related material from differential geometry, integral geometry, and combinatorial optimization is reviewed in Section 2. The concept of cut metrics is discussed in Section 2.4. In Sections 3 and 4 we show how to build graphs whose cut metric approximate any given continuous Riemannian metric. Geocut algorithm and experimental results are presented in Section 5.

2. Related work and background

2.1. Differential geometry and active contours

Active contours is an interesting application of Differential Geometry [5] in computer vision. Since the intro-

duction of "snakes" [16], active contour models have been widely used for image segmentation. Original snakes represented contour models as a parametric mapping $C(v) = (x(v), y(v))$ for $v \in [0, 1]$. The energy of the model is

$$E(C) = \int_0^1 \alpha |C_v(v)|^2 + \beta |C_{vv}(v)|^2 - |\nabla I(C(v))| dv$$

where C_v and C_{vv} are the first and second derivatives of C with respect to contour parameter v, and $I : R^2 \rightarrow R^+$ is a given image in which we want to detect the object boundaries. Such energies can be minimized via gradient descent leading to a sequence of moving ("active") contours. Details for parametric active contours can be found in [14].

A noticeable development was the introduction of an implicit representation for active contours as level-sets of an auxiliary function [25, 21]. Unlike most of the snake based methods, this allows topological changes of the curve.

Another important step was the "geodesic active contour" model [6, 29]. The two terms in the energy corresponding to internal and external forces were combined into a single term. Their curve evolution is a result of minimizing the functional

$$E(C) \ = \ \int_0^{|C|_\mathcal{E}} g(|\nabla I(C(s))|) \ ds$$

where parameter s is specifically chosen as the (Euclidean) arc length on the contour, $|C|_\mathcal{E}$ is the Euclidean length of contour, and g is a strictly decreasing function converging to zero at infinity. It was shown that in many cases this method behaves better than its ancestors.

The formulation of [6, 29] can be viewed as a problem of finding *local geodesics* in a space with Riemannian metric computed from the image. Note that the (non-Euclidean) length of a contour in a given Riemannian space is

$$|C|_\mathcal{R} \ = \ \int_0^{|C|_\mathcal{E}} \sqrt{\tau_s^T \cdot D(C(s)) \cdot \tau_s} \ ds$$

where a positive definite matrix $D(\cdot)$ specifies the local Riemannian metric at a given point/pixel in the image and τ_s is a unit tangent vector to the contour. In fact, the contour energy $E(C)$ above is equal to $|C|_\mathcal{R}$ in case of an isotropic Riemannian metric

$$D(\cdot) \ = \ \mathbf{diag}(g(|\nabla I(\cdot)|)).$$

Like in most of the previous approaches, the algorithm in [6, 29] searches for some local minimum which is close to the initial guess. Numerical optimization is performed via level-sets. The same approach can be used for 3D segmentation via *minimal surfaces* (see [7] for details). Further generalizations of geodesic active contours and some anisotropic metrics are discussed in [17]. Regional properties of geodesic active contours are considered in [22].

Cohen et. al. [8] developed an algorithm for computing *minimal geodesics*, i.e. the global minimum of the same energy. Their approach is based on minimal paths and shares some similarities with the Dijkstra shortest-path algorithm. Connections between level-set methods and Dijkstra's algorithm are well known (e.g. see [25]).

2.2. Integral geometry and Crofton formulas

The name of Integral Geometry was introduced by Blaschke in [1]. The basic ideas have their origin in the theory of Geometric Probabilities. In fact, by using concepts of probability M. W. Crofton was the first to obtain some remarkable integral formulas of a purely geometrical character. These formulas can be considered as one of the starting points of Integral Geometry.

Below we review one classical Cauchy-Crofton formula that is crucial for the theory of graph cut geometry developed in this paper. This formula relates a length of a curve in R^2 to a measure of a set of lines intersecting it. We will introduce basic terminology and discuss some facts that are important for the consequent development of the material in this paper. More details about Crofton formula and Integral Geometry in general can be found in [24, 5].

Consider a straight line $L(\rho, \phi)$ in the plane R^2 determined by its *normal* parameters ρ and ϕ as shown on Figure 1. First, we will describe a reasonable way of assigning a measure to a given subset of straight lines. Consider a set $\mathcal{L} = \{(\rho, \phi) : \rho \geq 0, \ \phi \in [0, 2\pi]\}$ describing all straight lines L (see Figure 2) and Lebesgue measure on this set defined by its density $d\mathcal{L} \propto d\rho \cdot d\phi$. Lebesgue measure of a subset of straight lines $\mathcal{X} \subset \mathcal{L}$ is given by the integral $\int_\mathcal{X} d\mathcal{L}$. Note that any rigid motion $\mathcal{M}$ on the plane R^2 transforms a subset of lines $\mathcal{X}$ into another subset $\mathcal{X}^\mathcal{M}$. In fact [24, 5], Lebesgue measure is the only measure on $\mathcal{L}$ that is invariant under rigid motions so that $\int_\mathcal{X} d\mathcal{L} = \int_{\mathcal{X}^\mathcal{M}} d\mathcal{L}$.

