

An Experimental Comparison of Discrete and Continuous Shape Optimization Methods

Maria Klodt, Thomas Schoenemann, Kalin Kolev,
Marek Schikora, and Daniel Cremers

Department of Computer Science,
University of Bonn, Germany
{klodt,tosch,kolev,schikora,dcremers}@cs.uni-bonn.de

Abstract. Shape optimization is a problem which arises in numerous computer vision problems such as image segmentation and multiview reconstruction. In this paper, we focus on a certain class of binary labeling problems which can be globally optimized both in a spatially discrete setting and in a spatially continuous setting. The main contribution of this paper is to present a quantitative comparison of the reconstruction accuracy and computation times which allows to assess some of the strengths and limitations of both approaches. We also present a novel method to approximate length regularity in a graph cut based framework: Instead of using pairwise terms we introduce higher order terms. These allow to represent a more accurate discretization of the L_2 -norm in the length term.

1 Introduction

Shape optimization is at the heart of several classical computer vision problems. Following a series of seminal papers [2, 12, 15, 21, 27], functional minimization has become the established paradigm for these problems. In the spatially discrete setting the study of the corresponding binary labeling problems goes back to the spin-glas models introduced in the 1920's [19]. In this paper, we focus on a class of functionals of the form:

$$E(S) = \int_{int(S)} f(x) d^n x + \nu \int_S g(x) dS, \quad (1)$$

where S denotes a hypersurface in \mathbb{R}^n , i.e. a set of closed boundaries in the case of $2D$ image segmentation or a set of closed surfaces in the case of $3D$ segmentation and multiview reconstruction. The functions $f : \mathbb{R}^n \rightarrow \mathbb{R}$ and $g : \mathbb{R}^n \rightarrow \mathbb{R}^+$ are application dependent. In a statistical framework for image segmentation, for example, $f(x) = \log p_{bg}(I(x)) - \log p_{ob}(I(x))$ may denote the log likelihood ratio for observing the intensity $I(x)$ at any given point x given that x is part of the background or the object, respectively.

The second term in (1) corresponds to an isotropic measure of area (for $n = 3$) or boundary length ($n = 2$), measured by the function g .

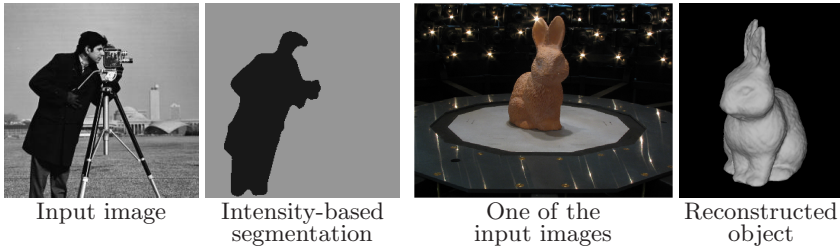


Fig. 1. Examples of shape optimization: Image segmentation and 3D reconstruction

In the context of image segmentation, g may be a measure of the local edge strength – as in the geodesic active contours [6, 22] – which energetically favors segmentation boundaries along strong intensity gradients. In the context of multiview reconstruction, $g(x)$ is typically a measure of the consistency among different views of the voxel x , where low values of g indicate a strong agreement from different cameras on the observed patch intensity – see for example [12]. Figure 1 shows examples of shape optimization using the example of image segmentation and multiview reconstruction.

Functionals of the form (1) can be globally optimized by reverting to implicit representations of the hypersurface S using an indicator function $u : \mathbb{R}^n \rightarrow \{0, 1\}$, where $u = 1$ and $u = 0$ denote the interior and exterior of S . The functional (1) defined on the space of surfaces S is therefore equivalent to the functional

$$E(u) = \int_{\mathbb{R}^n} f(x) u(x) \, d^n x + \nu \int_{\mathbb{R}^n} g(x) |\nabla u(x)| \, d^n x, \quad (2)$$

defined on the space of binary labelings u , where the second term in (2) is the weighted total variation norm which can be extended to non-differentiable functions in a weak sense.

In the current experimental comparison, we focus on functionals of the type (1) since they allow for the efficient computation of globally optimal solutions of region-based functionals. There exist numerous alternative functionals for shape optimization, including ratio functionals [20, 29]. Recently it was shown that some region-based ratio functionals can be optimized globally [25]. As this method did not yet reach a high popularity, we leave it for future discussion.

The functional (2) can be globally optimized in a spatially discrete setting: By mapping each labeling to a cut in a graph, the problem is reduced to computing the minimal cut. First suggested in [18], it was later rediscovered in [5] and has since become a popular framework for image segmentation [28] and multiview reconstruction [31]. More recently it was shown in [7, 8] that the same binary labeling problem (2) can be globally minimized in a spatially *continuous* setting as well. An alternative spatially continuous formulation of graph cuts was developed in [1].

