An Improved Riemannian Metric Approximation for Graph Cuts

Ondřej Daněk and Pavel Matula

Centre for Biomedical Image Analysis, Faculty of Informatics
Masaryk University, Brno, Czech Republic
{xdanek2,pam}@fi.muni.cz

Abstract. Boykov and Kolmogorov showed that it is possible to find globally minimal contours and surfaces via graph cuts by embedding an appropriate metric approximation into the graph edge weights and derived the requisite formulas for Euclidean and Riemannian metrics [3]. In [9] we have proposed an improved Euclidean metric approximation that is invariant under (horizontal and vertical) mirroring, applicable to grids with anisotropic resolution and with a smaller approximation error. In this paper, we extend our method to general Riemannian metrics that are essential for graph cut based image segmentation or stereo matching. It is achieved by the introduction of a transformation reducing the Riemannian case to the Euclidean one and adjusting the formulas from [9] to be able to cope with non-orthogonal grids. We demonstrate that the proposed method yields smaller approximation errors than the previous approaches both in theory and practice.

1 Introduction

Combinatorial optimization using graph cuts presents a powerful energy minimization tool that has become very popular in the recent years in computer vision and image processing fields, mainly for its efficiency and applicability to a wide range of problems [5]. For instance, modern approaches to image segmentation often express the problem in terms of minimization of a suitable energy functional and can be solved via graph cut based algorithms if the energy is graph representable [11,2]. The graph cut segmentation framework has several advantages over traditional methods such as applicability to N-D problems, straightforward integration of various types of regional or geometric constraints or the ability to find a global minimum in polynomial time [4,2].

Many of the energy functionals used in computer vision such as the Chan-Vese segmentation model [8] or the geodesic active contours [7] require minimization of a length dependent term, e.g. length of the segmentation boundary under a given metric (Euclidean in the former example and Riemannian in the latter one). In order to minimize such energies using graph cuts the metric approximation has to be embedded into the graph. A seminal paper in this area is due to Boykov and Kolmogorov [3] who showed that graph cuts can be viewed as hypersurfaces and that it is possible to compute geodesics and globally minimal surfaces via

I. Debled-Rennesson et al. (Eds.): DGCI 2011, LNCS 6607, pp. 71–82, 2011.

graph cuts for Euclidean and Riemannian metrics with arbitrarily small error depending primarily on the size of the neighbourhood system that is used to connect the nodes in the graph.

In [9] we have devised an improved Euclidean metric approximation that is invariant under image mirroring, applicable to images with anisotropic resolution and with a smaller approximation error, especially for small neighbourhoods that are used in computer vision (large neighbourhoods result in better approximation but also longer computation and high memory consumption). We have presented the practical benefits of our method on the graph cut based minimization of the Chan-Vese segmentation model [16]. In this paper, we extend our method to general Riemannian metrics, making it applicable to the graph cut based geodesic segmentation model [3] or stereo matching [14]. To achieve this, we set up a transformation matrix that projects a Riemannian space with a locally constant metric tensor onto the Euclidean plane. Subsequently, due to the linearity of the transformation, we are able to exploit the Cauchy-Crofton based formulas from [9] to derive the edge weights approximating the Euclidean metric in the transformed space and obtain a good approximation for the original problem. Nevertheless, the formulas are adjusted to be able to cope with non-orthogonal grids that may arise during the transformation. To demonstrate that this method gives a better approximation than [3] we plot and compare the approximation error it yields for lines under different angular orientations and also show practical examples on the graph cut based geodesic segmentation.

To conclude the introduction, we also refer the reader to recent methods based on continuous maximum flows [13] allowing to find globally optimal surfaces for both isotropic and anisotropic Riemannian metrics [1,15] minimizing the metrication artefacts, i.e. offering superior results compared to the traditional combinatorial graph cuts (dealt with in this paper) in many situations.

This paper is organized as follows. In Section 2 we state the problem and briefly describe the concept of cut metrics including current approaches to Euclidean metric approximation via graph cuts. In Section 3 we present the state-of-the-art approach to Riemannian metric approximation and give details about the proposed method extending our previous work. The comparison of both methods is available in Section 4 and in Section 5 we provide discussion on their complexity and computational demands. We conclude the paper in Section 6.

