

The Dahu graph-cut for interactive segmentation on 2D/3D images

Minh Ôn Vũ Ngọc

Edwin Carlinet

Jonathan Fabrizio

Thierry Géraud

Submission: 2021-03-15, Accepted: 2022-11-24

Abstract

Interactive image segmentation is an important application in computer vision for selecting objects of interest in images. Several interactive segmentation methods are based on distance transform algorithms. However, the most known distance transform, geodesic distance, is sensitive to noise in the image and to seed placement. Recently, the Dahu pseudo-distance, a continuous version of the minimum barrier distance (MBD), is proved to be more powerful than the geodesic distance in noisy and blurred images. This paper presents a method for combining the Dahu pseudo-distance with edge information in a graph-cut optimization framework and leveraging each's complementary strengths. Our method works efficiently on both 2D/3D images and videos. Results show that our method achieves better performance than other distance-based and graph-cut methods, thereby reducing the user's efforts.

1 Introduction

Segmentation is one of the most fundamental building blocks in computer vision and pattern recognition. Over the past several decades, researchers have proposed various segmentation approaches to deal with different tasks. However, automatic segmentation is still a big challenge to achieve accurate results because of poor image quality, ambiguity, and inhomogeneous object regions. Therefore, interactive segmentation, which involves human intervention, is proposed. It is a target driven task so that the users can add or remove markers to refine the current segmentation. The additional information is presented in many forms, including tracing points on the object's boundary [25, 5], scribbles/points inside and outside of the objects [6, 4, 14], or a bounding box [37]. We focus here on the approaches where the user puts scribbles on the image to define background and foreground regions.

Numerous approaches have been proposed for interactive segmentation. Many methods do not work on the image space but other image representations. Among them, the graph-cut-based framework, which was introduced by Boykov et al., has been the most popular one [6]. In this paper, an image is considered as a graph, and user inputs represent the hard constraints in this graph. The graph-cut problem is to find the minimum cut that optimizes an energy function, which balances region and boundary information. The image is first transformed into a probability map based on the color distribution of foreground/background pixels. Then, the minimization function transforms this probability map into a binary image that trades off the fidelity to the input and the regularity of the regions. To improve this method, GrabCut [37] is proposed by using a Gaussian Mixture Model (GMM) However, these methods have a limitation that produces short-cutting segmentation that cuts across the interior of an object due to the boundary term's bias [36].

Other researchers have proposed interactive segmentation approaches with morphological hierarchies, where the image representations are organized into the tree structures and produce hierarchies. Using this representation, interactive segmentation is similar to classifying the nodes of the underlying tree and reconstructing the image from the labeled tree. There are two kinds of tree: partition tree and threshold decomposition tree. Some typical partition trees used for interactive segmentation are the constrained connectivity (CC) [39], watershed hierarchies (WS) [13], and observation scale hierarchy [21]. For the second kind of tree, in [15, 8], the interactive segmentation relies on the tree of shapes. Our initial idea is tree, we do it directly on the image.

Besides, several interactive segmentation methods have relied on distance-based algorithms. Distance transforms are used in image processing to find the minimum distance between pixels and a set of markers. Interactive segmentation is equivalent to a classification of the pixel labels according to their distances to the set of markers. In [4, 14], the authors have applied the geodesic distance for interactive segmentation in images and videos. Price et al. [36] integrate geodesics distance with graph-cut to take advantage of their relative strengths.

Lately, a new path-wise distance metric, the minimum barrier distance (MBD) [40, 45, 23], has shown to have low sensitivity to noise, blur, and seed positions. This distance has been successfully used for salient objects detection [43, 45]. Subsequently, the Dahu pseudo-distance, a continuous version of the MBD in the domain of mathematical morphology, has been presented in [19]. The computation of the Dahu pseudo-distance relies on a tree representation: the Tree of Shapes [18]. This distance has been proven useful in salient object detection [29, 30, 28]. Inspired from [8], which segments object regions in the continuous image representation, we employ here a method for interactive segmentation using the

Dahu pseudo-distance. This paper introduces a new method for interactive segmentation that makes the six following contributions.

- We present a method to compute the exact Dahu pseudo-distance for multivariate images directly on the image and do not rely on the Tree of Shape.
- To save time, we propose an efficient method for computing the approximation Dahu pseudo-distance.
- We combine the Dahu pseudo-distance information with edge information in a graph-cut optimization framework for interactive segmentation.
- We propose a strategy to improve interactive segmentation by adding a probability map (computed using a Gaussian mixture model) as a channel of the image and take advantage of our ability to compute the Dahu pseudo-distance on multivariate images.
- We propose a complete scheme for interactive segmentation not only efficient on 2D images but also usable, thanks to the Dahu pseudo-distance, on 3D images (and potentially on n D images in general).
- We investigate the use of the Dahu pseudo-distance in our scheme for segmenting objects in videos.

The paper is organized as follows. Section 2 recalls some fundamental backgrounds related to our method. In Section 3, we present an efficient method to compute the exact/approximation Dahu pseudo-distance on multivariate images. Section 4 presents our proposed method, which leverages the Dahu pseudo-distance and edge information for interactive segmentation on 2D images. This section is the opportunity for us to show that our scheme is very versatile and we extend it to 3D images and video sequences. In Section 5, Section 6, and Section 7, our proposed method is evaluated and compared with state-of-the-art approaches for interactive segmentation on 2D/3D images and image sequences. The conclusions and perspectives are discussed in Section 8.

2 Background

This section recalls the definition of the MBD and the Dahu pseudo-distance, which is the cornerstone of our paper. From this distance, it is easy to derive a distance map or a saliency map, which may be used to perform segmentation.

2.1 The Minimum Barrier Distance (MBD)

The MBD has been defined in [40, 12], in which a gray-level image (Fig. 1a) is considered as a vertex-valued graph (Fig. 1b). Let $\pi = \langle \dots, \pi_i, \pi_{i+1} \dots \rangle$ denote the path of pixels on the graph. The link between π_i and π_{i+1} is the adjacency relation \mathcal{N} between two vertices (4-adjacency in 2D or 6-adjacency in 3D images). The *barrier strength* τ of a path π in the given gray-level image u is defined as:

$$\tau_u(\pi) = \max_{p_i \in \pi} u(p_i) - \min_{p_i \in \pi} u(p_i). \quad (1)$$

The barrier strength is the dynamic distance between the highest and lowest pixel value along this path (Fig. 2). The *minimum barrier distance* d_u^{MB} between two vertices x and x' in u is then defined as the minimum of the barrier strengths of all the paths between two given vertices:

$$d_u^{\text{MB}}(x, x') = \min_{\pi \in \Pi(x, x')} \tau_u(\pi), \quad (2)$$

where $\Pi(x, x')$ denotes the family of all paths that connect two points x and x' .

An example of the MBD is illustrated in Fig. 1b: the minimal path between two red points x, x' is depicted in blue and corresponds to the sequence of values $\langle 1, 0, 0, 0, 2 \rangle$; we obtain $d_u^{\text{MB}}(x, x') = 2$.

2.2 The Dahu pseudo-distance and its usage

The Dahu pseudo-distance The Dahu pseudo-distance is simply a variation of the MBD. The Dahu pseudo-distance, defined in [19], considers an image (Fig. 1a) as a surface (Fig. 1e). However, a scalar function is not well-suited to describe its elevation. In [19], 2D cubical complexes are used to describe this surface. A 2D cubical complex is a set of elements: 2D, 1D, and 0D elements, where 2D elements correspond to the original pixels, 1D and 0D elements are inter-pixels, which take the interval value from its adjacency 2D elements to perform a transition step. Those intervals represent all level-lines passing between any pair of pixels (Fig. 1c). It is illustrated in Fig. 1d, where the purple part corresponds to the 1D element with the purple border in Fig. 1f.

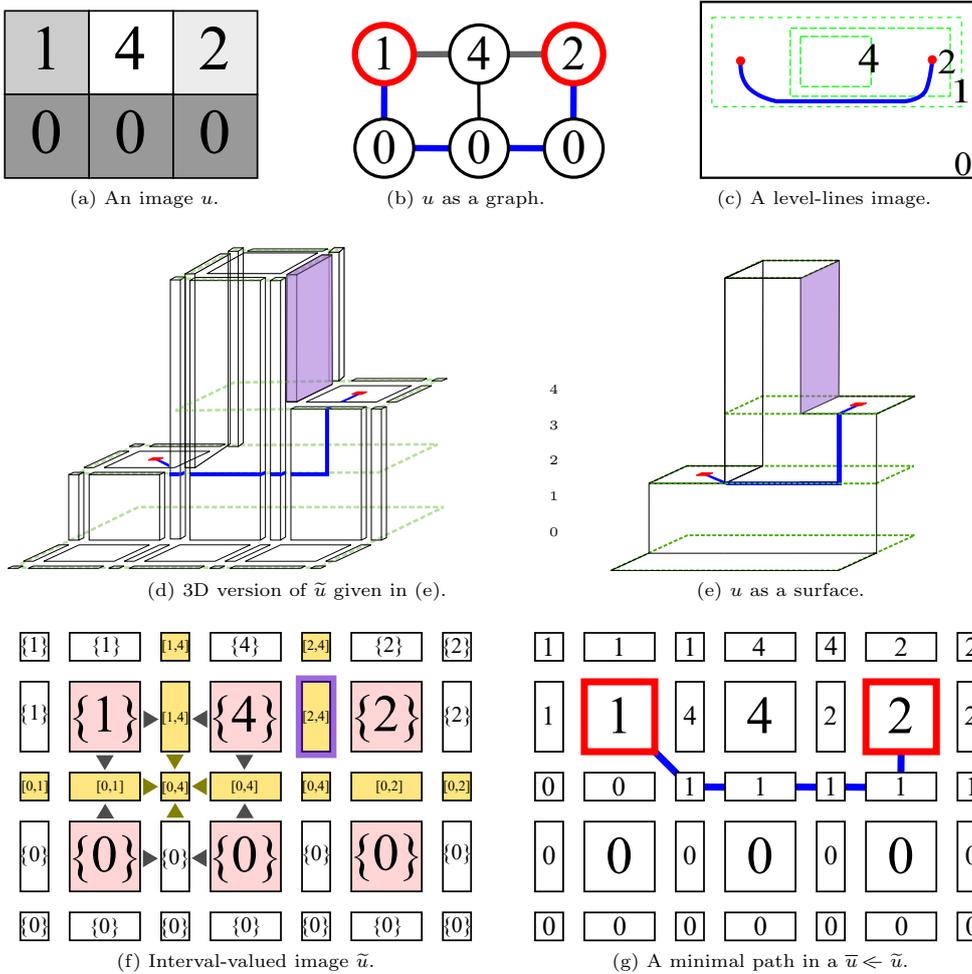


Figure 1: Image representations for computing barrier distances [19].