In this paper, we propose the first quantitative experimental comparison of spatially discrete and spatially continuous optimization methods for functionals

of the form (2). In particular, we focus on the quality and efficiency of shape optimization in discrete and continuous setting. Furthermore we propose a new approximation of the L_2 -norm in the context of graph cuts based optimization.

2 Spatially Discrete Optimization Via Graph Cuts

To solve the binary labeling problem (2) in a discrete setting, the input data is converted into a directed graph in form of a regular lattice: Each pixel (or voxel) in the input data corresponds to a node in the lattice. To approximate the metric g measuring the boundary size of the hypersurface S , neighboring nodes are connected. The degree of connectivity depends on the application. We defer details to Section 2.2.

Additionally a source node s and a sink node t are introduced. They allow to include the unary terms $f(x)u(x)$ for the pixels x : If $f(x) \geq 0$, an edge to the source is introduced, weighted with $f(x)$. Otherwise an edge to the sink weighted with $-f(x)$ is created.

The optimal binary labeling u corresponds to the minimal s/t -cut in the graph. An s/t -cut is a partitioning of the nodes in the graph into two sets S and T , where S contains the source s and T the sink t . Nodes $x \in S$ are assigned the label $u(x) = 0$, nodes $x \in T$ the label $u(x) = 1$. The weight of such a cut is the sum of the weights of all edges starting in S and ending in T .

2.1 Computing the Minimal Cut in a Graph

Efficient solvers of the minimal s/t -cut problem are based on computing the maximal flow in the graph [13]. Such methods are divided into three major categories: those based on augmenting paths [4, 11, 13], blocking flows [10, 17] and the push-relabel method [16]. Some of these methods do not guarantee a polynomial running time [4] or require integral edge weights [17]. To solve 2-dimensional problems of form (2) usually the algorithm of Boykov and Kolmogorov performs best [4]. For highly connected three-dimensional grids the performance of this algorithm breaks down [4] and push-relabel methods become competitive. Recently efforts were made to parallelize push-relabel-based approaches [9].

2.2 Approximating Metrics Using Graph Cuts

The question of how to approximate continuous metrics of the boundary size in a discrete setting has received significant attention by researchers. Boykov and Kolmogorov [3] show how to approximate any Riemannian metric, including anisotropic ones. In [24] they discuss how to integrate flux. A similar construction can be derived from the divergence theorem. In the following we limit our discussion to the isotropic case.

We start with a review of the method in [3] which replaces the L_2 -norm of the gradient in (2) by its L_1 -norm. For the Euclidean metric ($g(x) = 1 \forall x \in \mathbb{R}^n$) we then propose a novel discretization scheme which allows to use the L_2 -norm of the gradient by introducing higher order terms.

Approximation Using Pairwise Terms. Based on the Cauchy-Crofton formula of integral geometry, Boykov and Kolmogorov [3] showed that the metric given by g can be approximated by connecting pixels to all pixels in a given neighborhood. The respective neighborhood systems can be expressed as

$$N_R(x) = \left\{ x + \begin{pmatrix} a \\ b \end{pmatrix} \mid a, b \in \mathbb{Z}, \sqrt{a^2 + b^2} \leq R, \gcd(|a|, |b|) = 1 \right\}.$$

The constraint on the greatest common divisor avoids duplicate directions. The edge corresponding to $(a \ b)^\top$ is given a weight of $g(x)/\sqrt{a^2 + b^2}$. For $R = 1$ the obtained 4-connected lattice reflects the L_1 -norm of the gradient. With increasing R and decreasing grid spacing the measure converges to the continuous measure. This is not true when fixing the connectivity (i.e. when keeping R constant).

A Novel Length Approximation Using Higher Order Terms. The energy (2) involves the L_2 -norm of the generalized gradient of the $\{0, 1\}$ -function u . With the pairwise terms discussed above a large connectivity is needed to approximate this norm. In the following, we will show that a more accurate approximation of the L_2 -norm can be integrated in a graph cut framework, without increasing the connectivity. The key observation is that in a two-dimensional space a consistent calculation of the gradient is obtained by taking the differences to the upper and left neighbor in the grid - see Figure 2.

The Figure also shows the arising term. One easily verifies that this term satisfies the submodularity condition [26]. For a third order term as this one, this condition implies that the term can be minimized using graph cuts.