The following Cauchy-Crofton formula establishes a connection between Euclidean length $|C|_\mathcal{E}$ of a curve C in R^2 and a measure of a set of lines intersecting it.

$$\int n_c \ d\mathcal{L} \ = \ 2 \ |C|_\mathcal{E} \qquad (1)$$

Function $n_c(L)$ specifies a number of times any given line L intersects C (see Figure 2). In fact, Cauchy-Crofton formula (1) holds for any rectifiable curve [1][1]. If contour C is convex then (1) reduces to $\int_{\mathcal{L}_C} d\mathcal{L} = |C|_\mathcal{E}$ where $\mathcal{L}_C$ is a subset of lines intersecting C. That is, length of a convex contour equals Lebesgue measure of the set of lines intersecting it. This is one of the most simple and elegant examples of a Crofton-style formula in Integral Geometry.

[1]Moreover, (1) can be used to generalize the concept of length to a continuum of points [11]. It is important that the integral in (1) be in the Lebesgue sense rather than in the Riemann sense.

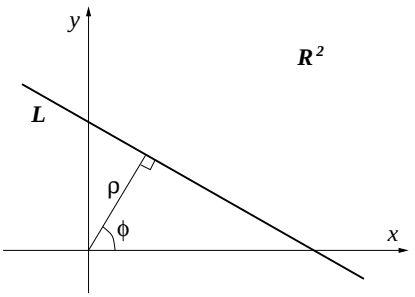
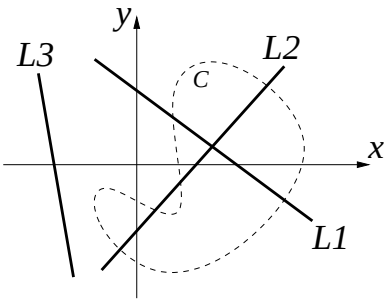
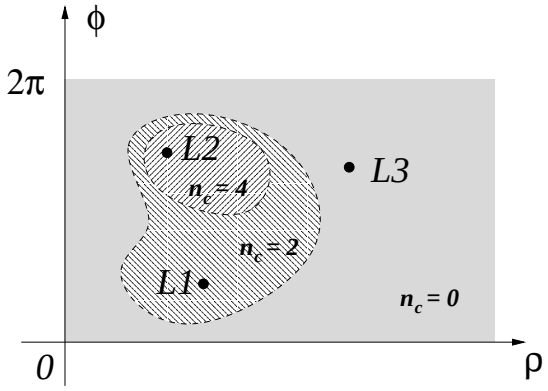


Figure 1. Normal parameters of a straight line L in R^2. The parameters $\rho \geq 0$ and $0 \leq \phi \leq 2\pi$ are the polar coordinates of the foot of the perpendicular from the origin onto the line. Points (x, y) on L satisfy $x \cos\phi + y \sin\phi = \rho$.

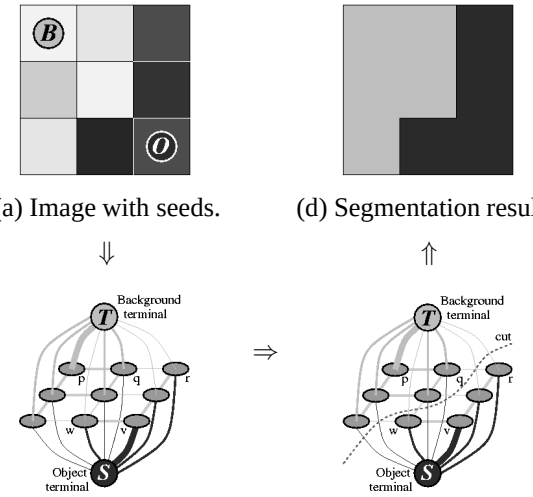


(a) Lines in R^2.



(b) Lines as points in $\mathcal{L}$.

