# GeoS: Geodesic Image Segmentation

Antonio Criminisi, Toby Sharp, and Andrew Blake

Microsoft Research, Cambridge, UK

**Abstract.** This paper presents GeoS, a new algorithm for the efficient segmentation of n-dimensional image and video data.

The segmentation problem is cast as approximate energy minimization in a conditional random field. A new, parallel filtering operator built upon efficient geodesic distance computation is used to propose a set of spatially smooth, contrast-sensitive segmentation hypotheses. An economical search algorithm finds the solution with minimum energy within a sensible and highly restricted subset of all possible labellings.

Advantages include: i) computational efficiency with high segmentation accuracy; ii) the ability to estimate an approximation to the posterior over segmentations; iii) the ability to handle generally complex energy models. Comparison with max-flow indicates up to 60 times greater computational efficiency as well as greater memory efficiency.

GeoS is validated quantitatively and qualitatively by thorough comparative experiments on existing and novel ground-truth data. Numerous results on interactive *and* automatic segmentation of photographs, video and volumetric medical image data are presented.

## 1 Introduction

The problem of image and video segmentation has received tremendous attention throughout the history of computer vision with excellent recent results being achieved both interactively [1,2] and automatically [3]. However, state of the art techniques are not fast enough for real-time processing of high resolution images (e.g. $> 2Mpix$). This paper describes a new, efficient algorithm for the accurate segmentation of n-dimensional, high-resolution images and videos.

Like many vision tasks, the segmentation problem is usually cast as energy minimization in a Conditional Random Field (CRF) [1,2,4,6,7]. This encourages spatial-smoothness and contrast-sensitivity of the final segmentation. The same framework is employed here; but in contrast to graph cut-based approaches here the segmentation is obtained as the labeling corresponding to the energy minimum (MAP solution) found within a restricted, sensible subset of all possible segmentations. Such a solution will be shown to be smooth and edge aligned. The segmentation posterior over the selected subspace can also be estimated, thus enabling principled uncertainty analysis (see also [5]). Quantitative comparisons with ground truth will demonstrate segmentation accuracy equal or superior

D. Forsyth, P. Torr, and A. Zisserman (Eds.): ECCV 2008, Part I, LNCS 5302, pp. 99–112, 2008.

to that of the global minimum as found by min-cut/max-flow[1]. Restricting the search space to a small, sensible one accounts for the computational efficiency.

Similar to the work of Bai et al. in [9], we also use geodesic transforms to encourage spatial regularization and contrast-sensitivity. However, GeoS differs from [9] in a number of ways: i) The technique in [9] assumes given user strokes and imposes an implicit connectivity prior which forces each region to be connected to one such stroke[2]. In contrast, our geodesic filter acts on the energy unaries (not the user strokes). This allows GeoS to generate segmentations with no topological restrictions. ii) GeoS is not specific to *interactive* segmentation and can be applied to *automatic* segmentation as well as other tasks such as denoising, stereo and panoramic stitching. iii) GeoS presents a clear energy to be minimized. This allows quantitative comparisons with other energy-based approaches such as GrabCut [2]. Finally, iv) Despite the complexity of both algorithms being optimally linear in the number of pixels, GeoS, thanks to its contiguous memory access and parallelism is much faster than [9] in practice.

Efficient segmentation via energy minimization has also been the focus of the dual-primal technique in [10] and the logarithmic $\alpha$-expansion scheme in [11]. In spatio-temporal MRFs efficiency may be gained by either reusing the graph flow or the search trees [12,13]. Instead, the efficiency of GeoS stems from its optimized memory access and its ability to exploit the power of modern multi-core architectures. In contrast, graph-cut does not lend itself to easy parallelization.

Finally, unlike graph-cut, our approximate minimization algorithm is not restricted to a specific family of energies. This enables us to experiment with more sophisticated models, like those containing *global* constraints.

## 2 Background on Distance Transforms

This section presents background on geodesic distances and related algorithms.

**Unsigned geodesic distance.** Given an image $I$ defined on a 2D domain $\Psi$, a binary mask $M$ (with $M(\mathbf{x}) \in \{0,1\} \forall \mathbf{x}$) and an "object" region $\Omega$ with $\mathbf{x} \in \Omega \iff M(\mathbf{x}) = 0$, the unsigned geodesic distance of each pixel $\mathbf{x}$ from $\Omega$ is defined as:

$$D(\mathbf{x}; M, \nabla I) = \min_{\{\mathbf{x}' | M(\mathbf{x}')=0\}} d(\mathbf{x}, \mathbf{x}'), \quad \text{with} \tag{1}$$

$$d(\mathbf{a}, \mathbf{b}) = \min_{\boldsymbol{\Gamma} \in \mathcal{P}_{\mathbf{a},\mathbf{b}}} \int_0^1 \sqrt{||\boldsymbol{\Gamma}'(s)||^2 + \gamma^2 (\nabla I \cdot \mathbf{u})^2} \; ds \tag{2}$$

with $\mathcal{P}_{\mathbf{a},\mathbf{b}}$ the set of all paths between the points $\mathbf{a}$ and $\mathbf{b}$; and $\boldsymbol{\Gamma}(s) : \Re \rightarrow \Re^2$ indicating one such path, parametrized by $s \in [0,1]$. The spatial derivative $\boldsymbol{\Gamma}'(s)$

[1] The fact that a local energy minimum may be more accurate than the global one should not come as a surprise. In fact [8] have discussed the limitations of the widely used unary + pairwise energies and the need for more realistic models.

[2] E.g. in [9] segmenting the image of a chess board into its black and white squares would require $8 \times 8 = 64$ user strokes.