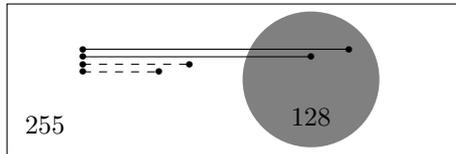


Figure 2: The barrier strength of a path: the barrier values of the two dashed paths are 0, and the barrier values of the two plain paths are $255 - 128 = 127$. The barrier strength of a path is not related to the length of this path.

Intuitively, a scalar image \bar{u} (Fig. 1g) is *included* in the interval-valued image \tilde{u} (Fig. 1f) for all pixels, $\bar{u}(x) \in \tilde{u}(x)$. Their inclusion relationship is denoted by $<$. The Dahu pseudo-distance is defined as:

$$d_u^{\text{DAHU}}(x, x') = \min_{\bar{u} < \tilde{u}} d_{\bar{u}}^{\text{MB}}(h_x, h_{x'}) \quad (3)$$

$$= \min_{\bar{u} < \tilde{u}} \min_{\pi \in \Pi(h_x, h_{x'})} \tau_{\bar{u}}(\pi), \quad (4)$$

where h_x is the 2D element, which corresponds to x . It means that we search for a MBD in the cubical complex, considering all the possible scalar functions \bar{u} that are “included” in the interval-valued map \tilde{u} . The optimal path between two red points in the interval-valued image is depicted as a blue path on Fig. 1g.

The usage of the Dahu pseudo-distance It is straightforward to derive a distance map from the Dahu pseudo-distance. Given a path cost function and a set X of seed points, a distance map S^{DAHU} can be computed by:

$$S_u^{\text{DAHU}}(x, X) = \min_{x' \in X} d_u^{\text{DAHU}}(x, x'), \quad (5)$$

In [19, 29, 30], the Dahu pseudo-distance can be easily computed for a single source (pixel) thanks to the tree of shapes [18]. An efficient version of the Dahu pseudo-distance for multivariate images is presented in [28]. However, it

needs additional time to construct the tree. This is the main reason that encourages us to propose a rational way to compute this distance from the original images directly. Our proposed method is presented in Section 3.

3 The Dahu distance on multivariate images and its approximation

In this section, we introduce algorithms to compute the distance map derived from the Dahu pseudo-distance. We introduce two versions: one with the exact computation of the Dahu pseudo-distance and one with an approximation. The novelty is that, to save time, these distance maps are computed directly from the images and do not require to build the tree of shape before.

3.1 Computing the exact Dahu pseudo-distance on multivariate images

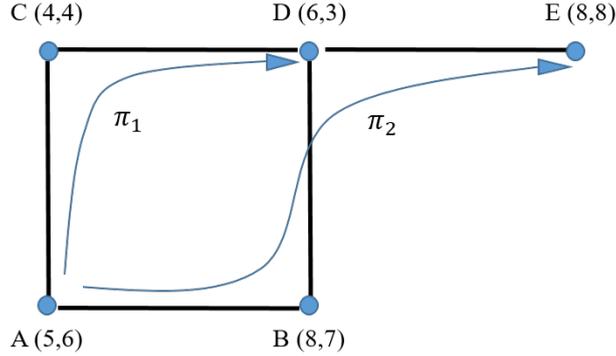


Figure 3: The MBD in the multivariate image (numbers inside the bracket indicate channel colors). π_1 and π_2 are respectively the shortest paths between $A \rightarrow D (= 5)$ and $A \rightarrow E (= 8)$ with respect to the MBD relying on the cityblock diameter.

The Dahu pseudo-distance in a multivariate image is defined as a vectorial function ϕ that takes in parameters the Dahu pseudo-distance on each channel i .

$$d_u^{\text{DAHU}}(x, x') = \min_{\pi \in \Pi(x, x')} \phi(d_{u_i}^{\text{DAHU}}(\pi(x, x'))), \quad (6)$$

In [12], the authors have used four different path-cost functions ϕ , which respectively correspond to the diameter ($\|d_{u_i}^{\text{DAHU}}(\pi(x, x'))\|$), maximum diameter ($\max(d_{u_i}^{\text{DAHU}}(\pi(x, x')))$), city-block diameter ($\sum_{i=0}^n (d_{u_i}^{\text{DAHU}}(\pi(x, x')))$), and volume of the bounding box ($\prod_{i=1}^n (d_{u_i}^{\text{DAHU}}(\pi(x, x')))$). The behaviors of the four path-cost functions are different, and depending on the application, we choose the most suitable one.

We want to compute the exact Dahu pseudo-distance $d_u^{\text{DAHU}}(x, X)$ for every pixel x of the image u from a set of seed X . For every pixel x of the image we save in $\mathbf{p}(x)$ the shortest path from the seed X to the pixel x . Inspired from the MBD's computation in [12], we present an algorithm to compute exactly the Dahu distance map in multivariate images with N channels (i.e., the Dahu pseudo-distance between every pixel in the image and the given seed). Our algorithm (see Algo. 2) is similar to Dijkstra's algorithm. To be able to compute $d_u^{\text{DAHU}}(x, X)$, two auxiliary maps U and L with N channels are used to keep track of the upper and lower bounds on the current path π for each pixel. The Dahu pseudo-distance is computed while considering an image as a landscape. Therefore, in the first step, we must compute the set-valued map \tilde{u} thanks to a span-based interpolation of the image u . $[\tilde{u}.low, \tilde{u}.up]$ represents the interval value of each pixel in the set-valued image \tilde{u} .

Our algorithm computes the exact Dahu distance map from the seed pixels X to other pixels in the image domain. The process follows two steps. During the first step (lines 2 - 10), we initiate the priority queue Q , min/max images L/U , and the pixels's *path*, which belong to the seed set. In the second step, all points in the domain are met using a propagation front (lines 12 - 31) starting from the seed points. An idea behind this function is that we consider all possible upper bounds U among the paths for every pixel in the domain and then find a path that maximizes the lower bound L . The Dahu pseudo-distance is simply the difference between the upper and lower bound (U and L).

Our algorithm can be explained thoroughly as follows. The propagation procedure is employed by using a hierarchical queue Q . Initially, the Dahu pseudo-distance $d_u^{\text{DAHU}}(x, X)$ of the seed pixels is set at the value 0. We also initialize the $\mathbf{p}(x)$, upper/lower bounds ($U(x)/L(x)$ respectively) at the seed pixels, and the value of the current scalar image \bar{u} . (lines 4-10). These seed pixels are respectively put into the queue Q . In the next step, we pop out pixels from the queue Q

Algorithm 1 GETVALUE: get the closest value l from inter-valued image \tilde{u} to the current scalar image \bar{u} ;
UPDATE: Update the min/max image (L/U) and the distance value $d_u^{\text{DAHU}}(n, X)$ along the path.

<pre> 1: GETVALUE (\tilde{u}, n, x, u_b) 2: ▷ Return: The closest value l of the interval $\tilde{u}(n)$ to $\bar{u}(x)$ 3: ▷ \tilde{u}: interval-valued image 4: ▷ x: current pixel 5: ▷ n: neighbor of $x, n \in \mathcal{N}(x)$ 6: ▷ \bar{u}: current value image 7: ▷ n: neighbor of $x, n \in \mathcal{N}(x)$ 8: for i in N channels do 9: if $\bar{u}_i(x) < \tilde{u}_i.\text{low}(n)$ then 10: $l_i \leftarrow \tilde{u}_i.\text{low}(n)$ 11: else if $\bar{u}_i(x) > \tilde{u}_i.\text{up}(n)$ then 12: $l_i \leftarrow \tilde{u}_i.\text{up}(n)$ 13: else 14: $l_i \leftarrow \bar{u}_i(x)$ 15: end if 16: end for 17: return l </pre>	<pre> 1: UPDATE (U, L, n, x, u_b) 2: ▷ Return: The Dahu pseudo-distance from n to seed set X 3: ▷ L, U: min/max image 4: ▷ x: current pixel 5: ▷ n: neighbor of $x, n \in \mathcal{N}(x)$ 6: ▷ \bar{u}: current value image 7: for i in N channels do 8: $U_i(n) \leftarrow \max(\bar{u}_i(n), U_i(x))$ 9: $L_i(n) \leftarrow \min(\bar{u}_i(n), L_i(x))$ 10: $d_{u_i}^{\text{DAHU}}(n, X) \leftarrow U_i(n) - L_i(n)$ 11: end for 12: return $d_u^{\text{DAHU}}(n, X)$ </pre>
---	---

Algorithm 2 Compute the exact Dahu distance map. Q : priority queue. \mathbf{p}/π : optimal/ current path

