IMAGE EDITING USING LEVEL SET TREES

Anastasia Dubrovina† *Rom Hershkovitz*† *Ron Kimmel*†

† Computer Science, Technion - Israel Institute of Technology

ABSTRACT

An efficient method for precise computation of image-aware geodesic distances for image editing algorithms is proposed. It exploits the connection between image representation as a mapping from a Cartesian grid and as a collection of its level sets, organized into a tree structure. The distance computation is reformulated in the domain of the image level sets, where it can be calculated without introducing approximation errors, which are unavoidable when working the image domain. Advantages of the proposed approach are demonstrated for image segmentation application.

***Index Terms*—** Level set tree, intrinsic distance calculation, image editing, segmentation

1. INTRODUCTION

Various image editing algorithms, for image segmentation [1, 2, 3], matting [4], denoising [3], and colorization [5, 3], are based on computing image-aware geodesic distances. These distances are usually computed in the image domain, where they can be efficiently approximated using Dijkstra's algorithm [6], or the fast marching [7, 8, 9] or fast sweeping methods [10, 11, 12]. Despite their wide use, each of the above algorithms introduces approximation errors and inconsistencies that may degrade the quality of the editing.

Instead of the standard representation over a Cartesian grid, an image may be alternatively though of as topographic map and represented by the set of its level lines [13]. These level lines may be further organized into a tree structure, producing a hierarchical, contrast-invariant image representation. It was previously exploited for image filtering and registration [14], quantization [15], segmentation [16, 17], and subpixel image level line evolution [18]. Fast algorithms for computing level set-based image representation were proposed in [14] and, recently, in [19].

In this paper, we exploit a related image representation, which we denote by a level set tree. We show that a certain type of image-aware geodesic distance defined in the image domain [5, 2, 4], can be cast into a distance measure defined over this tree. We show that the above distance can be calculated precisely using the level set tree image representation, with the same complexity as in [5, 2, 4], where it is approximated using Dijkstra's algorithm. We further demonstrate its favorable performance for image segmentation.

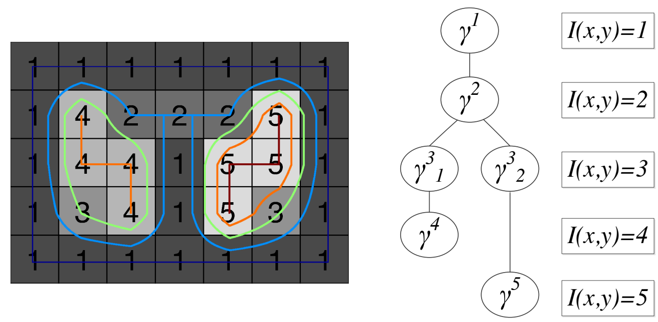

Fig. 1. Synthetic image with its level lines (a), and its level set tree (b).

1.1. Problem formulation

Given an image I defined over the domain $\Omega \subset \mathbb{R}^2$, we consider the following intrinsic image-aware distance measure

$$d(p,p') = \min_{C_{p,p'}} \int_0^1 \left|\nabla I(C(t)) \cdot \dot{C}(t)\right| dt, \tag{1}$$

where $C_{p,p'}(t)$ is a curve in Ω with end points $p, p' \in \Omega$. Yatziv and Sapiro [5] employed $d(p,p')$ to propagate color information from a set of manually colored regions (scribbles), to the rest of the image, while [2, 4] employed it to perform user-assited image segmentation.

Given two points $p, p' \in \Omega$, the above distance measures the minimal total variation of the image $I(x,y)$ along all parametric curves $C_{p,p'}(t)$. Intuitively, the shortest path (in terms of (1)) can be decomposed into segments that follow image level sets, with total length 0, since $|\nabla I \cdot \dot{C}(t)| = 0$ along the level set, and the segments perpendicular to the level lines, with total length $d(p,p')$.