2 Preliminaries

In this section we review the concept of cut metrics and explain the current approaches to Euclidean metric approximation via graph cuts. For simplicity we consider only the 2D case in the following text. However, the methods are directly extensible to 3D and this extension is covered in Section 3.4.

2.1 Cut Metrics

Given a connected weighted graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ a cut $\mathcal{C}$ is a partitioning of the graph nodes $\mathcal{V}$ into two disjoint subsets S and T such that $\mathcal{V} = S \cup T$. It is defined

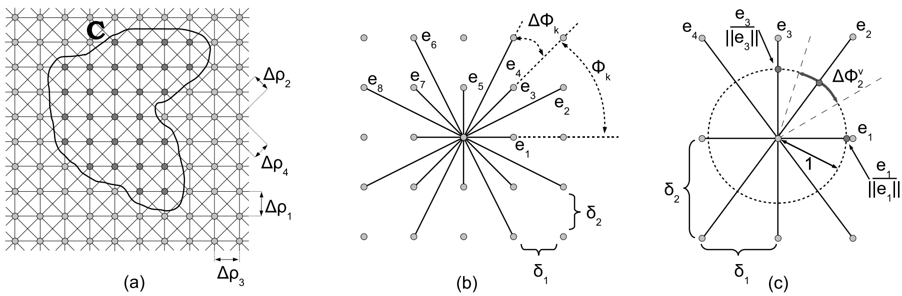

Fig. 1. (a) A grid graph with 8-neighbourhood system, $\Delta\rho_k$ denotes the distance between the closest lines generated by vector e_k. (b) 16-neighbourhood system and the computation of $\Delta\phi_k$ (c) 8-neighbourhood system and the computation of $\Delta\phi^v_k$.

by the set of edges connecting the nodes in S with nodes in T. The cost of a cut $|\mathcal{C}|_\mathcal{G}$ is the sum of the weights of its edges. Lets assume $\mathcal{G}$ is embedded in a regular orthogonal grid. Example of such graph is depicted in Fig. 1. In [3] Boykov and Kolmogorov showed that cuts in grid graphs can be viewed as hypersurfaces (in our case as contours) and vice versa. Looking at Fig. 1a we can see that the closed contour C naturally defines a cut in $\mathcal{G}$ by partitioning the graph nodes into two groups S and T depending on whether they lie inside or outside the contour. Alternatively, the cut induced by the contour can be defined as the set of edges it severs[1]. Due to this geometric meaning of cuts we can naturally interpret the cost of a cut as the "length" of a corresponding contour.

Now, imagine the image segmentation problem. In graph cut based image segmentation [2] the input image is converted to a graph where every node corresponds to an image voxel and the nodes are connected according to a chosen neighbourhood system. A grid graph is obtained where every cut can be viewed as a binary (or foreground/background) segmentation of the image as they both partition the image into two parts. Moreover, if the edge weights are chosen appropriately, the cost of every cut can resemble the energy of a chosen energy functional (e.g. the Chan-Vese model [8,16]) of the corresponding segmentation. Thus, minimizer of the energy can be found by finding a *minimal cut* in the graph. This is a fast operation that can be performed in polynomial time [4].

However, as mentioned earlier many of the energy functionals [7,8] contain a length dependent term. Usually, the length of the segmentation boundary is being minimized under a chosen metric. Hence, a metric approximation has to be embedded into the graph in order to optimize such energy functional via graph cuts. Because we have already explained that the cost of a cut can be interpreted as the length of a corresponding contour (in this case, the segmentation boundary plays the role of the contour) all we have to do is to choose the edge weights appropriately such that for any cut its cost approximates the length of the corresponding contour under a chosen metric. In [3] Boykov and

[1] Obviously, for a grid with a finite resolution the correspondence between a cut and a contour is not unique and multiple contours can be found representing the same cut.

Kolmogorov showed that this can be achieved with arbitrarily small error and devised the requisite edge weight formulas for Euclidean and Riemannian metrics. The complete set of metrics that can be approximated via graph cuts has then been described in [10].