We also considered the corresponding term in 3D space where each pixel is connected to three neighbors. The arising fourth order term – with values in $\{0, 1, \sqrt{2}, \sqrt{3}\}$ – is submodular. However it is not clear whether it can be minimized via graph cuts: It does not satisfy the sufficient conditions pointed out by Freedman [14].

From a practical point of view, in 2D the novel terms do not perform well: The length discretization only compares a pixel to those pixels in the direction of the upper left quadrant. Performance is boosted when adding the respective terms for the other three quadrants as well.

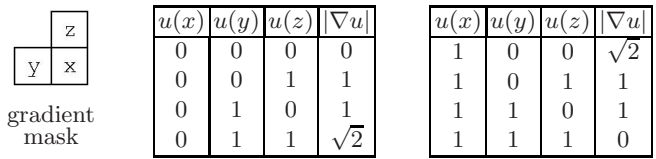


Fig. 2. The L_2 -norm of the 2D gradient as a ternary term. One easily verifies that this term is submodular.

3 Spatially Continuous Optimization Via Relaxation

More recently, it was shown that the class of functionals (2) can also be minimized in a spatially continuous setting by reverting to convex relaxations [7, 8]. By relaxing the binary constraint and allowing the function u to take on values in the interval between 0 and 1, the optimization problem becomes minimizing the convex functional (2) over the convex set

$$u : \mathbb{R}^n \rightarrow [0, 1]. \tag{3}$$

Global minimizers u^* of this relaxed problem can efficiently be computed (see section 3.2).

3.1 Convex Relaxation and the Thresholding Theorem

The following theorem [8, 30] assures that thresholding the solution u^* of the relaxed problem provides a minimizer of the original binary labeling problem (2). In other words the convex relaxation preserves global optimality for the original binary labeling problem.

Theorem 1. *Let $u^* : \mathbb{R}^n \rightarrow [0, 1]$ be a global minimizer of the functional (2). Then all upper level sets (i.e. thresholded versions)*

$$\Sigma_{\mu, u^*} = \{x \in \mathbb{R}^n \mid u^*(x) > \mu\}, \quad \mu \in (0, 1), \tag{4}$$

of u^ are minimizers of the original binary labeling problem (1).*

Proof. Using the layer cake representation of the function $u^* : \mathbb{R}^n \rightarrow [0, 1]$:

$$u^*(x) = \int_0^1 1_{\Sigma_{\mu, u^*}}(x) \, d\mu \tag{5}$$

we can rewrite the first term in the functional (2) as

$$\int_{\mathbb{R}^n} f u^* \, dx = \int_{\mathbb{R}^n} f \left(\int_0^1 1_{\Sigma_{\mu, x}} \, d\mu \right) \, dx = \int_0^1 \int_{\Sigma_{\mu, u^*}} f(x) \, dx \tag{6}$$

As a consequence, the functional (2) takes on the form:

$$E(u^*) = \int_0^1 \left\{ \int_{\Sigma_{\mu, u^*}} f \, dx + |\partial \Sigma_{\mu, u^*}|_g \right\} \, d\mu \equiv \int_0^1 \hat{E}(\Sigma_{\mu, u^*}) \, d\mu, \tag{7}$$

where we have used the coarea formula to express the weighted total variation norm in (2) as the integral over the length of all level lines of u measured in the norm induced by g . Clearly the functional (7) is now merely an integral of the original binary labeling problem \hat{E} applied to the upper level sets of u^* .

Assume that for some threshold value $\tilde{\mu} \in (0, 1)$ theorem 1 was not true, i.e. there exists a minimizer Σ^* of the binary labeling problem with smaller energy:

$$\hat{E}(\Sigma^*) < \hat{E}(\Sigma_{\tilde{\mu}, u^*}). \tag{8}$$

Then for the indicator function 1_{Σ^*} of the set Σ^* we have:

$$E(1_{\Sigma^*}) = \int_0^1 \hat{E}(\Sigma^*) \, d\mu < \int_0^1 \hat{E}(\Sigma_{\mu, u^*}) \, d\mu = E(u^*), \tag{9}$$

which contradicts the assumption that u^* was a global minimizer of (2). \square

Global minimizers of the functional (2) in a spatially continuous setting are therefore calculated as follows:

1. Compute a minimizer u^* of the energy (2) on the convex set of functions $u : \mathbb{R}^n \rightarrow \mathbb{R}$. Details are given in section 3.2.
2. Threshold the minimizer u^* at some value $\mu \in (0, 1)$ to obtain a binary solution of the original shape optimization problem. Although these solutions generally depend on μ , all of them are guaranteed to be global minimizers of (2). In all experiments in this paper we set $\mu = 0.5$.