Figure 2. Any line L on R^2 in (a) has a unique pair of normal parameters (ρ, ϕ). That is, lines L can be represented as points of the set $\mathcal{L} = \{(\rho, \phi) : \rho \geq 0,\ \phi \in [0, 2\pi]\}$ shown in (b). Note that any given contour C in (a) defines a function n_c on $\mathcal{L}$ that specifies a number of intersections with C. Different shades in (b) represent subsets of lines L where $n_c(L) = 0$, $n_c(L) = 2$, and $n_c(L) = 4$.



(a) Image with seeds. (d) Segmentation results.

(b) Graph. (c) Cut.

Figure 3. A simple 2D segmentation example for a 3×3 image. Boundary conditions are given by object seeds $\mathcal{O} = \{v\}$ and background seeds $\mathcal{B} = \{p\}$ provided by the user. The cost of each edge is reflected by the edge's thickness. Minimum cost cut is attracted to cheap edges.

2.3. Graph cut methods in vision

Graph cuts have been used for many early vision problems like stereo [23, 4, 18], segmentation [28, 26, 27, 2], image restoration [13, 4], texture synthesis [19], and many others. Below we briefly overview garph-based segmentation method in [2], which works as a foundation for our geocuts technique in Section 5.1. Also, we introduce some necessary terminology from combinatorial optimization.

An undirected graph $\mathcal{G} = \langle \mathcal{V}, \mathcal{E} \rangle$ is defined as a set of nodes (vertices $\mathcal{V}$) and a set of undirected edges ($\mathcal{E}$) that connect these nodes. An example of a graph is shown in Figure 3(b). Each edge $e \in \mathcal{E}$ in the graph is assigned a nonnegative weight (cost) w_e. There are also two special nodes called terminals. A cut is a subset of edges $C \subset \mathcal{E}$ such that the terminals become separated on the induced graph $\mathcal{G}(C) = \langle \mathcal{V}, \mathcal{E} \backslash C \rangle$. Each cut has a cost which is defined as the sum of the costs of the edges that it severs

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

A globally minimum cut on a graph with two terminals can be computed efficiently in low-order polynomial time via standard max-flow or push-relable algorithms from combinatorial optimization (e.g. [9]).

Graph cut formalism is well suited for segmentation of images. In fact, it is completely appropriate for N-dimensional volumes. The nodes of the graph can represent pixels (or voxels) and the edges can represent any neighborhood relationship between the pixels. A cut partitions the nodes in the graph. As illustrated in Figure 3 (c-d), this partitioning corresponds to a segmentation of an underlying image or volume. A minimum cost cut generates a segmentation that is optimal in terms of properties that are built into the edge weights.

2.4. "Cut metrics" vs. "path metrics"

Below we introduce a new concept of a *cut metric* on graphs. For better motivation, we will first discuss a related notion of a *path metric* which is more standard for graphs.

Consider a weighted graph $\mathcal{G} = \langle V, E \rangle$. "Length" can be naturally defined for any path $p_{AB} \subset E$ connecting two nodes $A, B \in V$ as the sum of edge weights along the path

$$|p_{AB}| = \sum_{e \in p_{AB}} w_e$$

The distance, or the shortest path, between any two nodes can be computed via Dijkstra algorithm (e.g. see [9]). Such distances correspond to a *path metric*[2] on the graph.

Path metrics are relevant in many computer vision applications (e.g. [12, 8]) based on Dijkstra-style optimization. A choice of the neighborhood system (graph topology) and edge weights determine a graph's path metric. This may significantly affect the quality of results. In fact, the size of the neighborhood system is important. For example, consider path metric distance maps[3] for simple 2D grid-graphs with 4, 8, and 128 neighborhood systems in Figure 4. The quality of segmentation results of Dijkstra based methods can suffer from "blockiness" (like in Figures 8(b)(e)) in case of "Manhattan" style metric in Figure 4(a). The path metrics in (b) and (c) are much closer to the Euclidean metric. In general, the segmentation results will be smoother if Dijkstra based method use larger neighborhood system.

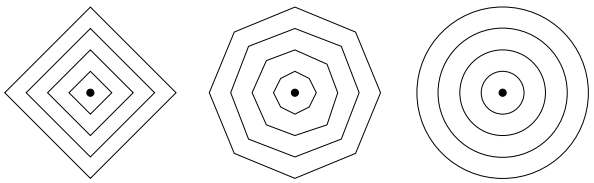
In this paper we introduce *cut metrics* on graphs which, in some sense, are complimentary or "dual" to *path metrics*. The major advantage of cut-based methods (see Section 2.3) over Dijkstra based segmentation techniques is that they are not limited to contours (1D paths) and can find globally optimal (minimal) hyper-surfaces in N-D cases. This significantly broadens the scope of useful applications.