2.2 Euclidean Metric Approximation

Lets assume the graph $\mathcal{G}$ is embedded in a regular orthogonal 2D grid with all nodes having topologically identical neighbourhood system and with δ_1 and δ_2 determining the spacing between the nodes in horizontal and vertical direction, respectively. An example of such a graph with 8-neighbourhood system and $\delta_1 = \delta_2$ is depicted in Fig. 1a. Further, let the neighbourhood system $\mathcal{N}$ be described by a set of vectors $\mathcal{N} = \{e_1, \ldots, e_m\}$. We assume that the vectors are listed in the increasing order of their angular orientation $0 \leq \phi_k < \pi$. We also assume that vectors e_k are undirected (we do not differentiate between e_k and $-e_k$) and shortest possible in given direction, e.g. 16-neighbourhood is represented by a set of 8 vectors $\mathcal{N}_{16} = \{e_1, \ldots, e_8\}$ as depicted in Fig. 1b. Finally, we define the distance between the nearest lines generated by vector e_k in the grid as $\Delta\rho_k$ (for the 8-neighbourhood system these are depicted in Fig. 1a).

Assuming to each edge e_k is assigned a particular weight $w_k^{\mathcal{E}}$ (this weight is the same for all nodes), imagine we are given a regular curve C as shown in Fig. 1a. Further, lets assume the cut induced by the contour is defined by the set of edges it intersects. The question is how to set the edge weights so that the capacity of the cut $|\mathcal{C}|_{\mathcal{G}}$ approximates the Euclidean length $|C|_{\mathcal{E}}$ of the contour. The method of [3,9] is based on the Cauchy-Crofton formula that links Euclidean length $|C|_{\mathcal{E}}$ of a contour C with a measure of a set of lines intersecting it:

$$|C|_{\mathcal{E}} = \frac{1}{2} \int_{\mathcal{L}} n_c(l)\, dl\,, \tag{1}$$

where $\mathcal{L}$ is the space of all lines and $n_c(l)$ is the number of intersections of line l with the contour C. Because every line in a plane is uniquely identified by its angular orientation ϕ and distance ρ from the origin the formula can be rewritten in the form:

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

discretizing the formula and following the derivation steps of [3,9] we end up with the following edge weights:

$$w_k^{\mathcal{E}} = \frac{\Delta\rho_k \Delta\phi_k}{2}\,, \tag{3}$$

where, as pointed out in [3], the distance between the closest lines generated by vector e_k in the grid can be calculated by dividing the area of the grid cell by the length of e_k:

$$\Delta\rho_k = \frac{\delta_1 \delta_2}{||e_k||}\,. \tag{4}$$

According to [3] using these weights $|\mathcal{C}|_{\mathcal{G}}$ converges to $|C|_{\mathcal{E}}$ as δ_1, δ_2, $\sup_k \Delta\phi_k$ and $\sup_k ||e_k||$ get to zero. Namely the value of $\sup_k \Delta\phi_k$ is important as it can be easily controlled by the choice of the neighbourhood. For dense neighbourhoods a very good approximation may be obtained [9].

Unfortunately, due to the computational demands small neighbourhoods are used in practice. For this reason, we have introduce an improved method in [9] having a smaller approximation error. It consists in replacing $\Delta\phi_k$ in Eq. 3 with:

$$\Delta\phi_k^v = \frac{\Delta\phi_k + \Delta\phi_{k-1}}{2} \, . \tag{5}$$

This modification attempts to calculate a better partitioning of the space of all angular orientations among the vectors e_k. In 2D this space is represented by a unit circle and points $\frac{e_k}{||e_k||}$ lying on the circle can be treated as samples from this space. To partition the space among these points a Voronoi diagram is computed on the unit circle and $\Delta\phi_k^v$ is chosen as the length of the Voronoi cell (circular arc) corresponding to $\frac{e_k}{||e_k||}$. Alternatively, $\Delta\phi_k^v$ is the measure of the lines closest to e_k in terms of their angular orientation. The whole construction is depicted in Fig. 1c and reduces to the Eq. 5 in 2D. Moreover, besides having a smaller approximation error this construction is also invariant under (horizontal and vertical) mirroring and directly generalizable to higher dimensions which the original approach of [3] from Fig. 1b is not.