3.2 Numerical Implementation

A minimizer of (2) must satisfy the Euler-Lagrange equation

$$0 = f(x) - \nu \operatorname{div} \left(g(x) \frac{\nabla u(x)}{|\nabla u(x)|} \right) \quad \forall x \in \mathbb{R}^n. \tag{10}$$

Solutions to this system of equations can be obtained by a large variety of numerical solvers. We discuss some of them in the following.

Gradient Descent. The right hand side of (10) is the functional derivative of the energy (2) and gives rise to a gradient descent scheme. In practice such schemes are known to converge very slowly.

Linearized Fixed-Point Iteration. Discretization of the Euler-Lagrange equation (10) leads to a sparse nonlinear system of equations. This can be solved using a fixed point iteration scheme that transforms the nonlinear system into a sequence of linear systems. These can be efficiently solved with iterative solvers, such as Jacobi, Gauss-Seidel, Successive over-relaxation (SOR), or even multi-grid methods (also called FAS for “full approximation schemes”).

The only source of nonlinearity in (10) is the diffusivity $d := \frac{g}{|\nabla u|}$. Starting with an (arbitrary) initialization, one alternates computing the diffusivities and solving the *linear* system of equations with fixed diffusivities. We choose the SOR method as in [23].

Parallelization on Graphics Processing Unit. PDE-based approaches are generally suitable for parallel computing on graphics cards: The gradient descent and Jacobi schemes are straightforward to parallelize. This does not hold for the standard Gauss-Seidel scheme as it requires sequential processing of the image. However, in its Red-Black variant the Gauss Seidel scheme is parallelizable. The same holds for its various derivatives such as SOR and FAS.

4 Quantitative Comparison

This section constitutes the main contribution of this paper. It provides a detailed quantitative comparison of the spatially discrete and spatially continuous shape optimization schemes introduced above. While both approaches aim at minimizing the same functional, we identified three important differences:

- The spatially discrete approach has an exact termination criterion and a guaranteed polynomial running time (for a number of maximum-flow algorithms). On the other hand, the spatially continuous approach is based on the iterative minimization of a non-linear convex functional. While the required number of iterations is typically size-independent (leading to a computation time which is linear in the number of voxels), one cannot speak of a guaranteed polynomial time complexity.
- The spatially discrete approach is based on discretizing the cost functional on a lattice and minimizing the resulting submodular problem by means of graph cuts. The spatially continuous approach, on the other hand is based on minimizing the relaxed problem in a continuous setting where the resulting Euler-Lagrange equations are solved on a discrete lattice. This difference gives rise to metrication errors of the spatially discrete approach which will be discussed in Section 4.1.
- The optimization of the spatially discrete approach is based on solving a maximum flow problem, whereas the spatially continuous approach is performed by solving a partial differential equation. This fundamental difference in the underlying computational machinery leads to differences in computation time, memory consumption and parallelization properties.

4.1 Metrication Errors and Consistency

Figure 4 shows a comparison of graph cut approaches with the continuous total variation (TV) segmentation, where we show several ways to deal with the discretization of the metric for graph cuts. None of the graph cut approaches produces such a smooth curve as the TV segmentation, although the 16-connected grid gets quite close to it. This inspired us to investigate the source for the metrication errors arising in graph cut methods.

On the 4-connected grid in \mathbb{R}^2 , for example, graph cuts usually approximate the Euclidean boundary length of the interface S as

$$|S| = \int_S dS \approx \frac{1}{2} \sum_i \sum_{j \in \mathcal{N}(i)} |u_i - u_j|, \quad (11)$$

where $\mathcal{N}(i)$ denotes the four neighbors of pixel i . This implies that the boundary length is measured in an L_1 -norm rather than the L_2 -norm corresponding to the Euclidean length. The L_1 norm clearly depends on the choice of the underlying grid and is not rotationally invariant. Points of constant distance in this norm form a diamond rather than a circle (see Figure 3). This leads to a preference of boundaries along the axes (see fig. 4(a)).

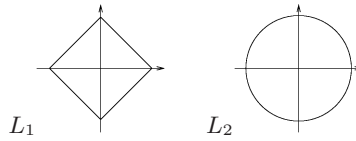


Fig. 3. 2D visualization of the L_1 -norm and the L_2 -norm for points of constant distance: Unlike the L_1 -norm, the L_2 -norm is rotationally invariant

This dependency on the underlying grid can be reduced by increasing the neighborhood connectivity. By reverting to larger and larger neighborhoods one can gradually eliminate the metrication error [3]. Increasing the connectivity leads in fact to better and better approximations of the Euclidean L_2 -norm (see fig. 4(b) and 4(c)).