The main intuition comes from an observation that a cut on a grid-graph $\mathcal{G} = \langle V, E \rangle$ embedded in R^n can be seen as a closed contours (in R^2) or as a closed surfaces (in R^n). "Length", or "area" in N-D, can be naturally defined for any cut $C \subset V$ as

$$|C|_{\mathcal{G}} = \sum_{e \in C} w_e \qquad (2)$$

which is simply the standard definition of cut cost from combinatorial optimization. Due to geometric interpretation of $|C|_{\mathcal{G}}$ as the "length" or "area" of the corresponding contour or surface C, we can talk about metric properties of cuts on graphs. We will use the term *cut metric*[4] in the context of geometric properties of graph cuts (as hypersurfaces) implied by the definition 2.

Similarly to path metric, all properties of cut metric on a graph are determined by the graph's neighborhood system and by edge weights. In fact, larger neighborhood systems allow both cut and path metrics to approximate continuous metrics. In the example of Figure 4 path metric approximates continuous Euclidean distances when weights of edges are equal to their Euclidean length. In fact, cut metric on a 2D grid-graph can obtain the same distance maps as in Figure 4 but the corresponding edge weights are different. Equation (4) in Section 3 shows that weights of edges should be inversely proportional to their Euclidean length.



(a) 4 n-system (b) 8 n-system (c) 128 n-system

Figure 4. Distance maps for path metrics on grid-graphs with different size neighborhood systems. In each case, weights of edges are equal to their Euclidean length. The contours represent nodes equidistant from a given center.

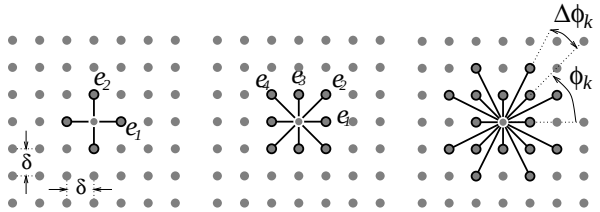
3. Euclidean Cut Metric on 2D grids

In this section we show how to build a 2D grid graph whose cut metric approximates Euclidean metric. In practice we are much more interested in approximating Riemannian metrics. In this section we use Euclidean metric as a simple example to introduce all the key ideas. In Section 4 we generalize them to Riemannian case.

[2] Despite popularity of the Dijkstra algorithm in computer science, the actual term "path metric" is not very common. However, it is used explicitly in the theory of Finite Metric Spaces [10, 20] that, in particular, studies embeddability of graphs in normed spaces.

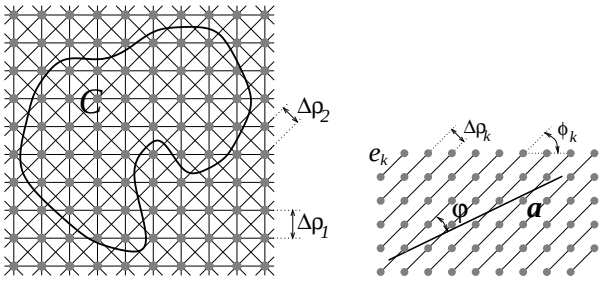
[3] Personal communications with Marie-Pierre Jolly.

[4] There should be no confusion with a term *cut semi-metric* used in [10, 20] for a very specific inter-node distances assigned depending on one specific fixed cut. We use the word "cut" generically. Our *cut metric* on a graph does not depend on one fixed cut.



(a) 4 n-system (b) 8 n-system (c) 16 n-system

Figure 5. Examples of neighborhoods in 2D.



(a) 8-neighborhood 2D grid (b) One family of lines

Figure 6. A regular grid.

3.1. Regular 2D Grids

In this section we discuss the structure and basic terminology for 2D grid graphs. We assume that all nodes are embedded in R^2 in a regular grid-like fashion with cells of size δ. We also assume that all nodes have topologically identical neighborhood systems. Some examples of possible neighborhood systems are shown in Figure 5. The example in Figure 6 (a) shows a regular grid when all nodes have identical 8-neighborhood systems as in Figure 5 (b).

Neighborhood systems of a regular grid $\mathcal{G}$ can be described by a set $\mathcal{N}_\mathcal{G} = \{e_k : 1 \le k \le n_\mathcal{G}\}$ of distinct (undirected[5]) vectors. For example, grids with an 8 neighborhood system is described by a set of four vectors $N_\mathcal{G} = \{e_1, e_2, e_3, e_4\}$ shown in Figure 5 (b). We will assume that vectors e_k are enumerated in the increasing order of their angular orientation ϕ_k so that $0 = \phi_1 < \phi_2 < ... < \phi_{n_\mathcal{G}} < \pi$. For convenience, we assume that e_k is the shortest length vector connecting two grid nodes in the given direction ϕ_k.