3 Riemannian Metrics

3.1 Introduction

Giving all the theory on Riemannian geometry is beyond the scope of this paper, therefore we confine ourselves only to the basic notation required for proper understanding of the rest of this section. In Riemannian geometry each point of the space is associated with a *metric tensor* M that controls how an inner product of two vectors is calculated. This tensor is a symmetric positive definite matrix (a bilinear form) that varies smoothly over the space. In case M is constant the Riemannian norm of a vector u is calculated as:

$$||u||_{\mathcal{R}} = \sqrt{u^T \cdot M \cdot u} \tag{6}$$

A standard Euclidean norm is obtained when M equals to the identity matrix. In the non-constant case the distance between two points in a Riemannian space does depend on their displacement from the origin, as opposed to the Euclidean geometry. However, full understanding of this case is not necessary for the oncoming derivations.

Geometrical interpretation of the metric tensor is intuitive and expresses local contraction or dilation of the space. The properties follow from the spectral decomposition of M. Because M is symmetric positive definite it has two real-valued eigenvalues λ_1 and λ_2 and two orthogonal eigenvectors. The space is then "stretched" in the directions corresponding to the eigenvectors by the value of

$\sqrt{\lambda_1}$ and $\sqrt{\lambda_2}$. This behaviour has interesting utilizations in the image processing field. In [7] Caselles et al. showed that the image segmentation problem can be solved by finding the shortest path (geodesic) in a Riemannian space with an image derived metric tensor. In each voxel the tensor dilates the space in the direction of image gradient making the segmentation follow edges and object boundaries. Another interesting application is in stereo matching [14].

3.2 State of the Art

Until the seminal work of Boykov and Kolmogorov [3] the geodesic segmentation problem was solved solely using the level set framework [12]. In [3] Boykov and Kolmogorov showed that in some sense graph cuts can be treated as a combinatorial counterpart of the level set method, both having implicit boundary representation (see also [2]). Using integral geometry they derived the edge weight formulas for Euclidean as well as general Riemannian metrics allowing graph cut based solution to the geodesic segmentation problem. The formulas have the following form in 2D:

$$w_k^{\mathcal{R}} = w_k^{\mathcal{E}} \cdot \frac{det\, M}{(u_k^T \cdot M \cdot u_k)^{3/2}}\,, \tag{7}$$

where u_k is a unit vector in the direction of e_k. As can be seen, in their approach the edge weights for a Riemannian metric are obtained by multiplying the Euclidean metric weights by a coefficient depending on the metric tensor M and the direction of e_k (the second term vanishes when M is the identity matrix). In the following subsection we propose a different method and show that it has a smaller approximation error.

3.3 Proposed Method

Like in the Euclidean case, let us assume the graph $\mathcal{G}$ is embedded in a regular orthogonal 2D grid with all nodes having topologically identical neighbourhood system $\mathcal{N} = \{e_1, \ldots, e_m\}$. Examples of $\mathcal{N}_8$ and $\mathcal{N}_4$ grid graphs are depicted in Fig. 1a and Fig. 2a, respectively. In addition, we assume that the graph is placed in a Riemannian space with the metric tensor sampled in the graph nodes, i.e. each graph node p is associated with a metric tensor $M(p)$ (e.g. the image derived metric tensor [3]). Without loss of generality we also assume that the spacing between the graph nodes is 1 in all directions. Because $M(p)$ already defines space stretching it is possible to embed the grid resolution right into the matrix M (more on this topic later in this section). Finally, our aim is to set the edge weights $w_k^{\mathcal{R}}$ such that for any contour C the cost of the corresponding cut $|\mathcal{C}|_{\mathcal{G}}$ (i.e. the sum of the weights of the edges the contour severs) approximates the Riemannian length of the contour $|C|_{\mathcal{R}}$.

Choosing one of the graph nodes, let us assume the metric tensor $M(p)$ (we will omit the node specification in the following text) is locally constant around p. Further, recall that the geometric interpretation of M is space dilation and that the space is dilated by the amount corresponding to the square root of the