Yet, a computationally efficient solution to the labeling problem requires to fix a choice of connectivity. And for any such choice, one can show that the metrication error persists, that the numerical scheme is not *consistent* in the sense that a certain residual reconstruction error (with respect to the ground truth) remains and cannot be eliminated by increasing the resolution.

Since the spatially continuous formulation is based on a representation of the boundary length by the L_2 -norm:

$$|S| = \int_S dS = \int |\nabla u| dx = \int \sqrt{u_x^2 + u_y^2} dx, \quad (12)$$

the resulting continuous numerical scheme does not exhibit such metrication errors (see fig. 4(f)). The TV segmentation performs optimization in the convex set of functions with range in $[0, 1]$. It hence allows intermediate values where the graph cut only allows binary values.

The proposed third order graph cuts discretization of the L_2 -norm (see fig. 4(d) and 4(e)) computes the same discretization of the L_2 -norm, however allowing only for binary values. Hence, in this discretized version, the Euclidean length is computed for angles of 45° and 90° to the grid, by using only a 4-connected grid. Therefore the third order L_2 -norm leads to similar results on a 4-connected grid as second order terms on an 8-connected grid.

Figure 5 shows a synthetic experiment of solving a minimal surface problem with given boundary constraints using the example of a bounded catenoid. As the true solution of this problem can be computed analytically, it is suitable for a comparison of different solvers. The experiment compares graph cuts and continuous TV minimization. It demonstrates that the 6-neighborhood graph cuts method completely fails to reconstruct the correct surface topology – in contrast to the full 26-neighborhood which approximates the Euclidean metric in a better way. However, discretization artifacts are still visible in terms of polyhedral blocky structures. Figure 5 also shows the deviation of the computed catenoid solutions from the analytic ground-truth for increasing volume resolution. It shows that for a fixed connectivity structure the computed graph cut

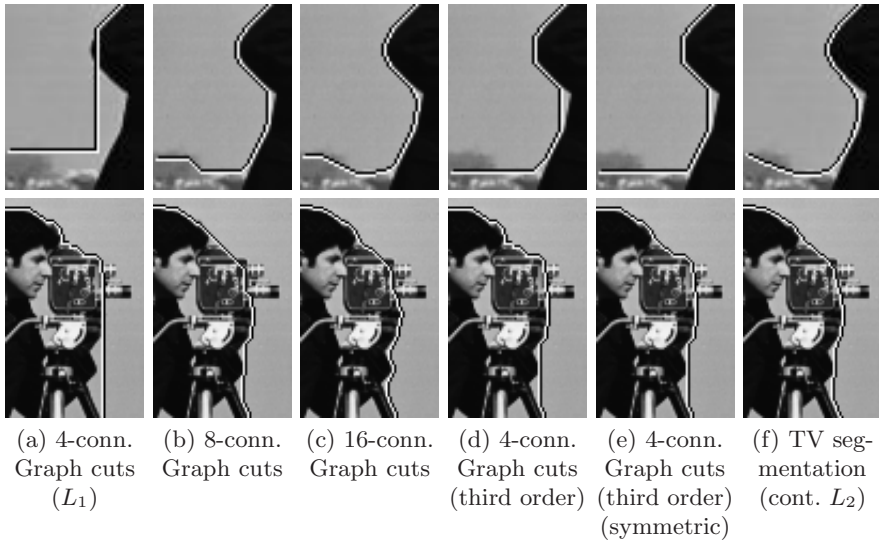


Fig. 4. Comparison of different norms and neighborhood connectivities for discrete and continuous optimization for image segmentation (Close ups of the cameraman image from figure 1). The experiment shows that a 16-connected graph is needed for the discrete solution to obtain similar results to the continuous solution.

solution is not consistent with respect to the volume resolution. In contrast, for the solution of the continuous TV minimization the discretization error decays to zero.

Figure 6 shows an experiment for real image data. In this multiview reconstruction problem the data fidelity term is dominant, therefore the discrete and the continuous solutions are similar for the same volume resolution ($108 \times 144 \times 162$). Increasing the volume resolution to $216 \times 288 \times 324$ gives more accurate results for the continuous TV formulation, while a graph cut solution for this resolution was not feasible to compute because of RAM overflow.

4.2 Computation Time

Numerous methods exist to solve either the discrete or the continuous optimization tasks. A comparison of all these methods is outside the scope of our paper. Instead we pick a few solvers we consider competitive. For all graph cut methods we use the algorithm in [4], which is arguably the most frequently used in Computer Vision. We test all discretizations mentioned above.