For example, let us consider the synthetic image in Figure 1. Assuming that the image values were sampled from a smooth function $I(x,y)$, the distance between a pixel where $I = 2$ and any pixel where $I = 1$ equals 1, since they belong to two adjacent level sets, shown with cyan and blue curves, respectively. Thus, the distance between them equals to the absolute difference between the two level set values. A more surprising observation is that the distance between the pixels with $I = 2$ and $I = 3$ also equals 1, for the same reason.

Consider now computing $d(p, p')$ in the above example, using Dijkstra's algorithm or the fast marching method. It is easy to see that both methods produce the correct distance between pixels p, p' with $I(p) = 1$ and $I(p') = 2$, but fail for a pair p, p' with $I(p) = 2$ and $I(p') = 3$ (assuming standard 4 or 8-connected pixel neighborhood). Note, that using Dijkstra's algorithm, a partial remedy may be to consider larger pixel neighborhood, at the cost of increased computation complexity.

Thus, neither fast marching, nor Dijkstra's algorithm are the right tools to compute the continuous distance measure defined in Equation (1). On the other hand, the discussion above hints that this distance can be naturally computed using the level set-based image representation. Following this line of thought, we suggest a method for *exact* computation of the distance $d(p, p')$, based on exploitation of the connectivity of the level sets of the image I. We encode this connectivity using a structure we call the *level set tree* of I, and show that (1) can be alternatively computed by traversing the level set tree, instead of approximating it on a graph defined in the image plane, while maintaining the same linear complexity as in [5, 4].

2. LEVEL SET TREES

Given an image I defined over a domain Ω, its k-level set, which we will denote by γ^k, is given by

$$\gamma^k = \{p \in \Omega \mid I(p) = k\}\,. \tag{2}$$

A level set γ^k may consist of zero, one, or more connected components. Let us further denote the i-th connected component of the k-level set by γ_i^k, and the set of all different connected level set components by $\Gamma = \{\gamma_i^k\}_{i,k}$.

These level set components of I can be organized into a tree structure, for instance, by the order of their geometrical inclusion. Figure 1 illustrates the level set tree obtained for a synthetic image discussed previously. The level set tree vertices are given by the connected level set components γ_i^k. The tree edges connect pairs of vertices representing adjacent (in the image plane) connected level set components.

Formally, we describe the level set tree by a weighted graph (Γ, E, F), where Γ is as described above, and $E \subseteq \Gamma \times \Gamma$ is the set of the tree edges. The graph edge weights $F : E \to \mathbb{R}^+$ are given by

$$F(\gamma^k, \gamma^m) = |k - m|, \tag{3}$$

where γ^k, γ^m are two graph vertices connected by an edge. Thus, each edge in E is weighted by the absolute difference of the image values corresponding to γ^k, γ^m, $|k - m|$. Carr *et al*. [20] defined a related structure for general sampled functions, that they called an augmented contour tree, but without introducing edge weights.

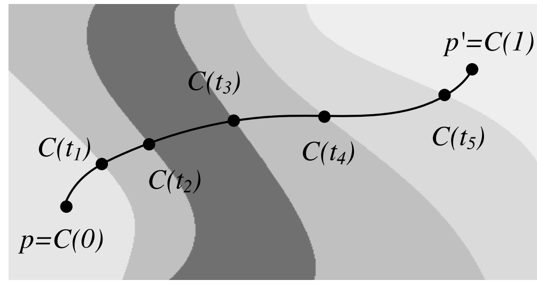

Fig. 2. Decomposition of a path $C(t)$ into a series of segments belonging to adjacent level sets.

2.1. Relation between intrinsic distances and level set trees

To explain how the intrinsic distance $d(p, p')$ may be computed using the level set tree described in the previous section, let us first re-write the integrand in Equation (1) in a somewhat different form

$$\begin{aligned}\left|\nabla I(C(t)) \cdot \dot{C}(t)\right| dt &= \left|\frac{dI}{dx}\frac{dx}{dt} + \frac{dI}{dy}\frac{dy}{dt}\right| dt \\ &= \left|\frac{dI}{dx}dx + \frac{dI}{dy}dy\right| = |dI(C(t))|\,.\end{aligned} \tag{4}$$

Using the above formulation, the intrinsic distance $d(p, p')$ reads

$$d(p, p') = \min_{C_{p,p'}} \int_0^1 |dI(C(t))|\,, \tag{5}$$

and measures the minimal total variation of $I(x, y)$ over all possible paths connecting the points p and p'.