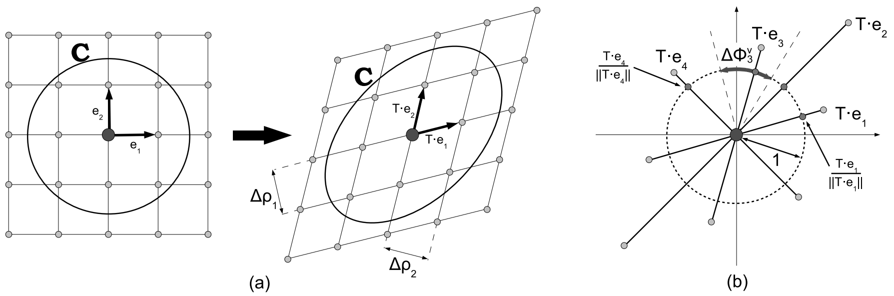

Fig. 2. (a) A grid graph with 4-neighbourhood system and a circular contour and the effect of space transformation by matrix T. $\Delta\rho_k$ is the distance between the closest lines generated in the transformed grid by the transformed vectors e_k. (b) 8-neighbourhood system transformed by the matrix T and the computation of $\Delta\phi_k^v$.

eigenvalues of M. To reflect this, we set up a symmetric transformation matrix T having the same eigenvectors as M but with eigenvalues being the square root of those of M. We have that:

$$M = T^T \cdot T\,. \tag{8}$$

Subsequently, the matrix T can be used to project the Riemannian space with a locally constant metric tensor onto the Euclidean plane. It holds that measuring Euclidean distances in the transformed space is equivalent to measuring Riemannian distances in the original space.

Lemma 1. *Given two points u and v, a constant metric tensor M and the corresponding transformation matrix T, it holds that the Euclidean distance of $T \cdot u$ and $T \cdot v$ equals to the Riemannian distance of u and v.*

Proof. $d_{\mathcal{E}}(T \cdot u, T \cdot v) = ||T \cdot (u - v)||_{\mathcal{E}} = \sqrt{(T \cdot (u-v))^T \cdot (T \cdot (u-v))} =$
$= \sqrt{(u-v)^T \cdot T^T \cdot T \cdot (u-v)} = \sqrt{(u-v)^T \cdot M \cdot (u-v)} = ||u-v||_{\mathcal{R}} =$
$= d_{\mathcal{R}}(u, v)$ □

The effect of the space transformation using the matrix T on the local neighbourhood of graph node p is illustrated in Fig. 2a. Depending on T we may obtain a skewed non-orthogonal grid. However, note that due to the linearity of the transformation, the number of intersections between the grid and the contour C is preserved. This has an important corollary – if the edge weights are chosen so that the cut cost approximates the Euclidean length in the transformed space we would obtain also approximation of the original Riemannian length.

The last question remains, how to set up the edge weights so that the cut cost approximates the Euclidean length in the transformed space. Because the space is Euclidean, Eq. 3 based on the Cauchy-Crofton formula for Euclidean spaces is sufficient to obtain the edge weights. The only difference is the core set of lines used to compute $\Delta\rho_k$ and $\Delta\phi_k^v$ where the lines generated by the transformed neighbourhood system have to be considered. Because the transformed grid is potentially non-orthogonal, Eq. 4 can no longer be used. However, the distance between the closest lines generated in the grid by vector $T \cdot e_k$ can still be calculated by dividing the area of the grid cell by the length of $T \cdot e_k$:

$$\Delta\rho_k = \frac{det\, T}{||T \cdot e_k||_{\mathcal{E}}} = \frac{\sqrt{det\, M}}{||e_k||_{\mathcal{R}}} . \tag{9}$$

The calculation of $\Delta\phi_k^v$ remains the same, i.e. using a Voronoi diagram on a unit hypersphere (circle in 2D) the measure of angular orientations closest to $T \cdot e_k$ is computed as illustrated in Fig. 2b. Alternatively, in 2D it can be obtained by averaging the angles between $T \cdot e_k$ and its two neighbours.

So far, we have considered only local behaviour of the Riemannian space. To obtain a generalization for the general case with varying metric tensor a different metric tensors $M(p)$ is used to compute the edge weights according to the derived formulas for each graph node p. Thus, w_k is no longer the same across all nodes as in the Euclidean case. Finally, notice that the derived formulas are in accordance with our previous results. In particular, the Euclidean space with node spacing δ_1 and δ_2 from Section 2.2 can be simulated using the following constant metric tensor and the corresponding transformation matrix:

$$M = \begin{pmatrix} \delta_1^2 & 0 \\ 0 & \delta_2^2 \end{pmatrix} \qquad T = \begin{pmatrix} \delta_1 & 0 \\ 0 & \delta_2 \end{pmatrix} , \tag{10}$$

where M embeds the space stretching along x and y axis of the coordinate system by δ_1 and δ_2, respectively. Using this matrix Eq. 9 reduces to Eq. 4 and the same edge weights are derived.