For the TV segmentation we implemented sequential methods on the CPU and parallel solvers on a Geforce GTX 8800 graphics card using the CUDA framework. Both implementations are based on the SOR method. On the CPU we use the usual sequential order of pixels, and on the GPU the corresponding parallelizable Red-Black scheme. A termination criterion is necessary as the number of required iterations depends on the length weight ν . We compare the

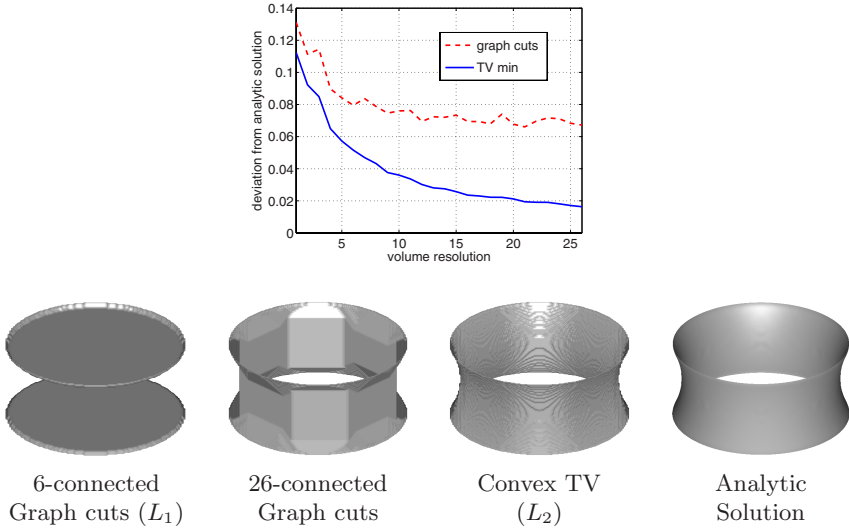


Fig. 5. Comparison of discrete and continuous optimization methods for the reconstruction of a catenoid: While the discrete graph cut algorithm exhibits prominent metrication errors (polyhedral structures), the continuous method does not show these. The plot shows the accuracy of the 26-connected graph cuts and the continuous TV method in dependence of the volume resolution. The consistency of the continuous solution is validated experimentally in the sense that the reconstruction error goes to zero with increasing resolution.

segmentations every 50 iterations and stop as soon as the maximal absolute difference drops below a value of 0.000125.

Evaluation for 2D Shape Optimization. Table 1 shows run-times for all mentioned methods. The task is image segmentation using a piecewise constant Mumford-Shah with fixed mean values 0 and 1. The main conclusions are summarized as follows:

- The TV segmentation profits significantly from parallel architectures. According to our results this is roughly a factor of 5. It should be noted that the GPU-implementation usually requires more iterations as the Red-Black order is used.
- The graph cut based methods clearly outperform the TV segmentation.
- While for the graph cut methods the 16-connected pairwise terms give generally the best results (they are largely free from grid bias), they also use up the most run-time.

Evaluation for 3D shape optimization. Table 2 shows run-times of the different optimization methods for the 3D catenoid example shown in figure 5. We detect three main conclusions:

- The 6-connected graph cuts method is the fastest, however it computes the wrong solution (see figure 5).

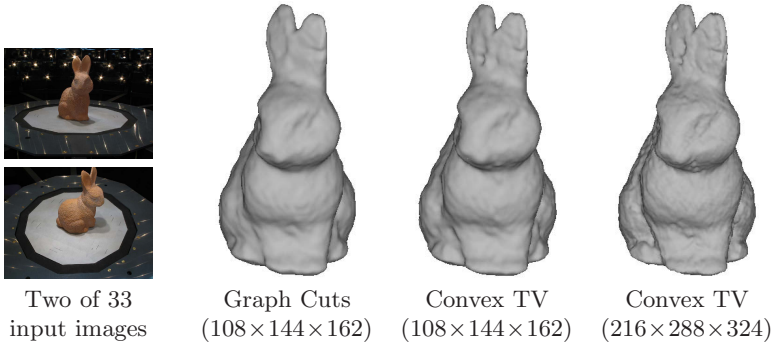


Fig. 6. Comparison of discrete and continuous optimization for multiview 3D reconstruction (presented in [23]): Due to the dominant data fidelity term, the discrete and continuous reconstructions are similar for the same volume resolution. However, for increasing resolution more accurate results can be achieved with the continuous formulation, while graph cuts rapidly come across memory limitations.