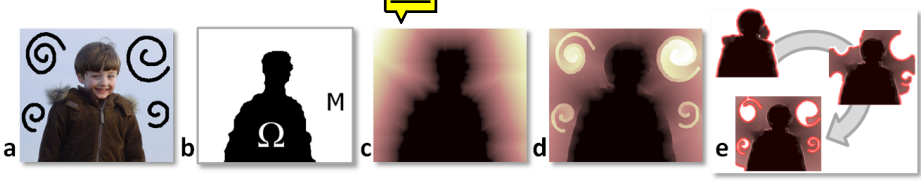



**Fig. 1.** $\mathcal{O}(N)$ **geodesic distance transform algorithms. (a)** Original image, $I$; **(b)** Input mask $M$ with "object" $\Omega$. **(c)** Distance $D(\mathbf{x}; M, \nabla I)$ from $\Omega$ (with $\gamma = 0$ in (1)); **(d)** *Geodesic* distance from object ($\gamma > 0$) computed with the raster scan algorithm in [18] (two complete raster-scan passes suffice). Note the large jump in the distance $D$ in correspondence with strong edges. **(e)** Different stages of front propagation of the algorithm in [19], eventually leading to a geodesic distance similar to the one in (d).

is $\boldsymbol{\Gamma}'(s) = \partial\boldsymbol{\Gamma}(s)/\partial s$. Also, the unit vector $\mathbf{u} = \boldsymbol{\Gamma}'(s)/||\boldsymbol{\Gamma}'(s)||$ is tangent to the direction of the path. The factor $\gamma$ weighs the contribution of the image gradient versus the spatial distances. Equation (1) generalizes the conventional Euclidean distance; in fact, $D$ reduces to the Euclidean path length for $\gamma = 0$.

**Distance transform algorithms.** Excellent surveys of techniques for computing *non-geodesic* distance transforms may be found in [14,15]. There, two main kinds of algorithms are described: *raster-scan* and *wave-front propagation*. Raster-scan algorithms are based on kernel operations applied sequentially over the image in multiple passes [16]. Instead, wave-front algorithms such as Fast Marching Methods (FMM) [17] are based on the iterative propagation of a pixel front with velocity $F$.

Geodesic versions of both kinds of algorithms may be found in [18] and [19], respectively. An illustration is shown in fig. 1. Both the Toivanen and Yatziv algorithms produce approximations to the actual distance and both have optimal complexity $\mathcal{O}(N)$ (with $N$ the number of pixels). However, this does not mean that they are equally fast in practice. In fact, FMM requires accessing image locations far from each other in memory. Thus, the limited memory bandwidth of modern computers limits the speed of execution of such algorithms much more than their modest computational burden. In contrast, Toivanen's technique (employed here) accesses the image memory in *contiguous* blocks, thus minimizing such delays. This yields speed up factors of at least one order of magnitude compared to [19]. Algorithmic details are presented in the Appendix.

## 3 Geodesic, Symmetric Morphology

This section introduces a new filtering operator which constitutes the basis of our segmentation process. The filter builds upon efficient distance transforms.

**Geodesic morphology.** The two most basic morphological operations – erosion and dilation – are usually defined in terms of binary structured elements acting on binary images. However, it is possible to redefine those operations as functions of real-valued image distances, as follows. Equation (1) leads to the following definition of the *signed* geodesic distance from the object *boundary*:

$$D_s(\mathbf{x}; M, \nabla I) = D(\mathbf{x}; M, \nabla I) - D(\mathbf{x}; \overline{M}, \nabla I), \tag{3}$$

with $\overline{M} = 1 - M$. It follows that dilation and erosion may be obtained as

$$M_d(\mathbf{x}) = [D_s(\mathbf{x}; M, \nabla I) > \theta_d], \quad M_e(\mathbf{x}) = [D_s(\mathbf{x}; M, \nabla I) > -\theta_e] \tag{4}$$

with $\theta > 0$ the diameter of the disk-shaped structured element. The indicator function [.] returns 1 if the argument is true and 0 otherwise. More useful, *idempotent* filters (an operator $f$ is idempotent iff. $f(f(x)) = f(x)$) such as closing and opening are achieved as:

$$M_c(\mathbf{x}) = [D(\mathbf{x}; \overline{M_d}, \nabla I) > -\theta_e], \quad M_o(\mathbf{x}) = [D(\mathbf{x}; M_e, \nabla I) > \theta_d] \tag{5}$$

respectively. Redefining known morphological operators in terms of real-valued distances allows us to: i) implement those operators very efficiently, and ii) introduce contrast sensitivity effortlessly, by means of geodesic processing. Next, a further modification to conventional morphology is introduced.

**Adding symmetry.** Closing and opening are asymmetrical operations in the sense that the final result depends on the order in which the two component operations are applied to the input mask (see fig. 2g,h). However, in image filtering one would just wish to define the dimension of the regions to be removed (e.g. noise speckles) and apply the filter without worrying about the sequentiality of operations within the filter. Here we solve this problem by defining the following new, symmetrical filter:

$$M_s(\mathbf{x}; M, I) = [D_s^s(\mathbf{x}; M, \nabla I) > 0] \tag{6}$$

where the symmetric, signed distance $D_s^s$ is defined as:

$$D_s^s(\mathbf{x}; M, \nabla I) = D(\mathbf{x}; M_e, \nabla I) - D(\mathbf{x}; \overline{M_d}, \nabla I) + \theta_d - \theta_e, \tag{7}$$

with $M_e$ and $\overline{M_d}$ defined earlier. The additional term $\theta_d - \theta_e$ enforces the useful idempotence property; *i.e.* it keeps unaltered the remaining signal structure. Formulating morphological operations in terms of real-valued distances allows us to perform symmetrical mixing of closing and opening via (7). The only two geometric parameters $\theta_d, \theta_e$ are very intuitive as they correspond to the maximum size of the foreground and background noise speckles to be removed.

In summary, the operator (6) generalizes existing morphological operations by adding symmetry and edge-awareness. In fact, setting $\gamma = 0$ and then $\theta_d = 0$ ($\theta_e = 0$) reproduces conventional closing (opening). Figure 2 illustrates the filtering process for 1D and 2D toy examples. Isolated peaks and valleys are simultaneously removed while maintaining unaltered the remaining signal. Equipped with this new tool we can now focus on the segmentation problem.