<pre> 1: ▷ Return: Dahu distance map $d_u^{\text{DAHU}}(x, X)$ 2: for all $x \in X$ do 3: $\text{path}(x) \leftarrow \langle x \rangle$ 4: $\pi(x) \leftarrow \langle x \rangle$ 5: $L(x), U(x) \leftarrow \tilde{u}.\text{low}(x)$ 6: $\bar{u}(x) \leftarrow \tilde{u}.\text{low}(x)$ 7: $Q.\text{enqueue}(x, \phi(U(x)))$ 8: $d_u^{\text{DAHU}}(x, X) \leftarrow 0$ 9: end for 10: while Q is not empty do 11: $x \leftarrow Q.\text{dequeue}()$ 12: for all neighbors n of p do 13: $l \leftarrow \text{GETVALUE}(\tilde{u}, n, x, u_b)$ 14: $\bar{u}(n) \leftarrow l$ 15: $vmin \leftarrow \min(L(x), \bar{u}(n))$ </pre>	<pre> 16: $vmax \leftarrow \max(U(x), \bar{u}(n))$ 17: if $\phi(vmin) > \phi(L(n))$ then 18: $\pi(n) \leftarrow \pi(x) \wedge n$ 19: $L(n) \leftarrow vmin$ 20: $U(n) \leftarrow vmax$ 21: $d \leftarrow vmin - vmax$ 22: $Q.\text{enqueue}(n, \phi(U(n)))$ 23: if $\phi(d) < \phi(d_u^{\text{DAHU}}(n, X))$ then 24: $\mathbf{p}(n) \leftarrow \pi(n)$ 25: $d_u^{\text{DAHU}}(n, X) \leftarrow d$ 26: end if 27: end if 28: end for 29: end while </pre>
---	---

to the propagation process. Since we use interval-valued maps, we have to decide which level the neighbor pixel will be updated (line 15). The solution is straightforward: a pixel n is enqueued at the value of the interval $\tilde{u}(n)$ that is the closest to $\bar{u}(x)$. This GETVALUE process is mentioned in Algo. 1. For a pixel, which can be reached from different paths, we select the path that has a higher lower bound L (line 19-20) (as it is known that the Dahu pseudo-distance minimizes the distance between the highest and lowest altitude along the path). Next, we need to update the maximum/minimum values ($U(x)/L(x)$ respectively) along the path (lines 21-22). We then sort pixels in the queue according to the upper bound U (line 24). If the updated distance $\phi(d)$ is lower than its previous value, we update the optimal $\mathbf{p}(n)$ and its new distance value $d_u^{\text{DAHU}}(x, X)$ (lines 25-28). The process is repeated until the queue is empty.

A simple example of the Dahu pseudo-distance in the multivariate image $\bar{u} \ll \tilde{u}$ (Fig. 3) is illustrated in Fig. 4. $\pi_1 = \langle A, C, D \rangle$ and $\pi_2 = \langle A, B, D, E \rangle$ respectively represent the shortest paths between $A \rightarrow D$ and $A \rightarrow E$. If we use the city-block diameter as the path cost function, $d_u^{\text{DAHU}}(A, D) = 2 + 3 = 5$ and $d_u^{\text{DAHU}}(A, E) = 3 + 5 = 8$. The process to compute the exact Dahu pseudo-distance is illustrated in Fig. 4. The priority queue is shown in the left hand side and the propagation is shown in the right hand side.

The algorithm's complexity is $\mathcal{O}(mn \log n)$, where n is the number of pixels in the image, and m is the size of the fixed set of the possible values of the pixels in the image. This algorithm is time-consuming to compute the exact Dahu pseudo-distance. Therefore, to reduce the runtime and maintain the Dahu pseudo-distance quality, we introduce in Section 3.2, an efficient approximation.

3.2 Algorithm finding an approximation of the Dahu pseudo-distance

We proposed here a method to efficiently compute an approximation of the Dahu pseudo-distance. This method allows us to quickly compute the Dahu distance map using the front propagation approach, which is inspired from [23].

The algorithm for approximation Dahu distance map is exposed in Algo. 3 and consists of two steps. In the first step (lines 2 - 9), we initialize the upper/lower bound U/L respectively and distance the value $d_u^{\text{DAHU}}(x, X)$ of the seed pixels.

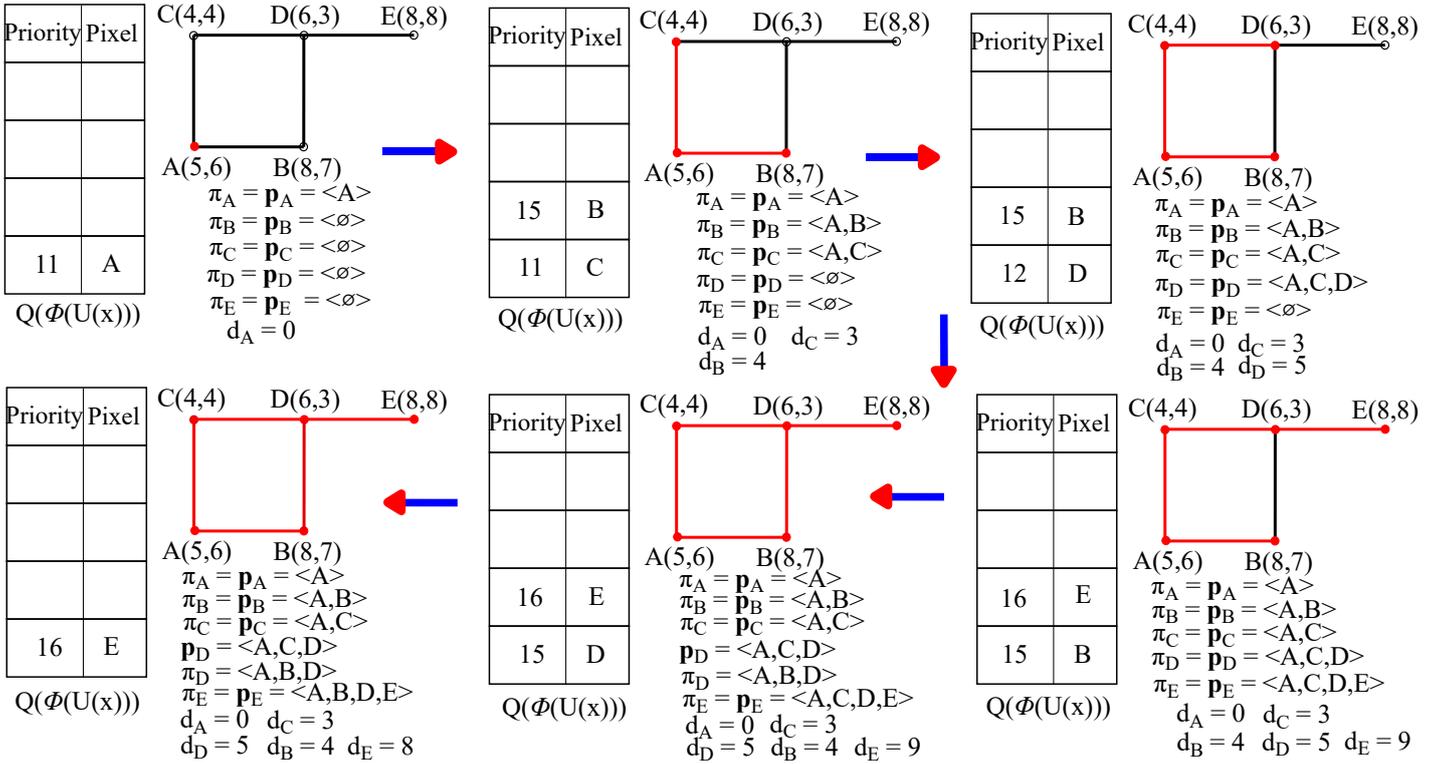


Figure 4: The computation of the exact Dahu pseudo-distance in the multivariate image \bar{u} . The priority queue (Q) keeps pixels in the order. The color represents the propagation process.

We then put these pixels into the queue Q and change the state of the seed pixels from “unseen” to “inqueue”.

In the propagation process, value l , which is got from GETVALUE function (Algo. 1), is used to update the lower/upper values ($L(n)/U(n)$ respectively) from the current pixel x (n is a neighbor of x), thereby updating the Dahu pseudo-distance $d_u^{\text{DAHU}}(n, X)$ (line 24). If the status of the neighbor pixel n of x is “unseen”, the Dahu pseudo-distance of n is simply updating the lower/upper values L/U along the path according to Algo.1 (lines 19 - 22). In the case where the status of the pixels is “inqueue”, we should compare whether the new cost $\phi(d)$ of propagation from p to n is lower than $\phi(d_u^{\text{DAHU}}(n, X))$. Once this condition is satisfied, pixel n is appended to the queue Q at slot $d_u^{\text{DAHU}}(x, X)$ (lines 25 - 28). This loop continues until all of the pixels are “done”. The approximation Dahu distance map is then obtained. The propagation process is depicted in Fig. 5. Note that the value of d_E in this case is 9 which is higher than the case of the exact Dahu pseudo-distance.

The complexity of this algorithm is $\mathcal{O}(n \log n)$, which is more efficient than the exact computation algorithm. In Section 5.1, we will evaluate the efficiency of our method comparing to the exact version.

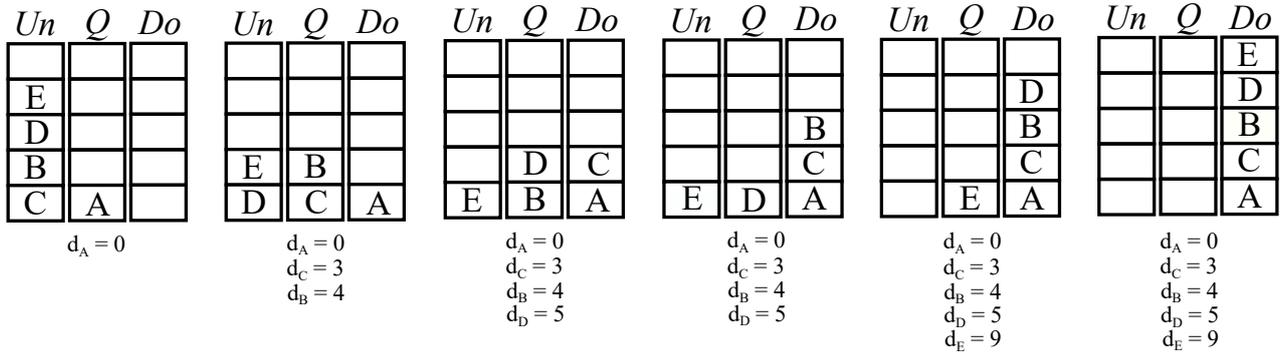


Figure 5: The process to compute the approximation Dahu pseudo-distance in the multivariate image \bar{u} in Fig. 3. Un, Q, Do represent the states *unseen, inqueue, done* of the pixels.