As shown in Figure 6 (b), each vector $e \in \mathcal{N}_\mathcal{G}$ generates a family of *edge-lines* on the corresponding grid. It is easy to check that the distance between the nearest lines in a family generated by e_k is

$$\Delta\rho_k = \frac{\delta^2}{|e_k|}$$

[5]We do not differentiate between e and $-e$

where δ is the cell-size of the grid and $|e_k|$ is the (Euclidean) length of vector e_k. Each family of edge lines is characterized by the inter-line distance $\Delta\rho_k$ and by its angular orientation ϕ_k. We will also use angular differences between the nearest families of edge lines $\Delta\phi_1 = \phi_2 - \phi_1,\ \Delta\phi_2 = \phi_3 - \phi_2, \ldots, \Delta\phi_{n_\mathcal{G}} = \pi - \phi_{n_\mathcal{G}-1}$.

So far we discussed only topological structure of the grid. Another important aspect of any graph are edge weights. We will use the following notation. If we set equal weights for all edges in the same family of lines, that is for all edges with orientation ϕ_k, then we use w_k to denote these common weight. For example, this will be the case when we want to approximate Euclidean or any other spatially homogeneous metric $D(\cdot) = const$. In a more general case we will use $w_\phi(p)$ for a weight of a (directed) edge that originates at node p and has orientation ϕ.

EXAMPLE 1 *As a simple illustration we would like to show that cut metric on a regular 2D grid implicitly assigns certain "length" to curves. For simplicity, consider a segment of a straight line a shown in Figure 6 (b). This segment can be considered as a part of some cut that severs edges on the grid. We can compute the cost of severed edges as follows.*

For a k^{th} family of edge lines on the grid we can easily count the number of intersections with a as

$$|a|_k = \frac{|a| \cdot |\sin\varphi|}{\Delta\rho_k} = \frac{|a \times e_k|}{\delta^2}$$

Summing over all families of edge-lines we get from (2)

$$|a|_\mathcal{G} = \frac{1}{\delta^2} \cdot \sum_{k=1}^{n_\mathcal{G}} w_k \cdot |a \times e_k| \qquad (3)$$

assuming constant edge weights w_k within the same family. This equation holds for a vector a with an arbitrary orientation. Thus, we can use (3) to visualize 2D distance maps. In particular, this equation gives distance maps identical to those in Figure 4 if edge weights w_k are appropriately chosen to approximate Euclidean metric (see formula 4).

3.2. Graph cuts and Cauchy-Crofton formula

In this section we will use integral geometry to establish a necessary technical link between the concepts of (discrete) cut metric on a grid (in combinatorial optimization) and (continuous) Euclidean metric on R^2 (in differential geometry). Consider a contour C in the same 2D space where grid graph $\mathcal{G}$ is embedded (as in Figure 6(a)). Contour C gives a binary partitioning of graph modes and therefore corresponds to a cut on $\mathcal{G}$. Then we can consider the length $|C|_\mathcal{G}$ of the contour imposed by the graph's cut metric (2). Below we derive edge weights on $\mathcal{G}$ so that the cut based length $|C|_\mathcal{G}$ is close to the Euclidean length $|C|_\mathcal{E}$. We will

also discuss our convergence result as the topology $\mathcal{N}_\mathcal{G}$ and the grid size δ get finer and finer.

First, we will assume that graph's topology $\mathcal{N}_\mathcal{G}$ is fixed. According to Cauchy-Crofton formula (1) we can represent the Euclidean length of C as an integral over the set of all straight lines $L(\rho, \phi)$ in $\mathcal{L}$

$$|C|_\mathcal{E} \;=\; \frac{1}{2}\int\; n_c \; d\mathcal{L} \;=\; \frac{1}{2}\int_0^\pi \int_{-\infty}^{+\infty} \; n_c(\rho,\phi) \;\; d\rho \; d\phi$$

where radius ρ is allowed to be negative while angle ϕ is restricted to the interval $[0, \pi]$. Remember, function n_c specifies how many times line $L(\rho, \phi)$ intersects contour C. By choosing an appropriate partitioning of the set $[0, \pi] \times R$ we can approximate the integral above by its partial sum

$$\begin{aligned} |C|_\mathcal{E} \;\; &\approx \;\; \frac{1}{2}\sum_{k=1}^{n_\mathcal{G}} \left(\sum_i n_c(i,k)\Delta\rho_k \right) \Delta\phi_k \;\; = \\ &= \;\; \sum_{k=1}^{n_\mathcal{G}} \;\; n_c(k) \;\cdot\; \frac{\delta^2 \cdot \Delta\phi_k}{2 \cdot |e_k|} \end{aligned}$$

where index i enumerates lines in the k^{th} family of edge-lines, $n_c(i,k)$ counts intersections of line i in the k^{th} family with contour C, and $n_c(k) = \sum_i n_c(i,k)$ is the total number of intersections of C with the k^{th} family of edge-lines.