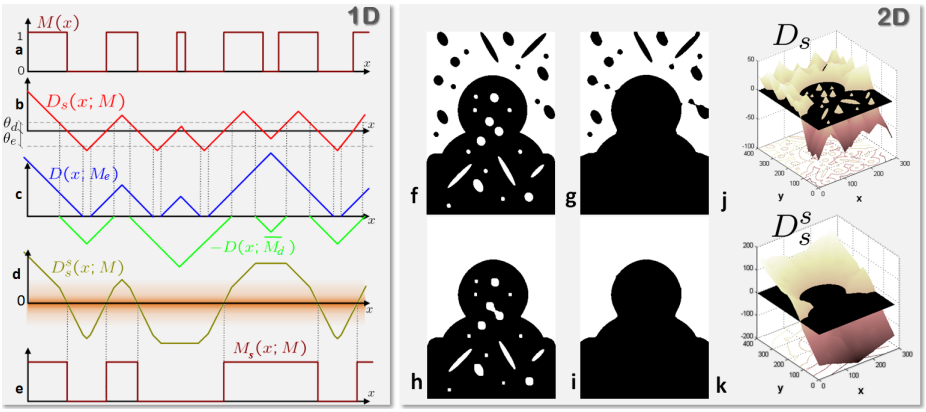



**Fig. 2. Symmetric filtering in 1D and 2D. (a)** Input, binary 1D signal $M$. **(b)** The initial signed distance $D_s$. **(c)** The two further unsigned distances for selected values of $\theta_d$, $\theta_e$. **(d)** The final signed distance $D_s^s$. **(e)** The filtered mask $M_s(x;M)$. Some of the peaks and valleys of $M(x)$ have been removed while maintaining the integrity of the remaining signal. For simplicity of explanation here no image gradient is used. Now let's look at a 2D example. **(f)** Original 2D mask $M$, **(g)** mask after closing, **(h)** after opening, **(i)** resulting mask $M_s$ after our symmetric filtering. **(j)** The distance $D_s(\mathbf{x})$ for the input 2D mask in (f). **(k)** The final distance $D_s^s(\mathbf{x})$ for $\theta^d = 10$ and $\theta^e = 11$. The intersection of $D_s^s(\mathbf{x})$ with the xy plane through 0 results in the filtered mask $M_s$ shown in (i). The parameters $\theta_d$ and $\theta_e$ are fixed in all (f,...,i).

## 4 Segmentation Via Restricted Energy Minimization

The binary segmentation problem addressed here is cast as minimizing an energy of type

$$E(\mathbf{z}, \boldsymbol{\alpha}) = U(\mathbf{z}, \boldsymbol{\alpha}) + \lambda V(\mathbf{z}, \boldsymbol{\alpha}) \tag{8}$$

with $\mathbf{z}$ the image data and $\boldsymbol{\alpha}$ the per-pixel labeling, with $\alpha_n \in \{\mathtt{Fg}, \mathtt{Bg}\}$. The subscript $n$ indexes the pixels and `Fg` (`Bg`) indicates foreground (background). The unary potential $U$ is defined as the sum of pixel-wise likelihoods of the form $U(\mathbf{z}, \boldsymbol{\alpha}) = -\sum_n \log p(z_n|\alpha_n)$; and the data-dependent pairwise term is $V(\mathbf{z}, \boldsymbol{\alpha}) = -\sum_{m,n\in\mathcal{N}} [\alpha_n \neq \alpha_m] \exp(-|z_n - z_m|/\eta)$. Here we use 8-neighborhood cliques $\mathcal{N}$. Flux may also be incorporated in (8) as a further unary term.

Sub-modular energies of the form (8) can be minimized exactly by min-cut. However, in image segmentation, finding the global minimum of such energy makes sense only provided that the energy model correctly captures the statistics of natural images. Recent work has shown that this is often *not* the case [8]. It has been observed that often local energy minima correspond to segmentations which are more accurate (compared to ground truth) than that yielded by the global minimum. Thus, a technique that can find good local minima efficiently becomes valuable. This section describes such an approximate and efficient technique. Later we will also show how such algorithm can be applied to energy models of a more general nature than the one in (8).

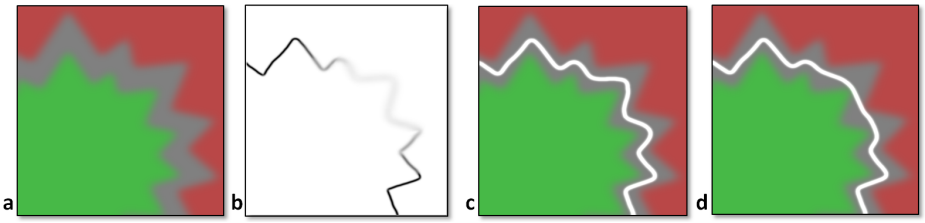



**Fig. 3. Filter behaviour in the presence of weak unaries. (a)** Input unaries (green for `Fg` and red for `Bg`), with large uncertain areas (in grey). **(b)** Magnitude of gradient of input image. **(c)** Computed segmentation boundary (white curve) for a *small* value of $\theta_d = \theta_e$. **(d)** As in (c) but for *large* $\theta$. Larger values of $\theta$ yield smoother segmentation boundaries in the presence of weak edges and/or weak unaries. In contrast, strong gradients "lock" the segmentation in place.