Table 1. 2D image segmentation: Run-times for the different optimization methods on two different images

Method	Cameraman Image			Berkeley Arc Image		
	$\nu = 1$	$\nu = 3$	$\nu = 5$	$\nu = 1$	$\nu = 3$	$\nu = 5$
Graph Cuts 4-connected	0.02s	0.1s	0.33s	0.06s	0.16s	0.53s
Graph Cuts 8-connected	0.05s	0.15s	0.4s	0.1s	0.27s	0.93s
Graph Cuts 16-connected	0.2s	0.35s	0.95s	0.33s	0.85s	2.7s
Graph Cuts L2 (1 quadrant)	0.03s	0.15s	0.45s	0.06s	0.19s	0.8s
Graph Cuts L2 (4 quadrants)	0.1s	0.25s	0.86	0.23s	0.53s	1.8s
TV w/ gradient descent (CPU)	111.38s	251.97s	259.87s	409.08s	636.28s	157.64s
TV w/ SOR (CPU)	10.9s	13.26s	10.2s	35.89s	103.5s	39.26s
TV w/ red-black SOR (GPU)	2s	2.7s	2s	7.6s	28.3s	8.6s

Table 2. Run-times for the 3D catenoid example

Graph cuts 6-connected	13 s
Graph cuts 26-connected	12 min 35 s
TV w/ SOR (CPU)	9 min 36 s
TV w/ red-black SOR (GPU)	30 s

- The run-time of the graph cut method changes for the worse with high connectivities, and gets slower than the TV optimization, both on CPU and GPU. Note that this limitation is due to the fact that the Boykov-Kolmogorov algorithm [4] is optimized for sparse graph structures. For denser (3D) graphs alternative push-relabel algorithms might be faster.
- The parallel implementation of the TV method allows for a speed up factor of about 20 compared to the CPU version.

4.3 Memory Consumption

With respect to the memory consumption the TV segmentation is the clear winner: It requires only one floating point value for each pixel in the image. In contrast, graph cut methods require an explicit storage of edges as well as one flow value for each edge. This difference becomes important for high resolutions, as can be seen in the experiment in figure 6.

5 Conclusion

A certain class of shape optimization functionals can be globally minimized both in a spatially discrete and in a spatially continuous setting. In this paper, we reviewed these recent developments and presented an experimental comparison of the two approaches regarding the accuracy of reconstructed shapes and computational speed. A detailed quantitative analysis confirms the following differences:

- Spatially discrete approaches generally suffer from metrication errors in the approximation of geometric quantities such as boundary length or surface area. These arise due to the binary optimization on a discrete lattice. These errors can be alleviated by reverting to larger connectivity. Alternatively, we showed that higher-order terms allow to implement an L_2 -norm of the gradient, thereby providing better spatial consistency without extending the neighborhood connectivity. As the spatially continuous formulation is not based on a discretization of the cost functional but rather a discretization of the numerical optimization (using real-valued variables), it does not exhibit metrication errors in the sense that the reconstruction errors decay to zero as the resolution is increased.
- The spatially continuous formulation allows for a straight-forward parallelization of the partial differential equation. As a consequence, one may obtain lower computation times than respective graph cut methods, in particular for the denser graph structures prevalent in 3D shape optimization.
- While the discrete graph cut optimization can be performed in guaranteed polynomial time, this is not the case for the analogous continuous shape optimization. While respective termination criteria for the convex optimization work well in practice, defining termination criteria that apply to any shape optimization problem remains an open problem.

Acknowledgments

This research was supported by the German Research Foundation, grant #CR 250/3-1.

References

1. Appleton, B., Talbot, H.: Globally minimal surfaces by continuous maximal flows. *IEEE Trans. Pattern Anal. Mach. Intell.* 28(1), 106–118 (2006)
2. Blake, A., Zisserman, A.: *Visual Reconstruction*. MIT Press, Cambridge (1987)