4 Our proposed method: The Dahu graph-cut method

We present here our method for interactive segmentation using the Dahu pseudo-distance. In general, the distance-based methods measure the distance between pixels and markers on images to analyze the difference between the foreground and

Algorithm 3 Compute the approximation Dahu distance map. Q : priority queue. $state$: pixel's state

```

1: ▷ Return: Dahu distance map  $d_u^{\text{DAHU}}(x, X)$ 
2: for all  $x \in X$  do
3:    $state(x) \leftarrow \text{"inqueue"}$ 
4:    $L(x), U(x) \leftarrow \tilde{u}.low(x)$ 
5:    $\bar{u}(x) \leftarrow \tilde{u}.low(x)$ 
6:    $d_u^{\text{DAHU}}(x, X) \leftarrow 0$ 
7:    $Q.enqueue(d_u^{\text{DAHU}}(x, X))$ 
8: end for
9: while  $Q$  is not empty do
10:   $x \leftarrow Q.dequeue()$ 
11:  if  $state(x) = \text{"done"}$  then
12:    skip
13:  end if
14:   $state(x) \leftarrow \text{"done"}$ 
15:  for all neighbors  $n$  of  $x$  do
16:     $l \leftarrow GETVALUE(\tilde{u}, n, x, u_b)$ 
17:     $\bar{u}(n) \leftarrow l$ 
18:    if  $state(n) = \text{"unseen"}$  then
19:       $d_u^{\text{DAHU}}(n, X) \leftarrow UPDATE(U, L, n, x, u_b)$ 
20:       $Q.enqueue(n, \phi(d_u^{\text{DAHU}}(n, X)))$ 
21:       $state(n) \leftarrow \text{"inqueue"}$ 
22:    else
23:      if  $state(n) = \text{"inqueue"}$  and  $\phi(d_u^{\text{DAHU}}(n, X)) >$ 
 $\phi(d_u^{\text{DAHU}}(x, X))$  then
24:         $d \leftarrow UPDATE(U, L, n, x, u_b)$ 
25:        if  $\phi(d_u^{\text{DAHU}}(n, X)) > \phi(d)$  then
26:           $d_u^{\text{DAHU}}(n, X) \leftarrow d$ 
27:           $Q.enqueue(n, \phi(d_u^{\text{DAHU}}(n, X)))$ 
28:        end if
29:      end if
30:    end if
31:  end for
32: end while

```

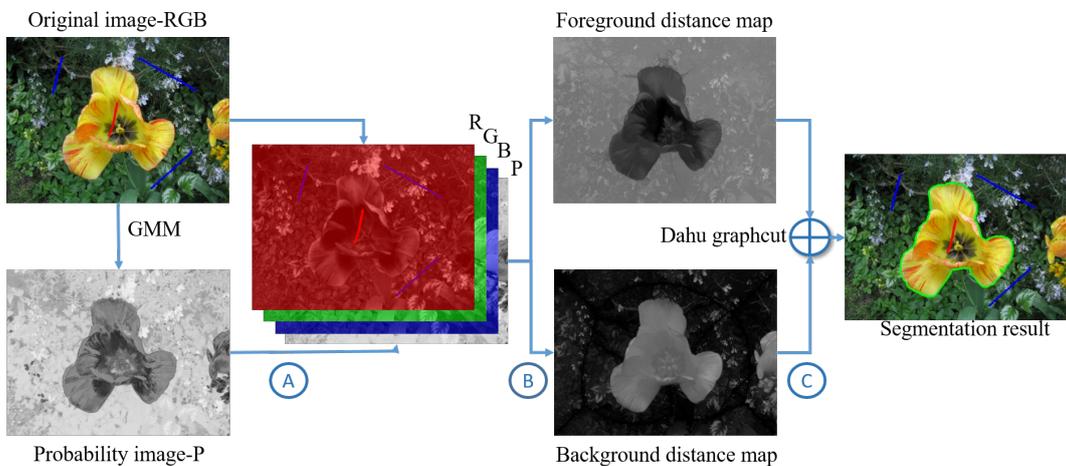


Figure 6: Our scheme for interactive segmentation. A probability image, computed from the markers on the image (using GMM), is combined with the color image (A). Then, the foreground/background Dahu distance maps from every pixel to the markers are computed (B). An optimized graph-cut model is used to segment object regions in the image (C).

background [4, 14, 36]. Usually, this strategy is faced with Additionally, in [36], the authors state that the distance-based segmentation often produces a noisy segmentation on the complex background or foreground.

Our method is decomposed into three steps, which are illustrated in Fig. 6. In the first step, we generate the probability image from the given markers using the gaussian mixture model (GMM). Next, we propose an approach to improve interactive segmentation by adding this probability map as a channel of the image and taking advantage of the Dahu pseudo-distance’s ability to handle multivariate images, as discussed in Section 3.2. Then, we respectively compute the Dahu distance maps from the background/foreground markers. In the last step, these distance maps are considered as the confidence map or the unary term for the optimized graph-cut model. Our method leverages both the Dahu pseudo-distance and edge information, combined in a graph-cut framework to segment object regions in the image. We will detail our framework in the following subsections. Our method is very versatile and easily extended to 3D images and video sequences without effort.

4.1 Compute the probability image and merge with the color image

As previously mentioned, our method starts with two types of user-provided scribbles: the foreground and background scribbles. The problem now is to learn prior information from scribbles and propagate it to the rest of the image. A common approach is to compute a probability map representing the likelihoods of pixels that belong to the foreground (resp. background). This likelihood can be computed from the provided scribbles via the fast kernel density estimation [4, 15, 36] or the GMM [11, 37]. In our case, we use two GMMs with $K = 5$ components on the background and foreground regions. From the obtained GMMs, we predict and normalize the pixel probability that can be associated with the foreground or background.

Besides the advantage of being a fast algorithm for learning the mixture model, GMM has a problem with the number of marker points. When the number of provided points per mixture is not sufficient, the usability of GMM may cause loss of information from the original image. Thus, we merge the probability image from GMM with the original color image to obtain more color and prior information from the scribbles. In the next subsection, we present how to deduce a confidence map from this combined image.

4.2 Compute the multivariate Dahu distance maps

The idea of using the Dahu pseudo-distance for interactive segmentation is inspired from [15, 8, 28], which use the level set trees (tree of shapes) to compute the exact distance. However, segmentation on the tree has a problem when the object of interest is cut across by some background level-lines. In that case, we cannot segment the foreground and the background correctly. Thanks to Algo. 3, the Dahu pseudo-distance can now be computed directly on multivariate images. We apply the Dahu pseudo-distance to compute the distance maps from foreground/background scribbles on the combination of the color channels and the probability image. These distance maps are used to compute a confidence map for the graph-cut model.

We choose the city-block diameter as the function ϕ for computing our distance. The Dahu pseudo-distance on an N channels image I is computed as:

$$d_u^{\text{DAHU}}(x, x') = \min_{\pi \in \Pi(x, x')} \frac{1}{N} \sum_{i=0}^N (d_{u_i}^{\text{DAHU}}(\pi(x, x'))), \quad (7)$$

The confidence map $C(x)$ is defined as:

$$C(x) = \frac{d_u^{\text{DAHU}}(x, l)}{d_u^{\text{DAHU}}(x, X) + d_u^{\text{DAHU}}(x, \bar{X})} \quad (8)$$

where \bar{X} is the marker that has an opposite label \mathcal{L} of X and l can be X or \bar{X} .

The confidence map is then integrated into the optimization graph-cut framework, which is presented in Section 4.3.

4.3 The Dahu graph-cut on 2D images

In many cases, the Dahu pseudo-distance between the pixels and the background and foreground scribbles are equal. That is ambiguous for us to determine the label for these pixels. The lack of edge modeling in distance-based approaches limits their ability to precisely localize object boundaries, something at which graph-cut methods generally excel. Here, we present a method for combining the Dahu pseudo-distance with edge information in a graph-cut optimization framework, leveraging each's complementary strengths.

A graph-cut problem tries to find the optimal label image \mathcal{L} to minimize the energy function as below.

$$E(I) = \sum_{x \in \Omega} (C(x) + \lambda \sum_{x' \in \mathcal{N}(x)} f_{\mathcal{L}}(x, x')) \quad (9)$$

where Ω is the domain of the image I , the Dahu confidence map $C(x)$ is considered as the unary region term and $f_{\mathcal{L}}(x, x')$ expresses the pairwise term for assigning the cost between a pixel pair x and x' :

$$f_{\mathcal{L}}(x, x') = \begin{cases} 0 & \text{if } \mathcal{L}_x = \mathcal{L}_{x'} \\ \exp\left(\frac{-|u_x - u_{x'}|^2}{2\sigma^2}\right) & \text{if } \mathcal{L}_x \neq \mathcal{L}_{x'} \end{cases} \quad (10)$$

This cost is designed to assign small values to pairs of neighbor pixels with different labels and high distinction. It aids in placing edges near object boundaries. λ value controls the trade-off between unary and pairwise terms. Small λ favors the impact of the Dahu pseudo-distance. We try with several values of λ and finally set the value of λ equal to 0.2 and the experiments (Section 5) show that our method achieves stable performance with this value of λ . We aim to minimize Eq. (9) with respect to \mathcal{L}_x to achieve optimal pixel labeling. For this reason, we adopt the max-flow algorithm [6] to solve this problem.

Our proposed method can also be extended for interactive segmentation on 3D images. We present it in Section 4.4.

4.4 The Dahu graph-cut on 3D images

Three-dimensional images are increasingly used in many applications, especially in medical images. Here, we show how to use our Dahu graph-cut for interactive segmentation of 3D images.

There is no added difficulty in higher dimensions. The distance map on the 3D images is computed using the propagation approach similar to the distance computation on 2D images. The Dahu distance maps are computed; then, the graph-cut method is applied to segment the regions. The difference is that the Dahu pseudo-distance is computed on 3D voxels by considering the relationship between neighboring voxels in different slices. This means that our method is directly usable in 3D. The strength of our method is that in contrast to many traditional methods, which handle each 2D slice of the 3D image separately, our method needs markers on the background and foreground on only one slice and segment the area for all slices in the 3D image.