Key to our algorithm is the minimization of the energy in (8) by efficient search of the solution $\boldsymbol{\alpha}^*$ over a restricted, parametrized 2D manifold of all possible segmentations. Let us define $\boldsymbol{\theta} = (\theta_d, \theta_e) \in \mathcal{S}$, with $\mathcal{S} \subset \mathbb{R}^2$. As described earlier, given a value of $\boldsymbol{\theta}$ the geodesic operator (6) has the property of removing isolated regions (with dimensions $< \boldsymbol{\theta}$) from foreground and background in binary images. Therefore, if we can adapt our filter to work on real-valued unaries, then for different values of $\boldsymbol{\theta}$ different levels of spatial smoothness would be obtained and thus different energy values. The segmentation we are after is

$$\boldsymbol{\alpha}^* = \boldsymbol{\alpha}(\boldsymbol{\theta}^*), \qquad \text{with} \qquad \boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta} \in \mathcal{S}} E(\mathbf{z}, \boldsymbol{\alpha}(\boldsymbol{\theta})).$$

Next we focus on the details of the GeoS algorithm.

**Segmentation proposals.** In a binary segmentation problem, given the *real*-valued log likelihood ratio: $L(\mathbf{x}) = \log p(z_{n(\mathbf{x})}|\alpha_{n(\mathbf{x})} = \texttt{Fg}) - \log p(z_{n(\mathbf{x})}|\alpha_{n(\mathbf{x})} = \texttt{Bg})$ we redefine the mask $M(\mathbf{x}) \in [0, 1]$ as a log-odds map $M(\mathbf{x}) = \sigma(L(\mathbf{x}))$ with $\sigma(.)$ the sigmoid transformation $\sigma(L) = 1/(1 + \exp(-L/\mu))$ [3]. The distance (1) then becomes:

$$D(\mathbf{x}; M, \nabla I) = \min_{\mathbf{x}' \in \Psi} (d(\mathbf{x}, \mathbf{x}') + \nu M(\mathbf{x}')) \tag{9}$$

with $d(.)$ as in (2). $\nu$ (trained discriminatively) establishes the mapping between the unary beliefs and the spatial distances. Different segmentations are achieved for different values of $\boldsymbol{\theta}$ via (6). Please refer to [20] for related work on (*non geodesic*) generalized distance transforms.

Figure 3 illustrates the effect of applying our filter (6) to *weak*, real-valued unaries. Larger values of $\theta$ not only tend to remove isolated islands (as illustrated earlier) but also produce smoother segmentation boundaries, in the presence of weak contrast and/or uncertain unaries. Furthermore, strong edges "lock" the segmentation in place. In summary, our filter produces segmentations which are smooth, edge-aligned and agree with the unaries. Thus the filter is ideally suited to be used for the generation of plausible segmentation hypotheses.

[3] In all experiments in this paper the value of $\mu$ is fixed to $\mu = 5$.

**Energy minimization.** We now search for the value $\boldsymbol{\theta}^*$ corresponding to the lowest energy $E_{GeoS} = E(\mathbf{z}, \boldsymbol{\alpha}(\boldsymbol{\theta}^*))$. For each value of $\boldsymbol{\theta}$ the segmentation operation in (6) requires 4 unsigned distance transforms. Thus, a naïve exhaustive search for $N_d \times N_e$ values of $\boldsymbol{\theta}$ would require 4 $N_d$ $N_e$ distance computations. However, it is easy to show that by pre-computing distances the load is reduced to only $2 + N_d + N_e$ operations [4], with an associated memory overhead. All of the above distance transforms are independent of each other and can be computed *in parallel* on appropriate hardware. Therefore, in a machine with $N_c$ processors (cores) the total time $T$ taken to run exhaustive search is $T = (2 + (N_d + N_e)/N_c)\, t$, with $t$ the unit time required for each unsigned distance transform (9). An economical gradient descent optimization strategy may also be employed here. Comparative efficiency results are presented in section 5.

**Selecting the search space.** An important question at this point is how to choose the search space $\mathcal{S}$. As discussed earlier, $\boldsymbol{\theta}$ are intuitive parameters which represent the maximum size of the regions to be removed. Therefore, $\mathcal{S}$ must depend on the image resolution and on the spatial extent of noisy regions within the unary signal. Unless otherwise stated, for the approximately VGA-sized images used in this paper we have fixed $\mathcal{S} = \{5, 6, ..., 15\} \times \{5, 6, ..., 15\}$ (and thus $N_d = N_e = 10$).

**Estimating the segmentation posterior.** Computing the full CRF posterior $p(\boldsymbol{\alpha}) = 1/Z_p \ \exp\left(-E(\boldsymbol{\alpha})/\sigma_p\right)$ is impractical [5]. However, importance sampling [21] allows us to approximate $p(\boldsymbol{\alpha})$ with its Monte Carlo mean $\tilde{p}(\boldsymbol{\alpha})$. The *proposal distribution* $q(\boldsymbol{\alpha})$ can be computed as $q(\boldsymbol{\alpha}) = 1/Z_q \ \exp\left(-E(\boldsymbol{\alpha}(\boldsymbol{\theta}))/\sigma_q\right)$, $\forall \boldsymbol{\theta} \in \mathcal{S}$ (and $q(\boldsymbol{\alpha}) = 0 \ \forall \boldsymbol{\theta} \notin \mathcal{S}$). Then $\tilde{p}_N^q(\boldsymbol{\alpha}) = 1/n \sum_{i=1}^{n} p(\boldsymbol{\alpha}(\boldsymbol{\Theta}_i))/q(\boldsymbol{\alpha}(\boldsymbol{\Theta}_i))$, with the $N$ samples $\boldsymbol{\Theta}_i$ generated from a uniform prior over $\mathcal{S}$. Since $\mathcal{S}$ is a small, quantized 2D space, in practice $\boldsymbol{\Theta}_i$ are generated deterministically by exploring the entire $\mathcal{S}$. The parameters $\sigma_q, \sigma_p$ are trained discriminatively from hundreds of manually-labelled trimaps (e.g. fig. 4d). The estimated CRF posterior $\tilde{p}_N^q(\boldsymbol{\alpha})$ is used in fig. 4c"',d"' to compute the segmentation mean $\tilde{\boldsymbol{\alpha}} = \int_{\boldsymbol{\alpha}} \boldsymbol{\alpha} \tilde{p}_N^q(\boldsymbol{\alpha}) d\boldsymbol{\alpha}$ and the associated variance $\Lambda_{\boldsymbol{\alpha}}$. In interactive video segmentation, the quantity $\Lambda_{\boldsymbol{\alpha}}$ may for instance be used to detect unstable segmentations and ask the user to improve the appearance models by adding more strokes. Proposals sampled from $\mathcal{S}$ may also be fused together via QPBO [22].