If we choose constant edge weights within each family of edge lines as

$$w_k = \frac{\delta^2 \cdot \Delta\phi_k}{2 \cdot |e_k|} \tag{4}$$

then we have

$$|C|_\mathcal{E} \;\; \approx \;\; \sum_{k=1}^{n_\mathcal{G}} \; n_c(k) \cdot w_k \;\; \approx \;\; |C|_\mathcal{G}$$

The approximation error on the left hand side is due to the difference between an integral and its partial sum. As we make partitioning finer this error of approximation converges to zero. In our case partitioning size is determined by the grid size δ and by $\sup_k |\Delta\phi_k|$.

The approximation on the right hand side is due to the fact that the number of intersections between the contour and an edge-line might be different from the actual number of edges along this line that are severed by the cut corresponding to the contour. Examples of this phenomenon can be found in Figure 6(a) in places where some contour wiggles are comparable with the size of edges $|e_k|$. This source of error converges to zero as $|e_k|$ gets smaller. We can prove the following convergence theorem (the proof is omitted).

Theorem 1 *(Pointwise Convergence) If C is a continuously differentiable regular curve in R^2 intersecting each straight line a finite number of times then*

$$|C|_\mathcal{G} \to |C|_\mathcal{E}$$

as δ, $\sup_k |\Delta\phi_k|$, and $\sup_k |e_k|$ get to zero.

4. Riemannian Metrics in 2D and in 3D

The results in the previous section generalize to non-Euclidean metric spaces. In this section we show how to set edge weights on a grid so that its cut metric approximates a given continuous Riemannian metric. We follow the same basic steps as in Section 3.2 for a simpler case of Euclidean metric. Due to space limitations we skip the proofs.

First we consider a 2D Riemannian space with a constant metric $D(\cdot) = const$. We use the following Crofton-style formula for Riemannian length $|C|_\mathcal{R}$ of contour C

$$\int \frac{\det D}{2(u_L^T \cdot D \cdot u_L)^{3/2}} \; n_c \; d\mathcal{L} \;\; = \;\; 2\;|C|_\mathcal{R}$$

where u_L is the unit vector in the direction of the line L. This formula holds for any continuously differentiable regular curve C in R^2 and can be derived as an exercise in Integral Geometry [24].

Following the same approach as in the previous section, we obtain an expression for edge weights for a 2D grid

$$w_k = \frac{\delta^2 \cdot |e_k|^2 \cdot \Delta\phi_k \cdot \det D}{2 \cdot (e_k^T \cdot D \cdot e_k)^{3/2}} \tag{5}$$

in case of a constant Riemannian metric $D(\cdot) = const$. Note that equation (5) reduces to (4) if we plug in the identity matrix $D = \mathbf{I}$ corresponding to Euclidean metric.

Now we consider a more general 2D Riemannian space where metric (tensor) $D(p)$ continuously varies over points p in space. In this case we need to construct a directed graph. The expression for weights is very similar to (5)

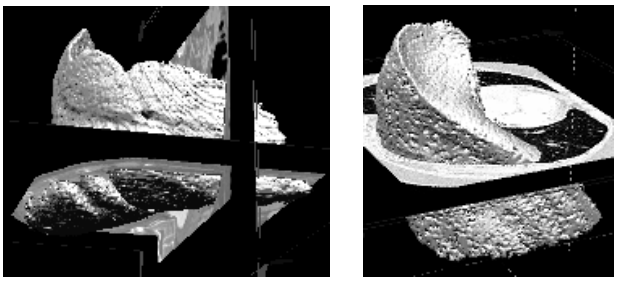
$$w_k(p) = \frac{\delta^2 \cdot |e_k|^2 \cdot \Delta\phi_k \cdot \det D(p)}{2 \cdot (e_k^T \cdot D(p) \cdot e_k)^{3/2}} \tag{6}$$

The difference is that edge weights now depend not only on edge's orientation k but also on node/pixel p where this (directed) edge originates. Note also that two opposite edges originating in the same node/pixel p and having orientations ϕ_k and $\pi + \phi_k$ are assigned the same weight $w_k(p)$.