4.5 The Dahu graph-cut on video sequences

Our segmentation method can be easily used to segment moving objects in video sequences. To do so, we consider a video sequence as a 3D image where each frame is a slice image, we look for the shortest path in a sequence of voxels that minimizes the Dahu pseudo-distance as done in Section 4.4. Starting from one frame (i.e., one 3D slice), the propagation process is then accomplished on the entire video. As for 2D images, after computing the Dahu pseudo-distance, the Dahu graph-cut is used to segment the target object in the video. It is possible to put some additional markers to correct the segmentation result until all the target object regions are segmented.

This scheme allows us, from the user-defined scribbles, to provide a background/foreground segmentation. The efficiency of this scheme will be evaluated in the next section; however, thanks to the usage of the Dahu pseudo-distance, our scheme is much more flexible, and it is important to notice this point. We combine the original image with the probability image in the first step, but we are not restricted to 2D/3D color channel images. The Dahu pseudo-distance can manage more dimensions; we can then apply our scheme to 3D images or video sequences. We will study these cases in the following sections.

5 Experimental Results on 2D images

In this section, we present an evaluation of the performance of the Dahu pseudo-distance. First, we analyze the difference between the approx- and exact-Dahu distances. Second, the Dahu graph-cut is compared with other state-of-the-art methods for interactive segmentation. We conduct extensive experiments on four publicly available benchmarks, including Grabcut [5], Weizmann [1], Gulshan [22], and ECSSD [38], to demonstrate the effectiveness and the generalization capacities of our proposed method. Finally, we conduct an ablation study to analyse the contributions of all the components of our method.

5.1 An evaluation of the approx-Dahu distance

We evaluate here the practical usefulness of the approx-Dahu distance map (Algo. 3). As presented in Section 2.2, the Dahu pseudo-distance is a continuous version of the MBD. Especially, in [28], the author demonstrated that the ability to distinguish object/background and the noise stability of the Dahu pseudo-distance is better than the original MBD. Thus, we want to investigate the ability of our approx-Dahu pseudo-distance to see if it still inherits the strong properties of the Dahu pseudo-distance. To do that, we respectively compute the difference between the approx-Dahu pseudo-distance and MBD [12] with the exact-Dahu pseudo-distance. Besides, we also consider the effect of the number of seed points to the distance by evaluating the mean absolute error (MAE) values between distances. Additionally, we compare the approx- and exact-Dahu distance in terms of execution time.

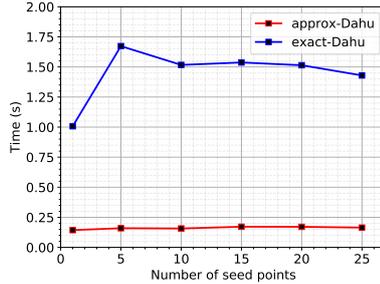
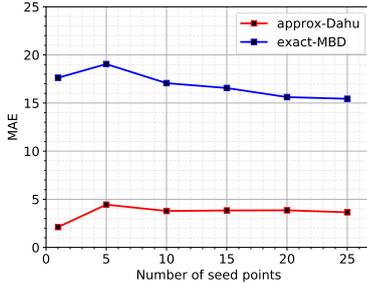
First, we use 151 images from the Gulshan dataset [22]. The average image’s resolution is 500 x 375. The number of seed points (ranged from 1 to 25) is generated randomly in the image. The MAE is defined as follow:

$$MAE = \sum_{i=1}^N \frac{|d_u^{\text{DAHU}}(x_i, X) - d_u^{\text{MB}}(x_i, X)|}{N} \quad (11)$$

where x_i are all pixels, X the set of seed points, d_u^{DAHU} represents the exact-Dahu distance maps and d_u^{MB} is the MBD map.

The MAEs between the approx-Dahu pseudo-distances/exact-MBD and the exact-Dahu pseudo-distance are illustrated in Fig. 7a. At the first glance, we can see that the difference between the approx- and exact-Dahu pseudo-distance is small, much closer than the difference between the exact-Dahu pseudo-distance and the exact-MBD. It proves the usefulness of our approximation to preserve the properties of the Dahu pseudo-distance for discriminating the object/background and the noise stability. Additionally, the differences between the Dahu pseudo-distances and the exact-MBD decrease when we increase the numbers of seed points. The more seed points we use, the more similarities between these distance maps are.

Second, a comparison of the execution time between the approx- and exact-Dahu pseudo-distance w.r.t the number of seed points is illustrated in Fig. 7b. The experiment presented here were conducted on a computer with a 2.6 GHz CPU



(a) MAEs between the approx-Dahu/MBD distances and the exact-Dahu distance. (b) Execution time of the approx/exact-Dahu pseudo-distances.

Figure 7: Comparison between the approx- and exact-Dahu pseudo-distances.

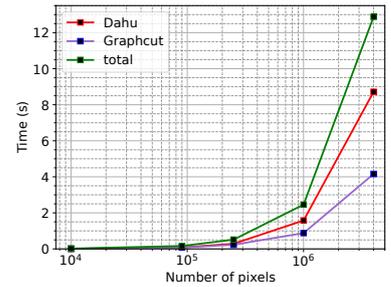


Figure 8: Execution time of our method.

and 8GB of RAM. At first glance, our proposed method is eight times faster than the exact version. Additionally, the executed time of the approx-Dahu algorithm is stable with the number of seeds. It corresponds to the complexity of the algorithm that we presented in Section 3.2. The execution time of our proposed method is also illustrated in Fig. 8.

Some distance maps and segmentation results computed using the “pure” approx/exact-Dahu distances and the MBD are presented in Fig. 9. In these examples, the backgrounds are not homogeneous such as the scene of the sky, the airfield, or the complicated scene in the room. The approx/exact-Dahu pseudo-distance maps seem to work better in these cases. It can be explained by the fact that the front propagation can pass through the inter-pixels. Subsequently, the Dahu pseudo-distance tends to decrease its path. In the next section, we evaluate the usage of the Dahu pseudo-distance for interactive segmentation.

5.2 Evaluation of the Dahu Graph-cut on 2D images

We evaluate in this section the use of our proposed method for interactive segmentation on four popular datasets: Grabcut [5], Weizmann [1], Gulshan [22], and ECSSD [38]. The prior markers in Gulshan and ECSSD datasets are some short scribbles while in Grabcut and Weizmann datasets, the given markers in the image are generated by using the morphological skeletonization method [9]. The target of interactive object segmentation is to classify pixels into two classes, foreground or background, so we use the F-measure index as our performance metric, as presented in [34, 33]. The higher weighted F-measure means better segmentation performance.

Methods	Grabcut+Weizmann	Gulshan	ECSSD
CC [39]	0.87 (0.06)	x	x
OS [21]	0.88 (0.07)	x	x
Geodesic Superpixel [17]	0.9 (0.09)	0.71(0.14)	0.82(0.11)
Graphcut [6]	0.91 (0.07)	0.65(0.19)	0.85(0.1)
Geodesic Graph-cut [36]	0.91 (0.07)	0.72(0.15)	0.82(0.11)
MSD [10]	0.91 (0.04)	0.72(0.14)	0.8(0.1)
Grabcut [37]	0.93 (0.06)	0.67(0.2)	0.85(0.15)
WSArea [34]	0.93 (0.04)	x	x
WSVol [34]	0.93 (0.04)	x	x
MBD [41]	0.94 (0.05)	0.73(0.18)	0.89(0.07)
Dahu Graph-cut	0.96 (0.04)	0.75 (0.16)	0.90 (0.07)

Table 1: Comparison of various segmentation methods with our Dahu Graph-cut on four different datasets using the F-measure metric (the standard deviation inside the parentheses).

Our method is compared to some related methods such as graph-cut [6], grabcut [37], geodesic distance graph-cut [36], geodesic superpixel [17], MBD [41], MSD [10] and some hierarchical segmentations (CC [39], OS [21], WSArea [27], WSVol [13]). The benchmark results are exposed in Table 1. Our proposed method achieves better results than geodesic distance, MBD methods, outperforms the state-of-the-art graph-cut, and hierarchical segmentation methods.

The segmentation results are illustrated in Fig. 10. Our method is more powerful and provides a smoother contour than the MBD and the geodesic distance. In the case of the geodesic distance, some small errors during the in the case of the Dahu pseudo-distance, the strength of a path only depends on the highest and lowest values along the path, not the path’s length. Our Dahu distance-based method also avoids the short-cutting segmentation in the case of the graph-cut method by using sufficient information from the Dahu distance map.