**Exploring more complex energy models.** In contrast to graph-cut, here the energy and its minimization algorithm are decoupled. This fact is advantageous since now the choice of class of energies is no longer dominated by considerations of tractability. Our technique can thus be applied to more complex energy models than the one in (8). As an example, below we consider energies containing global terms:

$$E(\mathbf{z}, \boldsymbol{\alpha}) = U(\mathbf{z}, \boldsymbol{\alpha}) + \lambda V(\mathbf{z}, \boldsymbol{\alpha}) + \kappa G(\mathbf{z}, \boldsymbol{\alpha}) \tag{10}$$

The global soft constraint $G$ cannot be written as a sum of unary and pairwise terms [23,24]. $G$ captures global properties of image regions and can be used,

[4] The distance $D_s$ need be computed only once per image as it does not depend on $\boldsymbol{\theta}$.

e.g. to encourage constraints on areas, global appearance, shape or context. For example in [23] $G = G(h_1, h_2)$ is defined as a divergence between region histograms $h_i$. General energy models of this kind have not been used much in the literature because of the lack of appropriate optimization techniques [25]. However, their usefulness is clear, and finding even approximate, efficient solutions is important. Results of this kind are presented in the next section.

## 5 Results and Applications

This section validates GeoS with respect to accuracy and efficiency. Qualitative and quantitative results on interactive and automatic image and video segmentation are presented.

**Interactive image segmentation.** Figure 4 shows a first example of interactive segmentation on a difficult standard test image showing camouflage [26]. The energy is defined as in (8). In this and all interactive segmentation examples, the unaries (fig. 4c) are obtained by: i) computing histograms over the RGB space quantized into $32^3$ bins from the user provided strokes, and ii) evaluating the `Fg` and `Bg` likelihoods on all image pixels. As expected the GeoS MAP segmentation in fig. 4c' looks like a version of the unaries but with higher spatial smoothness of labels. The GeoS solution is very similar to the min-cut one (fig. 4c"). The segmentation mean $\tilde{\boldsymbol{\alpha}}$ and variance are also computed. The mean image $\tilde{\boldsymbol{\alpha}}$ can be thought of as an automatically computed trimap.

**Computational efficiency.** Here we compare the run times of GeoS and min-cut. For min-cut we use the public implementation in [28] and also our own implementation which has been optimized for grid graphs. GeoS has been implemented using SSE2 assembly instructions, exploiting cache efficiency and multi-threading for optimal performance. The data-level parallelism (SSE2) is made possible by noting that four of the five terms in the equation in fig. 12 are independent of the current scan-line. All experiments are run on an Intel Core2 Duo desktop with 3GB RAM and $2 \times 2.6$GHz CPU cores.

Figure 5 plots the run time curves obtained when segmenting the "llama" image as a function of image size. Both min-cut curves show a slightly "superlinear" behavior, while GeoS is linear. On a $1600 \times 1200$ image GeoS ($N_c = 4, N_d = N_e = 10$) produces a 12-fold speed-up with respect to min-cut. On-line video segmentation may be achieved by gradient descent because of the high temporal correlation of the energy in consecutive frames (*cf.* fig.5c, denoted "g.d."). Using 2 steps of gradient descent on $2 \times 2$ grids (typical values) produces a 21-fold speed-up. Geos' efficiency gain increases non-linearly for larger resolutions. For instance, on a $25Mpix$ image the GeoS ($N_c = 4, N_d = N_e = 10$) produces a 33-fold speed-up and gradient-descent GeoS a **60**-fold speed-up with respect to min-cut. Finally, while min-cut's run times depend on the quality of the unaries (the more uncertain, the slower the minimization) GeoS has a fixed running cost, thus making its behaviour more predictable.

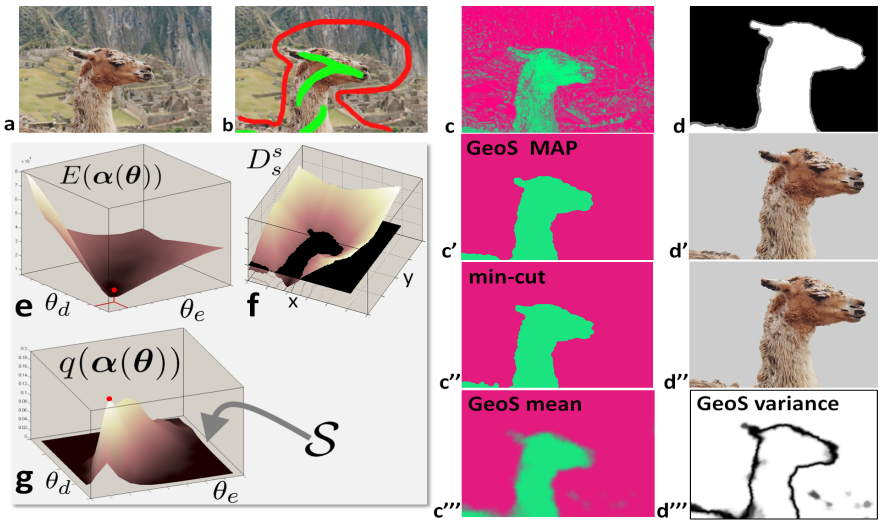



**Fig. 4. GeoS v min-cut for interactive segmentation. (a)** Input image.**(b)** user provided `Fg` and `Bg` strokes, **(c)** corresponding unaries (green for `Fg`, red for `Bg` and grey for uncertain). **(d)** ground truth segmentation (zoomed). **(e)** energy $E(\boldsymbol{\alpha}(\boldsymbol{\theta}))$, with the computed minimum marked in red. **(f)** The distance $D_s^s$ corresponding to the optimum $\boldsymbol{\theta}^*$. **(g)** The proposal distribution $q(\boldsymbol{\alpha}(\boldsymbol{\theta}))$. **(c',d')** Resulting *GeoS MAP* segmentation $\boldsymbol{\alpha}^*$ and corresponding `Fg` layer. **(c",d")** *Min-cut* segmentation on the same energy. **(c"')** *GeoS mean* segmentation $\tilde{\boldsymbol{\alpha}}$, see text. Uncertain pixels are shown in grey. **(d"')** corresponding *GeoS variance* (dark for high variance).