Now, assuming that the image I obtains a discrete set of values over the domain Ω, the path $C(t)$ can be recursively decomposed into a series of non-overlapping segments, belonging to adjacent image level sets

$$C_i(t) = \{C(t),\ t \in [t_i, t_{i+1})\}\,, \quad C(t) = \bigcup_{i \geq 0} C_i(t), \tag{6}$$

where

$$t_{i+1} = \min_{1 \geq t > t_i} t \quad \text{s.t.} \quad I(C(t)) \neq I(C(t_i)). \tag{7}$$

See an illustration of such a decomposition in Figure 2.

According to the above definition, any two consecutive points on the path, $C(t_i)$ and $C(t_{i+1})$, belong to a pair of adjacent level sets γ^{k_i} and $\gamma^{k_{i+1}}$, with values $k_i = I(C(t_i))$ and $k_{i+1} = I(C(t_{i+1}))$, respectively. Therefore, a path between a pair points in the image domain defines a corresponding path in the domain of the image level sets

$$C(t) = \bigcup_{i \geq 0} C_i(t) \qquad \Rightarrow \qquad C_\Gamma = \left\{\gamma^{k_i}\right\}_{i \geq 0}. \tag{8}$$

Now, the relation between the intrinsic distance described by Equations (1,5), and the image level set tree is apparent. By decomposing the path $C(t)$ into a series of such segments, the distance defined in Equation (5) becomes

$$d(p,p') = \min_{C_{p,p'}} \sum_{i\geq 0} \int_{t_i}^{t_{i+1}} |dI(C(t))| = \min_{C_{p,p'}} \sum_{i\geq 0} |k_{i+1} - k_i| \,. \tag{9}$$

According to the definition in Equation (3), the difference $|k_{i+1} - k_i|$ is equal to the weight $F(\gamma^{k_i}, \gamma^{k_{i+1}})$ of the edge between the level sets γ^{k_i} and $\gamma^{k_{i+1}}$ in the level set tree. Thus, the length of the minimal geodesic $C(t) \in \Omega$ between p and p' is equal to the length of the corresponding minimal path in the level set tree

$$d(p,p') = \min_{C_\Gamma(p,p')} \sum_{i\geq 0} F(\gamma^{k_i}, \gamma^{k_{i+1}}), \tag{10}$$

where by $C_\Gamma(p,p')$ we denote a path in the level set tree with end points at $\gamma^{I(p)}$ and $\gamma^{I(p')}$.

Minimal pass uniqueness The authors of [4] stated that there may exist several paths in the image domain minimizing the intrinsic distance (1) between a pair of pixels. On the contrary, in the level set tree, given by an undirected acyclic graph, there exists a single minimal length path between the two nodes of the tree corresponding to these pixels. In other words, for a given pair of pixels and the corresponding nodes of the level set tree, the corresponding minimal length path in the tree encodes all possible minimal length paths between these pixels in the image domain.

2.2. Distance computation using the level set tree

To compute the intrinsic distance from a set of source pixels $\{p_i\}$ to the rest of the image pixels $p \in \Omega$, we suggest the following simple algorithm.

1. Construct the level set tree (Γ, E, F) of the image I.
2. Detect nodes of the level set tree corresponding to the pixels $\{p_i\}$ - these are the source nodes $\{\gamma^k\}$. Compute distances from the source nodes to the rest of the nodes of the level set tree, using Dijkstra's algorithm.
3. For each pixel p in the image, assign to it the distance value obtained for its corresponding level set component.