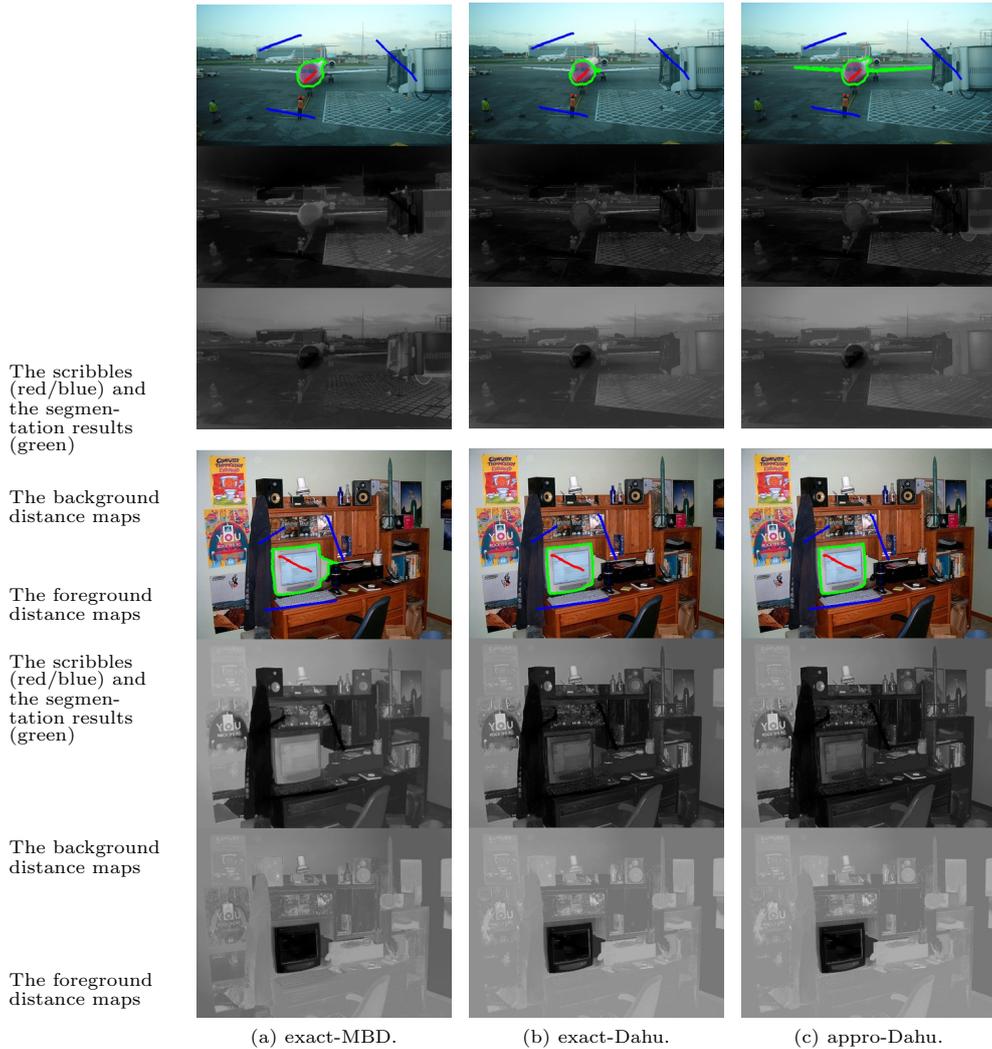


Figure 9: Comparison between the approx/exact-Dahu pseudo-distances and the exact-MBD.

5.3 Ablation study on the components of our proposed method

Our method is a combination of several components such as the Dahu distance map on color, probabilities from GMM-based prior, and graph-cut methods to achieve superior performance compared to other state-of-the-art methods. This subsection conducts an ablation study to analyse the relative contribution of each component to the success of the method. We test the following configurations: (1) Graph-cut [6] without using the Dahu pseudo-distance; (2) Dahu-cut on the color image without using the Graph-cut; (3) Dahu Graph-cut (RGB) without using the probability image; (4) Dahu Graph-cut (P) without using the color image; (5) Dahu Graph-cut (RGB-P) which combines all the components.

Methods	Grabcut+Weizmann	Gulshan	ECSSD
Graph-cut [6]	0.91 (0.07)	0.65(0.19)	0.85(0.1)
Dahu-cut	0.93 (0.07)	0.71(0.16)	0.88(0.1)
Dahu Graph-cut (RGB)	0.93 (0.05)	0.72(0.17)	0.88(0.08)
Dahu Graph-cut (P)	0.94 (0.06)	0.73(0.19)	0.89(0.09)
Dahu Graph-cut (RGB-P)	0.96 (0.04)	0.75 (0.16)	0.90 (0.07)

Table 2: Evaluation of various components of our method (Dahu pseudo-distance, graph-cut, color and probability) on different datasets using the F-measure metric (the standard deviation inside the parentheses).

The quantitative and qualitative results of the ablation study are exposed in Table 2 and Fig. 11. We can see that the regional information from the Dahu pseudo-distance improves significantly the segmentation performance. Indeed, by combining the Dahu pseudo-distance information with the edge modeling of the graph-cut method, our proposed method can prevent the short-cutting segmentation problem. Moreover, by integrating the prior information (probability) from the scribbles into the color image, our proposed method ameliorates the segmentation results. In summary, the contribution

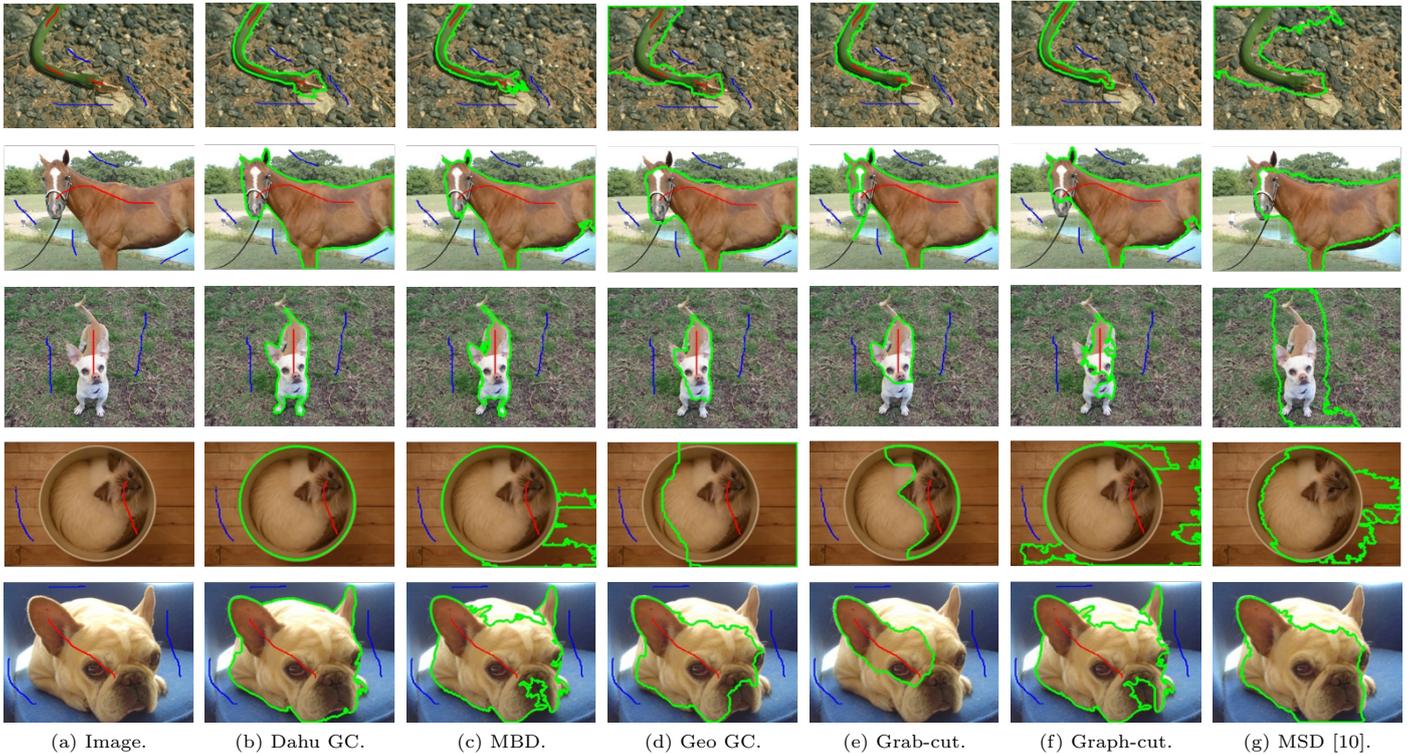


Figure 10: A comparison of the interactive segmentation results of state-of-the-art and our proposed methods.

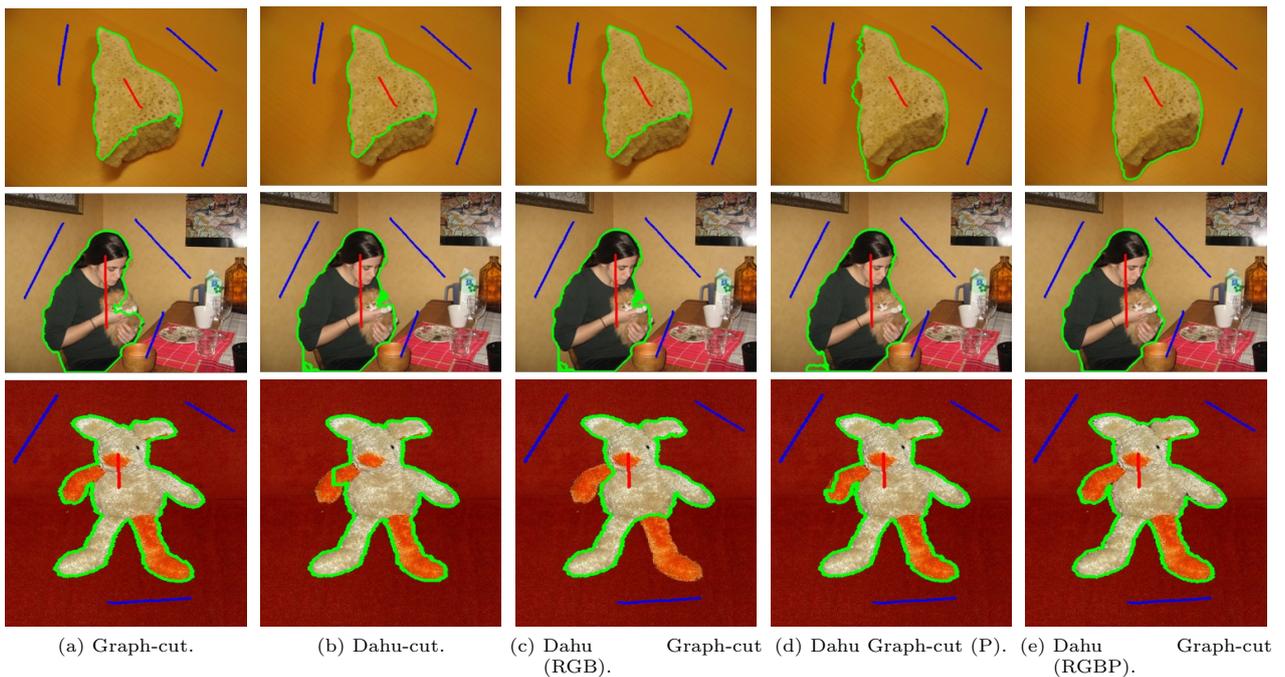


Figure 11: Segmentation results of the Dahu pseudo-distance on the color, probability and combined images. of each component is demonstrated in the final results.