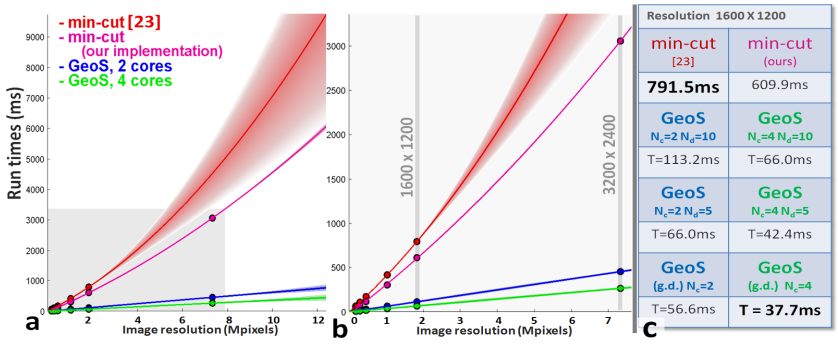
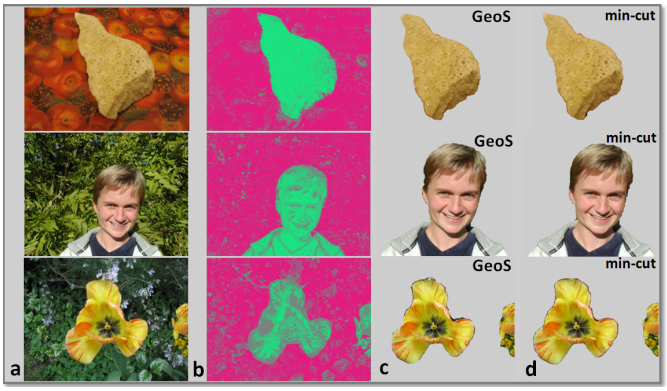



**Fig. 5. Run time comparisons. (a)** Run times for min-cut and GeoS for varying image size. Circles indicate our measurements. Associated uncertainties have been estimated by assuming Gaussian noise on the measurements [27]. Min-cut [28] fails to run on images larger than $1600 \times 1200$, thus yielding larger uncertainty for higher resolutions. **(b)** as in (a), zoomed into the highlighted region. Min-cut shows a slightly superlinear behaviour while GeoS is linear with a small slope. For large resolutions GeoS can be up to 60 times faster than min-cut. **(c)** Run-times for $1600 \times 1200$ resolution. Even for relatively low resolution images GeoS is considerably faster than min-cut. Identical energies are used for all four algorithms compared in this figure.




| % | GeoS v min-cut | GeoS v gr.-truth | Min-cut v gr.-truth |
|---|---|---|---|
| sponge | 0.22 | 0.21 | 0.18 |
| person | 0.23 | 0.51 | 0.44 |
| flower | 0.71 | 0.54 | 0.37 |
| llama | 0.87 | 0.62 | 0.32 |

e

**Fig. 6. Interactive image segmentation. (a)** Original test images: "sponge", "person" and "flower" from the standard test database in [8] (approx. VGA sized); **(b)** unaries computed from the user scribbles provided in [8]; **(c)** *GeoS* segmentations. **(d)** *min-cut* segmentations for the same energy as in (c), corresponding to a single iteration of GrabCut [26]. More iterations as in [26] help reduce the amount of manual interaction. **(e)** Percentage of differently classified pixels, see text.

When comparing GeoS with the algorithm in [29], GeoS yielded a roughly 30-fold speed-up factor while avoiding connectivity issues. Besides, the algorithm in [9,29] is designed for *interactive* segmentation only.

**Segmentation accuracy.** Figure 6 presents segmentation results on the standard test images used in [8]. To quantify the difference in segmentation quality between min-cut and GeoS we could use the relative difference between the minimum energy found, i.e. $\delta(E_{GeoS}, E_{min}) = (E_{GeoS} - E_{min})/E_{min}$, as in [8]. However, this is not a good measure since adding a constant term $\Delta E$ to the energy would not change the output segmentation while it would affect $\delta$. Thus $\delta$ can be made very small by choosing a very large $\Delta$. Here we chose to compare the GeoS and min-cut segmentations to each other and to the manually labelled ground truth by counting the number of differently classified pixels.

Results for the four example images encountered so far are reported in fig. 6e. In each case the optimum value of $\lambda$ (learned discriminatively for min-cut) was used. The min-cut and GeoS results are very close visually and quantitatively, with the number of differently labelled pixels well below 1% of the image area. The largest difference is for the "llama" image where the furry outline makes both solutions equally likely. All segmentations are also very close to the ground truth. The three GeoS segmentations in fig. 6c were obtained in $< 10ms$ each (to be compared with the much larger timings reported in [8]).