Finally, we state our results for a regular 3D grid graphs embedded in a 3D Riemannian space with metric $D(p)$. Each edge angular orientation is now determined by spherical angles $\Phi_k = \{\psi_k, \varphi_k\}$. Using Cauchy-Crofton formula for the area of surfaces in 3D Riemannian spaces we obtain the following edge weights:

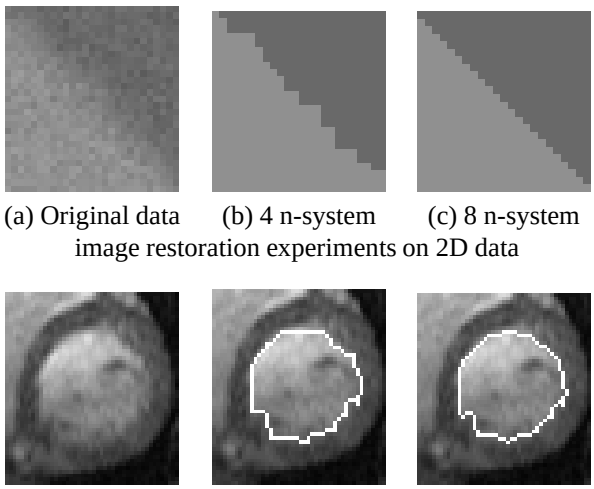
$$w_k(p) = \frac{\delta^3 \cdot |e_k|^3 \cdot \Delta\Phi_k \cdot \det D(p)}{\pi \cdot (e_k^T \cdot D(p) \cdot e_k)^2} \tag{7}$$

where $\Delta\Phi_k \;=\; \Delta\psi_k \cdot \Delta\varphi_k$ correspond to a given partitioning of the unit sphere among angular orientations $\Phi_1,\; \Phi_2, \ldots, \Phi_{n_\mathcal{G}}$ of edges in the neighborhood system.



(a) Liver, 144x170x170 (b) Lung, 253x165x205

Figure 7. Globally minimal surfaces for image-based Riemannian metrics (geocuts).



(a) Original data (b) 4 n-system (c) 8 n-system
image restoration experiments on 2D data

(d) Original data (e) 6 n-system (f) 26 n-system
object extraction experiments on 3D data

Figure 8. Reducing metrication artifacts.

5. Experimental Results

Cauchy-Crofton formulas from integral geometry were the key instruments for obtaining our technical results in Sections 3.2 and 4. These results establish an interesting link between two branches of mathematics: combinatorial optimization and differential geometry. In this section we show that both disciplines can benefit from this link. In fact, graph cuts can be used in differential geometry as a numerical method to compute globally optimal minimal surfaces in N-D Riemannian spaces for a given set of boundary conditions (Section 5.1). On the other hand, better approximation of continuous metrics can help many existing graph-cut based techniques in vision to reduce their metrication artifacts (Section 5.2).

5.1. "Geocuts" algorithm

Geocuts algorithm is a combination of the theoretical results in Section 4 with the graph-cut segmentation method in [2]. Equation 7 tells how to set edge weights so that the cost of graph cuts approximates Riemannian length/area of the corresponding contours/surfaces. [2] explains how to impose hard constraints (boundary conditions) on a graph. The globally optimal cut on the corresponding grid can be computed in close to linear time via a number of basic s/t graph cut methods from combinatorial optimization.

In Figure 7 we show a couple of examples of globally optimal minimal surfaces that we obtained using geocuts. In both cases we used the following boundary conditions. The "background" seeds were automatically set at voxels at the very border of the volume. The "object" seeds were manually placed in a single slice in the center of the object of interest. Then we applied a "max-flow" algorithm from [3] which (in one pass) finds a globally minimal surface (cut) among all surfaces that separate the "object" and "background" seeds. The running time for the examples in Figure 7 varies from 10 to 60 seconds on a 1.4GHz Pentium IV PC depending on the neighborhood size (up to 26).

In the experiments of Figure 7 we used an anisotropic Riemannian metric (induced from image I)

$$D(p) \;=\; g(|\nabla I|)\cdot \mathbf{I} \;+\; (1-g(|\nabla I|))\cdot \mathbf{u}\cdot \mathbf{u}^T$$

where $\mathbf{u} = \frac{\nabla I}{|\nabla I|}$ is a unit vector in the direction of image gradient at point p and $\mathbf{I}$ is the identity matrix. We used scalar function $g(x) = \exp(-\frac{x^2}{2\sigma^2})$. Note that in the coordinate system aligned with image gradient (s.t. $\mathbf{u} = (1,0,0)$) metric $D(p)$ is represented by the diagonal matrix

$$D(p) = \mathbf{diag}\,(\,1,\; g(|\nabla I|),\; g(|\nabla I|)\,)$$

It is known that anisotropic methods have advantages over isotropic techniques. At the same time, standard anisotropic diffusion techniques are typically slower than isotropic methods. Note that geocuts algorithm deals equally efficiently with isotropic or anisotropic metrics D.