2.3. Level set tree construction

We compute the level set tree of an image from a related image representation by a tree of shapes [14]. A *shape* is defined as all image pixels belonging to a connected level set component and its interior. Let us denote the outer boundary of a connected level set component γ^k by $J(\gamma^k)$. A shape S^k corresponding to the curve $J(\gamma^k)$ is defined as the image region enclosed by it

$$S^k = \{p \in \Omega \mid p \in \textit{the interior of } J(\gamma^k)\}. \tag{11}$$

The root of the tree of shapes is the whole image domain Ω, while its connectivity is naturally defined by the order of the geometric inclusion of the shapes.

Thus, given a shape S^k and its children $\{S^m\}_m$ in the tree of shapes, the corresponding node of the level set tree, γ^k, is given by all the pixels that belong to S^k, but not to any of its children

$$\gamma^k = \left\{ p \in \Omega \mid \; p \in S^k, p \notin \bigcup_m S^m \right\}. \tag{12}$$

The connectivity E of the level set tree is exactly that of the tree of shapes. To construct the tree of shapes we used a publicly available implementation of [14] from the Megawave library[1].

2.4. Algorithm complexity

The level set tree of an image consisting of N pixels can be constructed from the tree of shapes in linear time $O(N)$, if the image is quantized, and in $O(N \log N)$ otherwise [14, 19]. Distances from a given source point p to the rest of the image points can be calculated in $O(N)$, by exploiting the fact that the level set tree (Γ, E, F) is an undirected acyclic graph [21]. Thus, for quantized images, the overall complexity of the algorithm is $O(N)$, which is the same complexity as the one reported in [4], where the distances were approximated in the image plane using the modified Dijkstra's algorithm of [5].

3. USER-ASSISTED IMAGE SEGMENTATION

We applied the proposed method for intrinsic distance calculation, in conjunction with the algorithm for semi-supervised image segmentation of Bai and Shapiro [4]. In that setting, the user marks pixels belonging to the object and the background by drawing one or more scribbles (markers) inside the object and the background, respectively. Following the notations in [4], we denote the foreground scribble by $\mathcal{F}$ and the background scribble by $\mathcal{B}$.

These scribbles are first used to estimate the likelihood of the image pixels to belong to the foreground/background. Then, for each pixel $p \in \Omega$, we compute the intrinsic distance, defined in Equation (1), between it and the foreground/background scribbles. Instead of the image values, the foreground likelihood values are used as the distance weight. The obtained distances are denoted by $d_{\mathcal{F}}(p)$ and $d_{\mathcal{B}}(p)$, respectively. Finally, each pixel p is assigned the label of the scribble

[1] megawave.cmla.ens-cachan.fr/

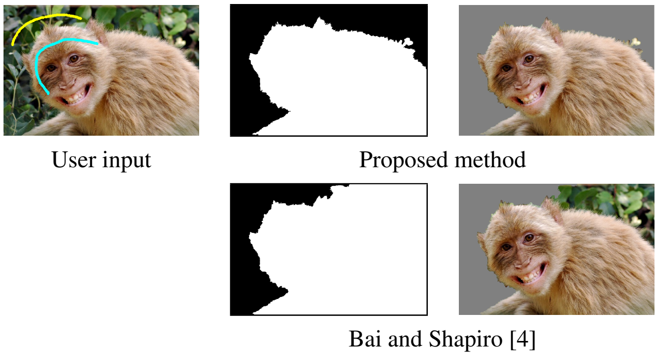

Fig. 3. Comparison of segmentation results obtained with the proposed method and with [4].

closest to it

$$\Omega_{\mathcal{F}} = \{ \; p \in \Omega \mid d_{\mathcal{F}}(p) \leq d_{\mathcal{B}}(p) \; \}, \quad \Omega_{\mathcal{B}} = \Omega \backslash \Omega_{\mathcal{F}}, \quad (13)$$

where $\Omega_{\mathcal{F}}$ and $\Omega_{\mathcal{B}}$ denote the object and the background regions.

The obtained segmentation results are shown in Figures 3 - 6. In Figure 3, the method of [4] fails to produce correct segmentation, due to the metrication error of the Dijkstra algorithm. Using the proposed distance calculation, we succeed to eliminate this error and produce more plausible segmentation.