**Contrast sensitivity.** In fig. 7 contrast-sensitivity enables thin protrusions to be segmented correctly, despite the absence of flux in the energy. Contrast is especially important with weaker unaries such as shown in fig. 3 and fig. 8. In fig. 8b,b', using patches to compute stereo likelihoods [3] causes their misalignment with respect to the foreground boundary. Using $\gamma > 0$ in GeoS encourages

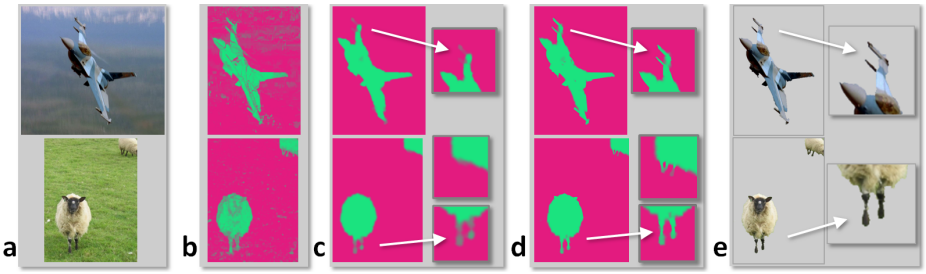



**Fig. 7. The effect of contrast on thin structures. (a)** Two input images. **(b)** unaries (from user strokes); **(c)** GeoS mean segmentation with no contrast. The smoothness prior makes thin protrusions (e.g. the sheep legs or the planes missiles) uncertain (grey). **(d)** GeoS mean segmentation with contrast enabled. Now the contrast-sensitive pairwise term correctly pulls the aeroplane and sheep thin protrusions in the foreground. **(e)** GeoS MAP segmentation for the contrast-sensitive energy in (d).

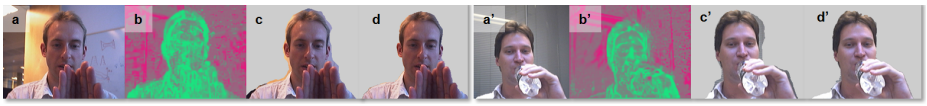



**Fig. 8. Segmentation results in the presence of weak, stereo unaries. (a,a')** Frames from two stereo videos. **(b,b')** Stereo likelihoods, with large uncertain areas (in grey). **(c,c')** GeoS segmentation, with no contrast sensitivity. **(d,d')** As in (c,c') but with contrast sensitivity. Now the segmentation accurately follows the person's outline.


**Fig. 9. Exploiting global constraints. (a)** Original test image; **(b)** user provided `Fg` and `Bg` strokes; **(c)** corresponding unaries; **(d)** min-cut segmentation with $\kappa = 0$; The circular traffic sign is missed out. **(e)** GeoS segmentation on same energy as in (d); **(f)** GeoS segmentation on energy with global constraint $G = |Area_{\mathtt{Fg}}/Area - 0.7|$.

the segmentation to follow the person's silhouette correctly (fig. 8d,d'). Next we experiment with more complex energies, containing global terms.

**Exploiting global energy constraints.** The example in fig. 9 shows the effect of the global constraint $G$ in (10). Energies of the kind in (10) cannot be minimized by min-cut. In the segmentations in fig. 9d,e (where $\kappa = 0$) the circular weight limit sign is missed. This problem is corrected in fig. 9f which uses the energy (10) (with $k > 0$). The additional global term $G$ is defined

as $G = |Area_{\texttt{Fg}}/Area - 0.7|$ to encourage the Fg region to cover about 70% of the image area. Similar results are obtained on this image by imposing soft constraints on global statistics of appearance or shape (see also [30]).

**Segmenting n-dimensional data.** Geodesic transforms and thus GeoS can easily be extended to more than 2 dimensions. Figure 10 shows an example of segmentation of the space-time volume defined by a time-lapse video of a growing flower. Figure 11 shows segmentation of 3D MRI data. In each case brush strokes applied in two frames suffice to define good unaries. Individual organs are highlighted in fig. 11 by repeated segmentation (see also [31]).


**Fig. 10. Batch, space-time segmentation of video. (a)** Three frames of a time-lapse video of a growing flower. **(b, c, d)** Three views of the segmented "video-cube". GeoS segmentation is achieved directly in the space-time volume of the video.

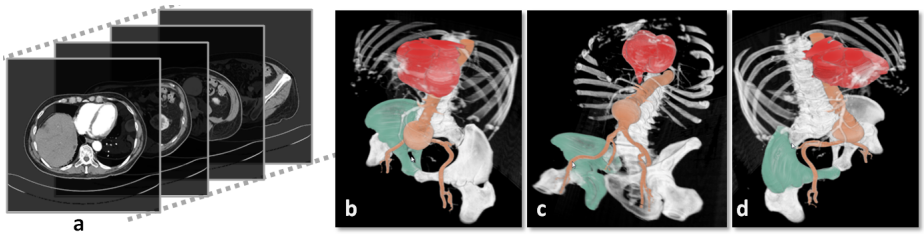



**Fig. 11. Segmentation of 3D, medical data. (a)** Some of the 294 $512 \times 512$ input grey-scale slices from a patient's torso. **(b,c,d)** GeoS segmentation results. Bones, heart and aorta have been accurately separated from the remaining soft tissue, directly in the 3D volume. Different organs have been coloured to aid visual inspection.

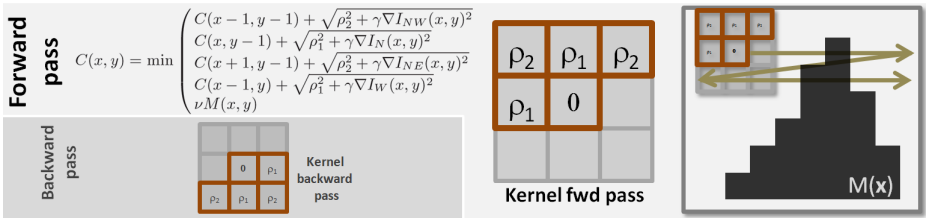



**Fig. 12. Efficient geodesic distance transform.** See appendix.

## 6 Conclusion

This paper has presented GeoS, a new algorithm for the efficient segmentation of n-D images and videos. The key contribution is an approximate energy minimization technique which finds the segmentation solution by economical search within a restricted space. Such space is populated by good, spatially-smooth, contrast-sensitive solution hypotheses generated by our new, efficient geodesic operator. The algorithm's reduced search space, contiguous memory access and intrinsic parallelism account for its efficiency even for high resolution data.