3.4 Extension to 3D

The Cauchy-Crofton based edge weights have the following form in 3D [3,9]:

$$w_k^{\mathcal{E}} = \frac{\Delta\rho_k \Delta\phi_k^v}{\pi} . \tag{11}$$

To compute $\Delta\rho_k$ Eq. 9 may be used as it is. The space of all angular orientations is represented by a unit sphere surface in 3D. The sphere surface area has to be partitioned among the points $\frac{T \cdot e_k}{||T \cdot e_k||}$ using a spherical Voronoi diagram [6] to obtain $\Delta\phi_k^v$.

4 Experimental Results

In this section, we present two experiments in which we compare our method with the method proposed by Boykov and Kolmogorov (BK) [3]. The first experiment measures the metrication error of both approximations for lines under different angular orientations. Assuming a straight line under all possible angular orientations we compare its length in the cut metric (i.e. the sum of the weights of the edges it severs) to the ground-truth (i.e. the actual Riemannian

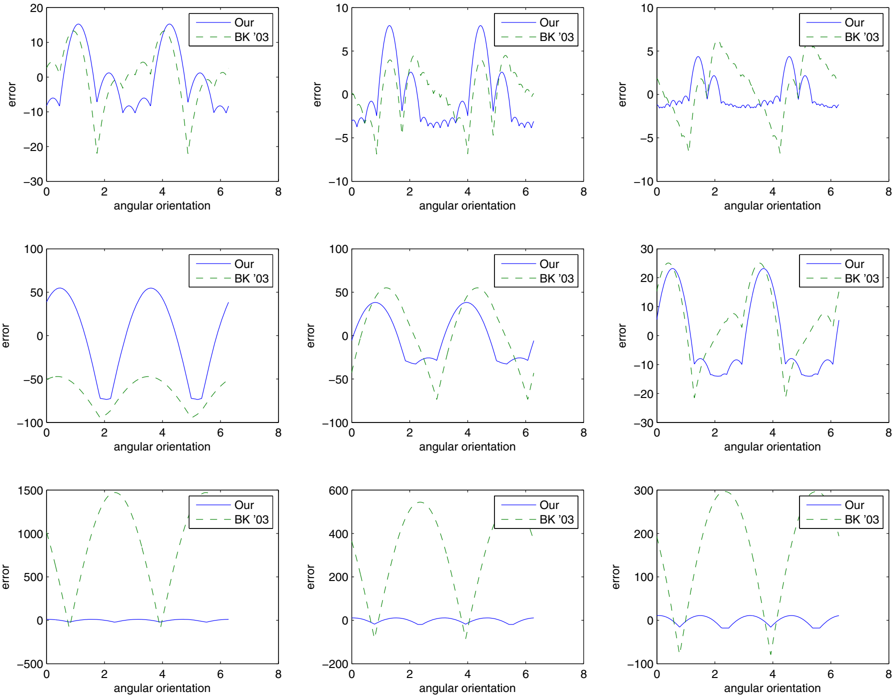

Fig. 3. Multiplicative metrication error (in percents) of the approximation for various metric tensors. Comparison of our and Boykov and Kolmogorov's [3] method. Left column: $\mathcal{N}_8$. Middle column: $\mathcal{N}_{16}$. Right column: $\mathcal{N}_{32}$. Top row: Space dilation under 10° by a factor of 2. Middle row: Space dilation under 60° by a factor of 15. Bottom row: Space dilation under 45° by a factor of 40.

length of the line) for several combinations of neighbourhood and a metric tensor. The multiplicative error of both approximations is depicted in Fig. 3. For small stretch factors (first two rows) the performance of both approaches is similar. Nevertheless, while the error of our method oscillates around zero in all cases, the BK method behaves unexpectedly in some situations underestimating the length under all angular orientations (first graph on the second row). With growing stretch factor the maximal error grows for both methods, particularly due to the growing value of $\sup_k \Delta\phi_k$ (noticeable in Fig. 2b). However, for larger stretch factors there is a huge difference in favor of our method. Especially for small neighbourhoods the assumption of infinitely small $\Delta\phi_k$ required to derive Eq. 7 is unrealistic causing a considerable error. Unfortunately, in practice even much larger stretch factors are common in which case the performance of both considered methods is going to be relatively poor unless a very dense neighbourhood is used.