5.4 The Dahu graph-cut on noisy electron microscope images

We present our method on more challenging electron microscope images. The ISBI12 EM Segmentation Challenge [2] is a neuron segmentation challenge that contains 30 noisy 512×512 images. The ground truth provides a collection of labeled connected component for each images. As multi-classes segmentation is out of the scope of our paper, we segment iteratively each region individually in the image. The markers are the points that have the maximum Euclidean distance value w.r.t the boundary of each region. We respectively consider the marker inside of the region as the foreground, and all others as the background. Fig. 12 shows results of our proposed method. (red/blue points are respectively the

foreground/background markers). We also use the F-score metric for the segmentation evaluation. Our method achieves the value of 0.78 for the F-score. The electron microscope images are not only noisy but also the distributions of every region are similar. By using the Dahu pseudo-distance to compute the foreground/background distance, our method provides the regional information to the graph-cut framework to improve the segmentation performances.

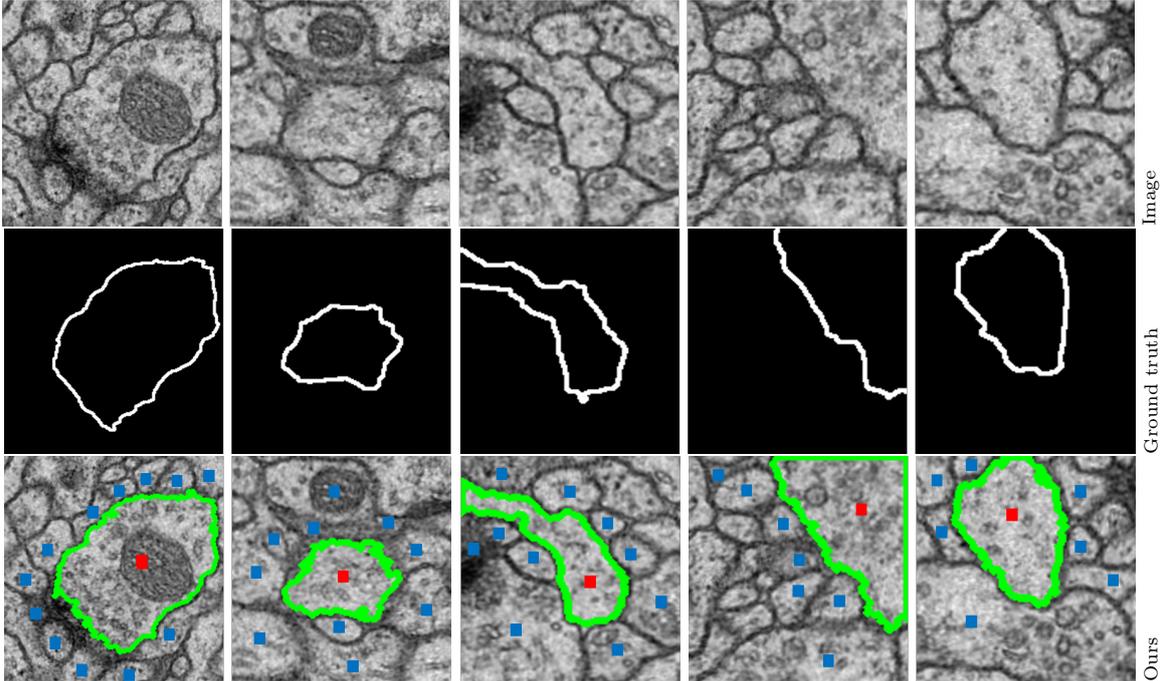


Figure 12: Our interactive segmentation results on Electron Microscope dataset.

5.5 Limitations and Perspectives

We present in Fig. 13 some failure cases of our method. The major challenge of the distance-based approaches is the low contrast. The Dahu pseudo-distance can not distinguish correctly object and background regions if their appearances are too similar. As a consequence, we need to put more markers on the image. For the perspectives, we can integrate the Dahu pseudo-distance in the deep learning model to enhance the interactive segmentation.

6 Experiments on interactive segmentation of 3D images

We test our 3D interactive segmentation method for lung segmentation. Accurate segmentation of the lung plays an important role in the quantitative management of patients. The lung dataset is taken from the COVID-19 3D CT dataset with 20 cases that contain 1800+ annotated slices [24].

We use the DICE metric to evaluate our results. The higher the score is, the better segmentation performance is. Let G , S denote ground truth and segmentation, respectively. The DICE is defined as follows:

$$DICE = \frac{2|G \cap S|}{|G| + |S|} \quad (12)$$

The segmentation performance of our method is compared with the geodesic distance [44], the MBD [12], graph-cut [6], and geodesic graph-cut [36]. The computation of the geodesic distance on 3D images is adapted from [44]. The post processing step is applied on all methods. The result of this comparison is exposed in Table 3. It shows that our method outperforms all the others.

The segmentation results are visualized in Fig. 14 and Fig. 15. Our method achieves satisfying segmentation results, even with one labeled slice, thereby reducing the user interaction. The graph-cut method usually requires more markers from the user to refine the segmentation result. On the other hand, the MBD is not strong enough to correctly segment the object’s edges due to the lack of an explicit edge-finding component. We also illustrate some 3D views of the segmentation result in Fig. 16.

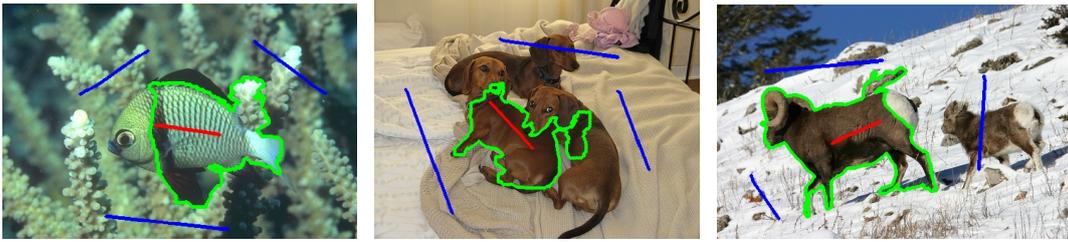


Figure 13: Some failure cases of the Dahu-based approach.

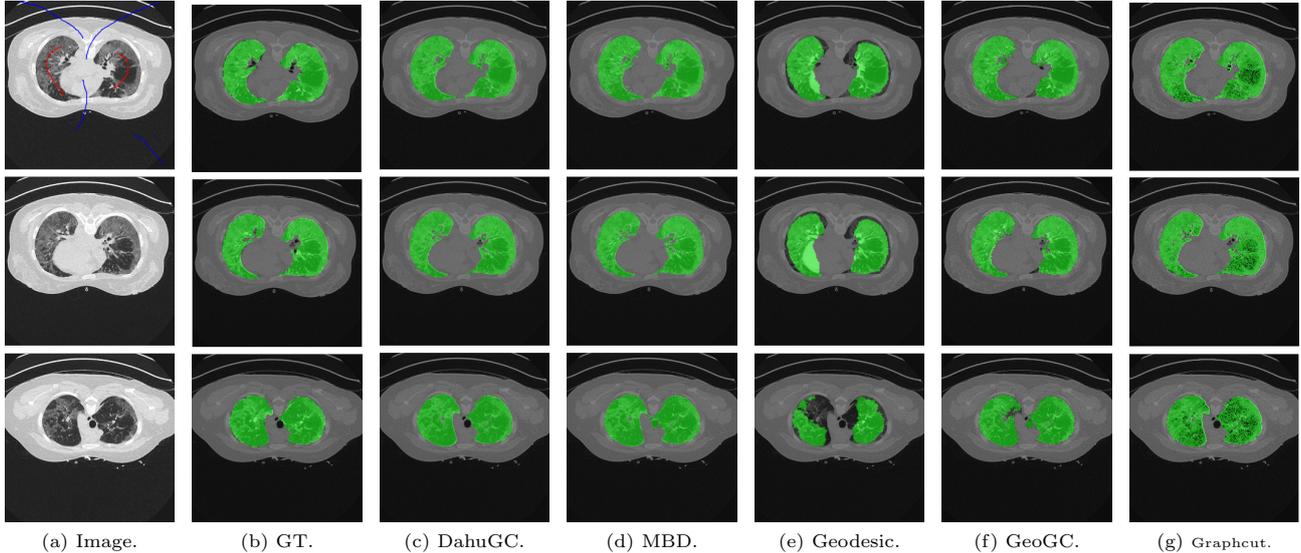


Figure 14: Lung segmentation of patient A.

7 Experiments on image sequences

Video object segmentation aims to segment target objects in every frame spatially. In other words, pixels in a video are grouped into spatio-temporal regions that produce coherence in appearance and motion [20]. Several state-of-the-art methods [7, 42, 26, 32, 16, 3] use different semi-supervised approaches to learn about the object and background. These methods only use the marker on the first frame and segment the object for the rest of the video.

We demonstrate here our framework on the DAVIS video segmentation dataset [31, 35], which consists of 50 videos with their ground truths. markers on every 10 frames to define the background and foreground regions. The markers are generated automatically from the segmentation ground truth by using the morphological skeleton method [9], which is also mentioned in Section 5.2. With these markers, we get an IoU of 70 and a recall of 80. The comparison with other methods is difficult as they use fewer markers but this provides a rough estimation. Furthermore, we have used an automatic generation of markers based on the skeleton of G.T, however, such a kind of marker is maybe not as representative as human scribble. Again, this shows the versatility of our method that can be extended to image sequence augmentation “out-of-the-box” and yet, be competitive with the state-of-the-art dedicated methods.

Our segmentation results are shown in Fig. 17. At first glance, our method achieves quite good results on the videos. Thanks to the Dahu pseudo-distance, the object information is automatically propagated to the following frames. This distance is also effective for grouping pixels in a video into spatio-temporal regions. Our framework can be used on different objects and backgrounds, even when a part of the object is hidden behind the background region (Fig. 17d).