Extensive comparisons between GeoS and min-cut show comparable accuracy; with GeoS running many times faster and being able to handle more general energies.

## References

1. Boykov, J., Jolly, M.P.: Interactive graph cuts for optimal boundary and region segmentation of objects in n-D images. In: IEEE ICCV (2001)
2. Rother, C., Kolmogorov, V., Blake, A.: Grabcut: Interactive foreground extraction using iterated graph cuts. In: ACM Trans. on Graphics (SIGGRAPH) (2004)
3. Kolmogorov, V., Criminisi, A., Blake, A., Cross, G., Rother, C.: Bilayer segmentation of binocular stereo video. In: IEEE CVPR (2005)
4. Criminisi, A., Cross, G., Blake, A., Kolmogorov, V.: Bilayer segmentation of live video. In: IEEE CVPR (2006)
5. Kohli, P., Torr, P.H.S.: Measuring uncertainty in Graph Cut solutions. In: ECCV (2006)
6. Kolmogorov, V., Zabih, R.: Multi-camera scene reconstruction via graph cuts. In: ECCV (2002)
7. Sinop, A.K., Grady, L.: A seeded image segmentation framework unifying graph cuts and random walker which yields a new algorithm. In: IEEE ICCV (2007)
8. Szeliski, R., Zabih, R., Scharstein, D., Veksler, O., Kolmogorov, V., Agarwala, A., Tappen, M., Rother, C.: A comparative study of energy minimization methods for markov random fields. In: ECCV (2006)
9. Bai, X., Sapiro, G.: A geodesic framework for fast interactive image and video segmentation and matting. In: IEEE ICCV (2007)
10. Komodakis, N., Tziritas, G., Paragios, N.: Fast, approximately optimal solutions for single and dynamic MRFs. In: IEEE CVPR (2007)
11. Lempitsky, V., Rother, C., Blake, A.: Logcut - efficient graph cut optimization for markov random fields. In: IEEE ICCV, Rio (2007)
12. Juan, O., Boykov, J.: Active graph cuts. In: IEEE CVPR (2006)
13. Kohli, P., Torr, P.: Dynamic graph cuts for efficient inference in markov random fields. PAMI (2007)
14. Fabbri, R., Costa, L., Torrelli, J., Bruno, O.: 2d euclidean distance transform algorithms: A comparative survey. ACM Computing Surveys 40 (2008)
15. Jones, M., Baerentzen, J., Sramek, M.: 3d distance fields: a survey of techniques and applications. IEEE Trans. on Visualization and Computer Graphics 12 (2006)
16. Borgefors, G.: Distance transformations in digital images. Computer Vision, Graphics and Image Processing (1986)
17. Sethian, J.A.: Fast marching methods. SIAM Rev. 41 (1999)

18. Toivanen, P.J.: New geodesic distance transforms for gray-scale images. Pattern Recognition Letters 17, 437–450 (1996)
19. Yatziv, L., Bartesaghi, A., Sapiro, G.: O(n) implementation of the fast marching algorithm. Journal of Computational Physics 212, 393–399 (2006)
20. Felzenszwalb, P.F., Huttenlocher, D.P.: Pictorial structures for object recognition. IJCV 61 (2005)
21. Ripley, B.D.: Stochastic Simulation. Wiley and Sons, Chichester (1987)
22. Rother, C., Kolmogorov, V., Lempitsky, V.T., Szummer, M.: Optimizing binary MRFs via extended roof duality. In: IEEE CVPR (2007)
23. Rother, C., Kolmogorov, V., Minka, T., Blake, A.: Cosegmentation of image pairs by histogram matching - incor. a global constraint into MRFs. In: CVPR (2006)
24. Kolmogorov, V., Boykov, J., Rother, C.: Applications of parametric maxflow in computer vision. In: IEEE ICCV, Rio (2007)
25. Kolmogorov, V., Zabih, R.: What energy functions can be minimized via graph cuts? IEEE Trans. PAMI 26 (2004)
26. Blake, A., Rother, C., Brown, M., Perez, P., Torr, P.: Interactive image segmentation using an adaptive GMMRF model. In: ECCV (2004)
27. Bishop, C.M.: Pattern Recognition and machine Learning. Springer, Heidelberg (2006)
28. `http://www.adastral.ucl.ac.uk/~vladkolm`
29. Bai, X., Sapiro, G.: A geodesic framework for fast interactive image and video segmentation and matting. Technical Report 2185, Institute of Mathematics and Its Applications, Univ. Minnesota Preprint Series(2008)
30. Cremers, D., Osher, S.J., Soatto, S.: Kernel density estimation and intrinsic alignment for shape priors in level set segmentation. IJCV 69 (2006)
31. Yatziv, L., Sapiro, G.: Fast image and video colorization using chrominance blending. IEEE Trans. on Image Processing 15 (2006)

## Appendix – Fast Geodesic Distance Transform

Given a map $M(\mathbf{x}) \in [0,1]$, in the forward pass the map is scanned with a $3 \times 3$ kernel from the top-left to the bottom-right corner and the intermediate function $C(\mathbf{x})$ is iteratively constructed as illustrated in fig. 12. The north-west, north, north-east and west components of the image gradient $\nabla I$ are used. The $\rho_1$ and $\rho_2$ local distances are usually set to $\rho_1 = 1$ and $\rho_2 = \sqrt{2}$. In the backward pass the algorithm proceeds from the bottom-right to the top-left corner and applies the backward kernel to $C(\mathbf{x})$ to produce the final distance $D(\mathbf{x})$ (*cf.* fig. 1). Larger kernels produce better approximations to the exact distance.