In the second experiment we have conducted a comparison on a practical image segmentation problem using the geodesic segmentation model [7,3]. We have used a 16-neighbourhood and the same parameters for both methods. For

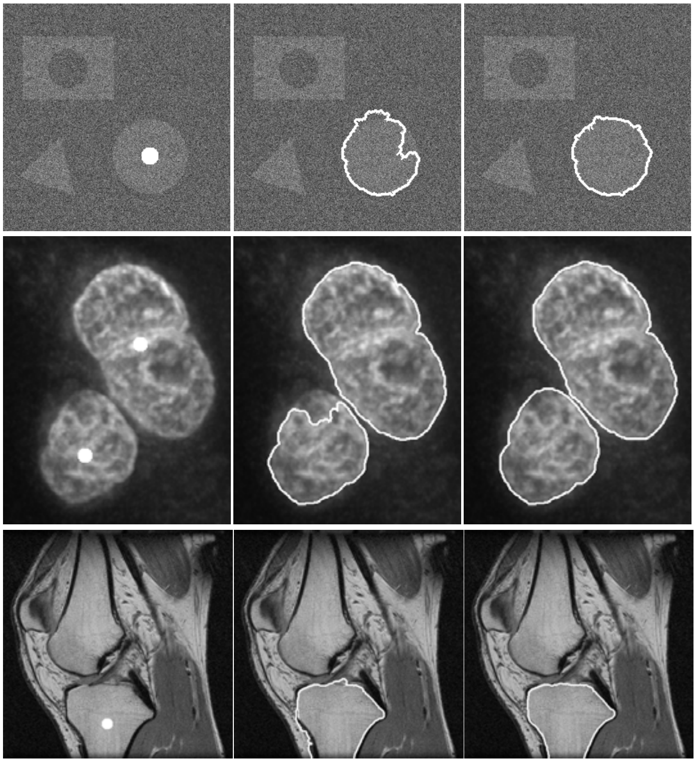

Fig. 4. Graph cut based geodesic image segmentation. Left column: Input image – synthetic and biomedical data. White foreground marker is placed inside the object(s) to be segmented. Middle column: Segmentation result using the Riemannian metric approximation of [3]. Right column: Result using the proposed method. 16-neighbourhood and the same parameters were used for both methods.

each of the images depicted in Fig. 4 a foreground marker was placed inside the objects to be segmented with background marker being the border of the image. Subsequently, image gradient was calculated in each pixel and a metric tensor was constructed dilating the space in the direction of the gradient. Exact formula for construction of the matrix M is available in [3]. Finally, a geodesic in the Riemannian space was found separating the foreground and background markers using the graph cut algorithm yielding the segmentation boundary. Apparently, our approach to the Riemannian metric approximation gives significantly better results for the particular examples. Besides these experiments, more practical tests were performed and, generally speaking, we have not found any example in which our method gives inferior results.

5 Discussion

In this section we would like to focus on the computational demands of the proposed method. As we have already demonstrated we are able to achieve smaller approximation errors, however, this comes for a price. While the BK method consisting of Eq. 7 has generally constant complexity (few basic mathematical operations) our method is clearly more demanding requiring the computation of circular or spherical Voronoi diagram in 2D or 3D, respectively, to obtain $\Delta\phi_k^v$. While this does not present a computational burden in 2D (it is equally fast when the neighbourhood vectors are sorted according to their angular orientation beforehand) it may cause a significant computational overhead in 3D. For instance, in cases where the metric tensor differs in each pixel, the spherical Voronoi diagram has to be recalculated repeatedly which results in our method being approximately 20 times slower making it hardly feasible. However, it is still useful in case of scalar (i.e., isotropic) or constant (such as those simulating anisotropic resolution in 2D and 3D [9]) Riemannian metrics where it is enough to compute the Voronoi diagram only once. Then again the method is equally fast as the BK approach. Also note that the difference between the two approximations can be sometimes minor in practice. Thus, a decision has to be always made regarding the trade-off between the precision and speed.