5.2. Reducing metrication errors

Many standard graph-cut based methods (see Section 2.3) use energy functions that include a term penalizing segmentation discontinuities. Our results allow such terms to represent geometrically justified length/area of segments boundaries. This could significantly reduce metrication errors. Figure 8(b) shows such errors in the context of image restoration via graph cut technique in [4] (Potts model) using standard 4-neighborhood system. In (c) we show the results of the same algorithm when Potts model interaction

penalties were set from Euclidean cut metric weights in (4) using 8-neighborhoods. Similarly, Figure 8(e) shows metrication artifacts generated by a graph-cut based object extraction method in [2]. The results in (e) are for a simple 6-neighborhood system. The results in Figure 8(f) use 26-neighborhoods with geometrically justifies weights in (7).

In fact, the idea of using bigger neighborhoods is not new in the context of Dijkstra based methods on graphs. In the context of graph cuts, however, it was not clear how to set weights for different types of edges. Our theoretical results remove this problem. Formulas (4,7) explain how to set edge weights to obtain desired geometric properties for regular neighborhoods of any size and configuration.

Acknowledgments We would like to thank Christophe Chefdhotel, Nikos Paragios (Siemens Corp. Research), and Silviu Minut (Michigan State) for sharing their insights on geodesic active contour models and level-set methods. Gareth Funka-Lea (Siemens Corp. Reaserch) and Ramin Zabih (Cornell) helped to find better ways to present our work. Olga Veksler (NEC Labs USA) contributed image restoration results.

References

[1] W. Blaschke. *Vorlesungen über Integralgeometrie*. Hamburger Math. Einzelschriften, 1937.

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

[3] Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. In *International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition (EMMCVPR)*, number 2134 in LNCS, pages 359–374, Sophia Antipolis, France, September 2001. Springer-Verlag.

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

[5] M. P. D. Carmo. *Differential Geometry of Curves and Surfaces*. Prentice-Hall, Inc., 1976.

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

[7] V. Caselles, R. Kimmel, G. Sapiro, and C. Sbert. Minimal surfaces: A three dimensional segmentation approach. *Numerische Methematik*, 77(4):423–451, 1997.

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

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

[10] M. M. Deza and M. Laurent. *Geometry of Cuts and Metrics*. Springer Verlag, 1997.

[11] J. Favard. Définition de la longueur et de l'aire. *C. R. Acad. Sci. Paris*, 194:344–346, 1932.

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

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

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

[15] H. Ishikawa. Exact optimization for Markov Random Fields with convex priors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2003, to appear.

[16] M. Kass, A. Witkin, and D. Terzolpoulos. Snakes: Active contour models. *International Journal of Computer Vision*, 2:321–331, 1988.

[17] R. Kimmel, R. Malladi, and N. Sochen. Images as embedded maps and minimal surfaces: Movies, color, texture, and volumetric medical images. *International Journal of Computer Vision*, 39(2):111–129, 2000.

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

[19] V. Kwatra, A. Schodl, I. Essa, and A. Bobick. Graphcut textures: image and video synthesis using graph cuts. In *Proceedings of SIGGRAPH*, July 2003, to appear.

[20] N. Linial. Finite metric spaces – combinatorics, geometry and algorithms. In *International Congress of Mathematicians*, volume III, pages 573–586, Beijing, China, 2002.

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

[22] N. Paragios and R. Deriche. Geodesic Active Regions: A new framework to deal with frame partition problems in computer vision. *Journal of Visual Communication and Image Representation*, 13:249–268, 2002.

[23] S. Roy. Stereo without epipolar lines: A maximum-flow formulation. *International Journal of Computer Vision*, 34(2/3):147–162, August 1999.

[24] L. A. Santaló. *Integral Geometry and Geometric Probability*. Addison-Wesley, 1979 (first printing 1976).

[25] J. Sethian. *Level Set Methods and Fast Marching Methods*. Cambridge University Press, 1999.

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

[27] O. Veksler. Image segmentation by nested cuts. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 339–344, 2000.

[28] Z. Wu and R. Leahy. 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, November 1993.

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