3. Boykov, Y., Kolmogorov, V.: Computing geodesics and minimal surfaces via graph cuts. In: IEEE Int. Conf. on Computer Vision, Nice, pp. 26–33 (2003)
4. Boykov, Y., Kolmogorov, V.: An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. IEEE Trans. on Patt. Anal. and Mach. Intell. 26(9), 1124–1137 (2004)
5. Boykov, Y., Veksler, O., Zabih, R.: Markov random fields with efficient approximations. In: Proc. IEEE Conf. on Comp. Vision Patt. Recog. (CVPR 1998), Santa Barbara, California, pp. 648–655 (1998)
6. Caselles, V., Kimmel, R., Sapiro, G.: Geodesic active contours. In: Proc. IEEE Intl. Conf. on Comp. Vis., Boston, USA, pp. 694–699 (1995)
7. Chambolle, A.: Total variation minimization and a class of binary MRF models. In: Rangarajan, A., Vemuri, B.C., Yuille, A.L. (eds.) EMMCVPR 2005. LNCS, vol. 3757, pp. 136–152. Springer, Heidelberg (2005)
8. Chan, T., Esedoğlu, S., Nikolova, M.: Algorithms for finding global minimizers of image segmentation and denoising models. SIAM Journal on Applied Mathematics 66(5), 1632–1648 (2006)
9. Delong, A., Boykov, Y.: A scalable graph-cut algorithm for n-d grids. In: Int. Conf. on Computer Vision and Pattern Recognition, Anchorage, Alaska (2008)
10. Dinic, E.A.: Algorithm for the solution of a problem of maximum flow in a network with power estimation. Soviet Mathematics Doklady 11, 1277–1280 (1970)
11. Edmonds, J., Karp, R.: Theoretical improvements in algorithmic efficiency for network flow problems. Journal of the ACM 19, 248–264 (1972)
12. Faugeras, O., Keriven, R.: Variational principles, surface evolution, PDE's, level set methods, and the stereo problem. IEEE Trans. on Image Processing 7(3), 336–344 (1998)
13. Ford, L., Fulkerson, D.: Flows in Networks. Princeton University Press, Princeton (1962)
14. Freedman, D., Drineas, P.: Energy minimization via graph cuts: settling what is possible. In: Int. Conf. on Computer Vision and Pattern Recognition, San Diego, USA, vol. 2, pp. 939–946 (June 2005)
15. Geman, S., Geman, D.: Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. IEEE Trans. on Patt. Anal. and Mach. Intell. 6(6), 721–741 (1984)
16. Goldberg, A., Tarjan, R.: A new approach to the maximum flow problem. Journal of the ACM 35(4), 921–940 (1988)
17. Goldberg, A.V., Rao, S.: Beyond the flow decomposition barrier. Journal of the ACM 45, 783–797 (1998)
18. Greig, D.M., Porteous, B.T., Seheult, A.H.: Exact maximum *a posteriori* estimation for binary images. J. Roy. Statist. Soc., Ser. B 51(2), 271–279 (1989)
19. Ising, E.: Beitrag zur Theorie des Ferromagnetismus. Zeitschrift für Physik 23, 253–258 (1925)
20. Jermyn, I.H., Ishikawa, H.: Globally optimal regions and boundaries as minimum ratio weight cycles. IEEE Trans. on Patt. Anal. and Mach. Intell. 23(10), 1075–1088 (2001)
21. Kass, M., Witkin, A., Terzopoulos, D.: Snakes: Active contour models. Int. J. of Computer Vision 1(4), 321–331 (1988)
22. Kichenassamy, S., Kumar, A., Olver, P.J., Tannenbaum, A., Yezzi, A.J.: Gradient flows and geometric active contour models. In: IEEE Int. Conf. on Computer Vision, pp. 810–815 (1995)

23. Kolev, K., Klodt, M., Brox, T., Esedoglu, S., Cremers, D.: Continuous global optimization in multiview 3d reconstruction. In: Yuille, A.L., Zhu, S.-C., Cremers, D., Wang, Y. (eds.) EMMCVPR 2007. LNCS, vol. 4679, pp. 441–452. Springer, Heidelberg (2007)
24. Kolmogorov, V., Boykov, Y.: What metrics can be approximated by Geo Cuts or global optimization of length/area and flux. In: IEEE Int. Conf. on Computer Vision, Beijing (2005)
25. Kolmogorov, V., Boykov, Y., Rother, C.: Applications of parametric maxflow in vision. In: IEEE Int. Conf. on Computer Vision, Rio de Janeiro, Brasil (2007)
26. Kolmogorov, V., Zabih, R.: What energy functions can be minimized via graph cuts?. *IEEE Trans. on Patt. Anal. and Mach. Intell.* 24(5), 657–673 (2004)
27. Mumford, D., Shah, J.: Optimal approximations by piecewise smooth functions and associated variational problems. *Comm. Pure Appl. Math.* 42, 577–685 (1989)
28. Rother, C., Kolmogorov, V., Blake, A.: GrabCut: interactive foreground extraction using iterated graph cuts. *ACM Trans. Graph* 23(3), 309–314 (2004)
29. Shi, J., Malik, J.: Normalized cuts and image segmentation. *IEEE Trans. on Patt. Anal. and Mach. Intell.* 22(8), 888–905 (2000)
30. Strang, G.: Maximal flow through a domain. *Mathematical Programming* 26(2), 123–143 (1983)
31. Vogiatzis, G., Torr, P., Cippola, R.: Multi-view stereo via volumetric graph-cuts. In: Int. Conf. on Computer Vision and Pattern Recognition, pp. 391–399 (2005)