Images with foreground objects with thin structures are another challenging case for the method of Bai and Shapiro - the metrication error accumulated over the minimal lengths paths prevents [4] from segmenting such objects correctly. Figure 4 presents an example of such image, segmented using the proposed method, alongside state-of-art results presented in [22] and [23], both obtained using significantly more complex techniques. [4] failed to segment the legs of the insect in the image correctly, with both standard 4 and 8-connected pixel neighborhoods.

We further compared the proposed method to the tree-reweighted message passing (TRW) algorithm described in [24], obtaining similar results with both methods, as shown in Figure 5. More segmentation exampled obtained with the proposed method are shown in Figure 6.

4. CONCLUSIONS

In this paper, we exploited image representation by the tree of its level sets, for efficient and precise computation of image-aware geodesic distances, utilized for image editing tasks. The proposed distance computation method does not suffer from approximation errors produced by standard algorithms that are applied in the image domain. In our future research, we plan to extend the proposed method for additional image-based distances computation and other image editing tasks.

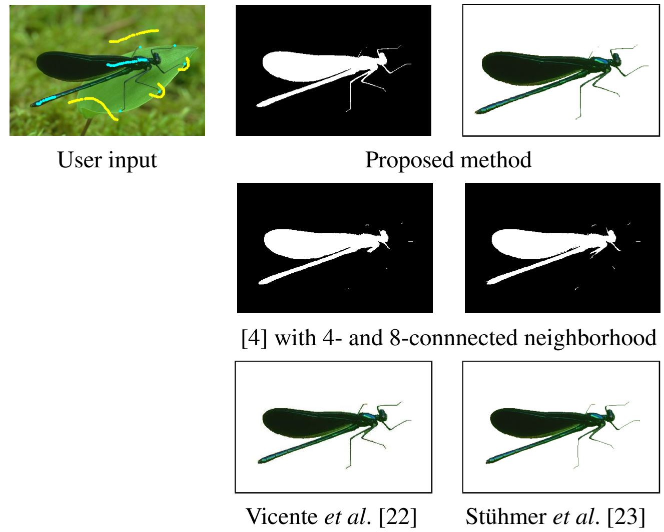

Fig. 4. Example of segmenting an image with thin structures.

Fig. 5. Segmentation examples from the Middlebury benchmark [24].

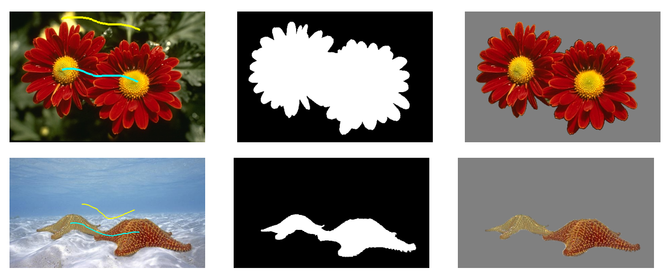

Fig. 6. Segmentation results obtained using the proposed method.

5. ACKNOWLEDGEMENTS

This work has been supported by grant agreement no. 267414 of the European Community FP7-ERC program.

6. REFERENCES

[1] T. Deschamps and L. D. Cohen, "Fast extraction of minimal paths in 3d images and applications to virtual en-

doscopy," *Medical Image Analysis*, vol. 5, no. 4, pp. 281–299, 2001.

[2] A. Protiere and G. Sapiro, "Interactive image segmentation via adaptive weighted distances," *Image Processing, IEEE Transactions on*, vol. 16, no. 4, pp. 1046–1057, 2007.

[3] A. Criminisi, T. Sharp, C. Rother, and P. Pérez, "Geodesic image and video editing." *ACM Trans. Graph.*, vol. 29, no. 5, p. 134, 2010.

[4] X. Bai and G. Sapiro, "Geodesic matting: A framework for fast interactive image and video segmentation and matting," *International Journal of Computer Vision*, vol. 82, no. 2, pp. 113–132, 2009.

[5] L. Yatziv and G. Sapiro, "Fast image and video colorization using chrominance blending," *Image Processing, IEEE Transactions on*, vol. 15, no. 5, pp. 1120–1129, 2006.