Another complication in our method that can be encountered is the construction of the transformation matrix T. In general, the construction of T from the matrix M requires spectral decomposition of M which can be slow. However, this can be avoided in most situations by directly constructing the matrix T instead of M such as in the image segmentation example where the eigenvectors (the image gradient) and eigenvalues (computed from the length of the gradient) are known so T can be constructed directly.

6 Conclusion

In this paper, we have presented a novel method for Riemannian metric approximation via graph cuts. It is based on the Cauchy-Crofton formula and is a generalization of our previously devised formulas for the Euclidean metric approximation on anisotropic grids. Using a transformation matrix we reduce the Riemannian case to the Euclidean one and modify our edge weight formulas to be able to cope with non-orthogonal grids. The proposed approximation has a smaller metrication error than other state-of-the-art methods which was verified both in theoretical and practical experiments. Implementation of the method described in this paper can be downloaded from *http://cbia.fi.muni.cz/projects/graph-cut-library.html*.

Acknowledgment. This work has been supported by the Ministry of Education of the Czech Republic (Projects No. MSM-0021622419, No. LC535 and No. 2B06052).

References

1. Appleton, B., Talbot, H.: Globally minimal surfaces by continuous maximal flows. IEEE Transactions on Pattern Analysis and Machine Intelligence 28, 106–118 (2006)
2. Boykov, Y., Funka-Lea, G.: Graph cuts and efficient n-d image segmentation (review). International Journal of Computer Vision 70(2), 109–131 (2006)

3. Boykov, Y., Kolmogorov, V.: Computing geodesics and minimal surfaces via graph cuts. In: ICCV 2003: Proceedings of the Ninth IEEE International Conference on Computer Vision, p. 26 (2003)
4. Boykov, Y., Kolmogorov, V.: An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. IEEE Transactions on Pattern Analysis and Machine Intelligence 26(9), 1124–1137 (2004)
5. Boykov, Y., Veksler, O.: Graph Cuts in Vision and Graphics: Theories and Applications. In: Handbook of Mathematical Models in Computer Vision, pp. 79–96. Springer, Heidelberg (2006)
6. Brown, K.Q.: Geometric transforms for fast geometric algorithms. Ph.D. thesis, Carnegie Mellon University, Pittsburgh, PA, USA (1979)
7. Caselles, V., Kimmel, R., Sapiro, G.: Geodesic active contours. In: IEEE International Conference on Computer Vision, pp. 694–699 (1995)
8. Chan, T.F., Vese, L.A.: Active contours without edges. IEEE Transactions on Image Processing 10(2), 266–277 (2001)
9. Daněk, O., Matula, P.: Graph cuts and approximation of the euclidean metric on anisotropic grids. In: VISAPP 2010: International Conference on Computer Vision Theory and Applications, vol. 2, pp. 68–73 (2010)
10. Kolmogorov, V., Boykov, Y.: What metrics can be approximated by geo-cuts, or global optimization of length/area and flux. In: ICCV 2005: Proceedings of the Tenth IEEE International Conference on Computer Vision, pp. 564–571 (2005)
11. Kolmogorov, V., Zabih, R.: What energy functions can be minimized via graph cuts? IEEE Transactions on Pattern Analysis and Machine Intelligence 26(2), 147–159 (2004)
12. Osher, S., Sethian, J.A.: Fronts propagating with curvature dependent speed: Algorithms based on Hamilton–Jacobi formulation. Journal of Computational Physics 79(1), 12–49 (1988)
13. Strang, G.: Maximum flows and minimum cuts in the plane. Journal of Global Optimization 47, 527–535 (2010)
14. Vogiatzis, G., Torr, P.H.S., Cipolla, R.: Multi-view stereo via volumetric graph-cuts. In: CVPR 2005: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2, pp. 391–398 (June 2005)
15. Zach, C., Niethammer, M., Frahm, J.M.: Continuous maximal flows and wulff shapes: Application to mrfs. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 1911–1918 (2009)
16. Zeng, Y., Chen, W., Peng, Q.: Efficiently solving the piecewise constant mumford-shah model using graph cuts. Tech. rep., Dept. of Computer Science, Zhejiang University, P.R. China (2006)