[6] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische mathematik*, vol. 1, no. 1, pp. 269–271, 1959.

[7] J. N. Tsitsiklis, "Efficient algorithms for globally optimal trajectories," *Automatic Control, IEEE Transactions on*, vol. 40, no. 9, pp. 1528–1538, 1995.

[8] J. A. Sethian, "A fast marching level set method for monotonically advancing fronts," *Proceedings of the National Academy of Sciences*, vol. 93, no. 4, pp. 1591–1595, 1996.

[9] L. Yatziv, A. Bartesaghi, and G. Sapiro, "O(N) implementation of the fast marching algorithm," *Journal of computational physics*, vol. 212, no. 2, pp. 393–399, 2006.

[10] P.-E. Danielsson, "Euclidean distance mapping," *Computer Graphics and image processing*, vol. 14, no. 3, pp. 227–248, 1980.

[11] H. Zhao, "A fast sweeping method for eikonal equations," *Mathematics of computation*, vol. 74, no. 250, pp. 603–627, 2005.

[12] R. Kimmel and R. P. Maurer, "Method of computing sub-pixel Euclidean distance maps," Sep. 26 2006, US Patent 7,113,617.

[13] V. Caselles, B. Coll, and J.-M. Morel, "A Kanizsa programme," in *12th International Conference on Analysis and Optimization of Systems Images, Wavelets and PDEs Paris*, 1996, pp. 356–359.

[14] P. Monasse and F. Guichard, "Fast computation of a contrast-invariant image representation," *Image Processing, IEEE Transactions on*, vol. 9, no. 5, pp. 860–872, 2000.

[15] C. Ballester, V. Caselles, and P. Monasse, "The tree of shapes of an image," *ESAIM: Control, Optimisation and Calculus of Variations*, vol. 9, pp. 1–18, 2003.

[16] Y. Xu, T. Géraud, and L. Najman, "Context-based energy estimator: Application to object segmentation on the tree of shapes," in *Image Processing (ICIP), 2012 19th IEEE International Conference on*. IEEE, 2012, pp. 1577–1580.

[17] ——, "Two applications of shape-based morphology: blood vessels segmentation and a generalization of constrained connectivity," in *Mathematical Morphology and Its Applications to Signal and Image Processing*. Springer Berlin Heidelberg, 2013, pp. 390–401.

[18] A. Ciomaga, P. Monasse, and J.-M. Morel, "Level lines shortening yields an image curvature microscope," in *Image Processing (ICIP), 2010 17th IEEE International Conference on*. IEEE, 2010, pp. 4129–4132.

[19] T. Géraud, E. Carlinet, S. Crozet, and L. Najman, "A quasi-linear algorithm to compute the tree of shapes of nd images," in *Mathematical Morphology and Its Applications to Signal and Image Processing*. Springer, 2013, pp. 98–110.

[20] H. Carr, J. Snoeyink, and U. Axen, "Computing contour trees in all dimensions," in *Proceedings of the eleventh annual ACM-SIAM symposium on Discrete algorithms*. Society for Industrial and Applied Mathematics, 2000, pp. 918–926.

[21] M. Thorup, "Undirected single-source shortest paths with positive integer weights in linear time," *Journal of the ACM (JACM)*, vol. 46, no. 3, pp. 362–394, 1999.

[22] S. Vicente, V. Kolmogorov, and C. Rother, "Graph cut based image segmentation with connectivity priors," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*. IEEE, 2008, pp. 1–8.

[23] J. Stühmer, P. Schröder, and D. Cremers, "Tree shape priors with connectivity constraints using convex relaxation on general graphs," in *IEEE International Conference on Computer Vision (ICCV)*, Sydney, Australia, December 2013.

[24] R. Szeliski, R. Zabih, D. Scharstein, O. Veksler, V. Kolmogorov, A. Agarwala, M. Tappen, and C. Rother, "A comparative study of energy minimization methods for markov random fields with smoothness-based priors," *IEEE PAMI*, vol. 30, no. 6, pp. 1068–1080, 2008.