8 Conclusions and Perspectives

We have presented a new scheme for interactive segmentation in 2D images. Our method leverages the advantages of the Dahu pseudo-distance and graph-cut method. We have shown that our scheme is easily adaptable to 3D images, especially 3D medical images. Our scheme is also usable on image sequences, that are considered as 3D images. This allows us to put markers on all frames.

The presentation of this scheme is an opportunity for us to introduce an efficient approach for computing the approximation of the Dahu pseudo-distance directly on the image. Moreover, by combining color and probability images, we

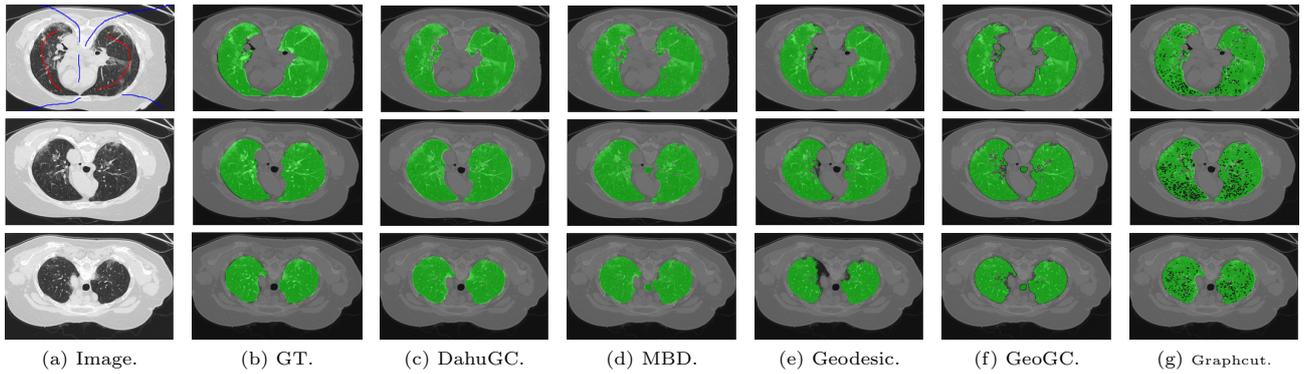


Figure 15: Lung segmentation of patient B.

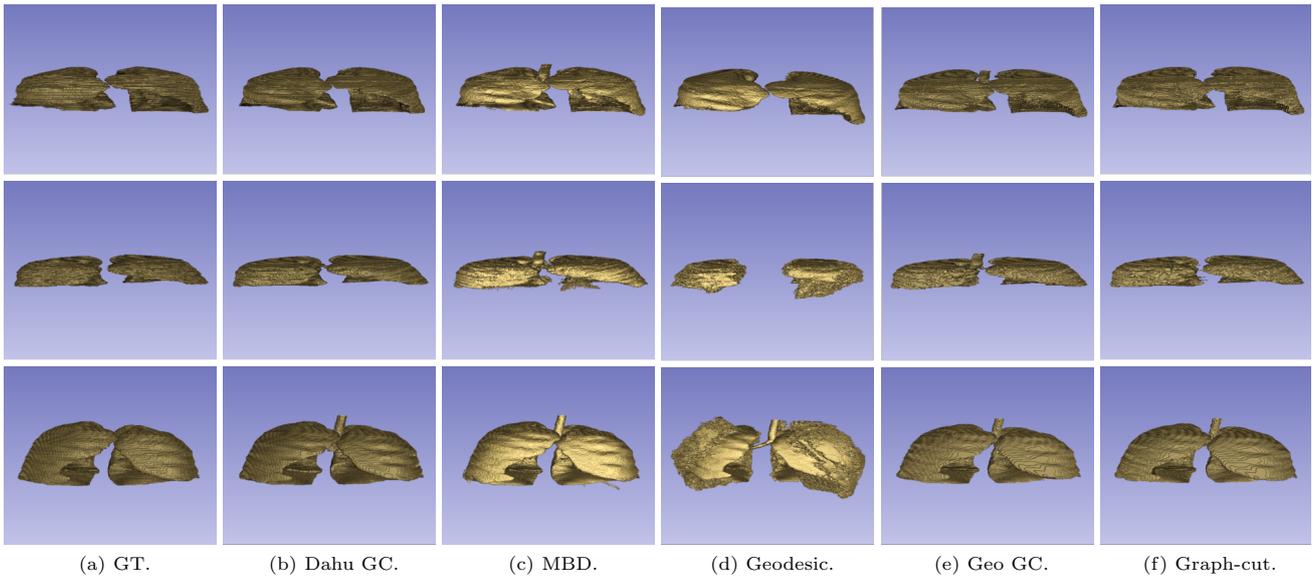


Figure 16: 3D view of the segmentation.

are able to provide good segmentation results. The performance of our proposed method outperforms state-of-the-art approaches.

Thanks to the usage of the Dahu pseudo-distance, which can be computed in n dimensions, we can use our method without problems on 3D images and image sequences. Furthermore, as we are not limited to 3D, we plan to extend our method to 3D video sequences to only need to tag one (or few) slice(s) of the entire sequence (and can be refined as needed). In the future, we also plan to combine this powerful distance with deep learning to improve the ability of this distance in many computer vision applications. Indeed, the Dahu distance map which is computed from the set of markers can be concatenated with the original image as the input of the neural network to provide more information for the task of interactive segmentation.

Acknowledgment

This work has been partially conducted in the context of the MOBIDEM project, part of the “Systematic Paris-Region” and “Images & Network” Clusters (France). This project is partially funded by the French Government and its economic development agencies.

References

- [1] Sharon Alpert, Meirav Galun, Achi Brandt, and Ronen Basri. Image segmentation by probabilistic bottom-up aggregation and cue integration. *Trans. on PAMI*, 34(2):315–327, 2011.
- [2] Ignacio Arganda-Carreras, Srinivas C Turaga, Daniel R Berger, Dan Cireşan, Alessandro Giusti, Luca M Gambardella, Jürgen

Methods	Dice score
Graph-cut [6]	0.86
Geodesic [44]	0.87
MBD [12]	0.95
Geodesic graph-cut [36]	0.95
Our method	0.96

Table 3: Quantitative results on the lung dataset (the COVID-19 3D CT dataset [24]).

Schmidhuber, Dmitry Laptev, Sarvesh Dwivedi, Joachim M Buhmann, et al. Crowdsourcing the creation of image segmentation algorithms for connectomics. *Frontiers in neuroanatomy*, page 142, 2015.

- [3] S Avinash Ramakanth and R Venkatesh Babu. Seamseg: Video object segmentation using patch seams. In *Proc. of CVPR*, pages 376–383, 2014.
- [4] Xue Bai and Guillermo Sapiro. A geodesic framework for fast interactive image and video segmentation and matting. In *Proc. of ICCV*, pages 1–8. IEEE, 2007.
- [5] Andrew Blake, Carsten Rother, Matthew Brown, Patrick Perez, and Philip Torr. Interactive image segmentation using an adaptive gmmrf model. In *Proc. of ECCV*, pages 428–441. Springer, 2004.
- [6] Yuri Y Boykov and M-P Jolly. Interactive graph cuts for optimal boundary & region segmentation of objects in nd images. In *Proc. of ICCV*, volume 1, pages 105–112. IEEE, 2001.
- [7] Sergi Caelles, Kevis-Kokitsi Maninis, Jordi Pont-Tuset, Laura Leal-Taixé, Daniel Cremers, and Luc Van Gool. One-shot video object segmentation. In *Proc. of CVPR*, pages 221–230, 2017.
- [8] Edwin Carlinet and Thierry Géraud. Morphological object picking based on the color tree of shapes. In *2015 International Conference on Image Processing Theory, Tools and Applications (IPTA)*, pages 125–130. IEEE, 2015.
- [9] John Chaussard, Michel Couprie, and Hugues Talbot. Robust skeletonization using the discrete λ -medial axis. *Pattern Recognition Letters*, 2011.
- [10] Chao-Te Chou, Wei-Chih Tu, and Shao-Yi Chien. Minimum spanning distance for image segmentation. In *Proc. of ICASSP*. IEEE, 2018.
- [11] Yung-Yu Chuang, Brian Curless, David H Salesin, and Richard Szeliski. A bayesian approach to digital matting. In *Proc. of CVPR*, volume 2, pages II–II. IEEE, 2001.
- [12] Krzysztof Chris Ciesielski et al. Efficient algorithm for finding the exact minimum barrier distance. *Journal of CVIU*, 123:53–64, 2014.
- [13] Jean Cousty and Laurent Najman. Incremental algorithm for hierarchical minimum spanning forests and saliency of watershed cuts. In *Proc. of ISMM*, pages 272–283. Springer, 2011.
- [14] Antonio Criminisi, Toby Sharp, and Andrew Blake. Geos: Geodesic image segmentation. In *Proc. of ECCV*, pages 99–112. Springer, 2008.
- [15] Anastasia Dubrovina, Rom Hershkovitz, and Ron Kimmel. Image editing using level set trees. In *Proc. of ICIP*. IEEE, 2014.
- [16] Qingnan Fan, Fan Zhong, Dani Lischinski, Daniel Cohen-Or, and Baoquan Chen. Jumpcut: non-successive mask transfer and interpolation for video cutout. *ACM Trans. Graph.*, 34(6):195–1, 2015.
- [17] Jie Feng, Brian Price, Scott Cohen, and Shih-Fu Chang. Interactive segmentation on rgbd images via cue selection. In *Proc. of CVPR*, pages 156–164, 2016.
- [18] T. Géraud, E. Carlinet, S. Crozet, and L. Najman. A quasi-linear algorithm to compute the tree of shapes of n -D images. In *Proc. of ISMM*. Springer, 2013.
- [19] Thierry Géraud, Yongchao Xu, Edwin Carlinet, and Nicolas Boutry. Introducing the Dahu pseudo-distance. In *Proc. of ISMM*. Springer, 2017.
- [20] Matthias Grundmann, Vivek Kwatra, Mei Han, and Irfan Essa. Efficient hierarchical graph-based video segmentation. In *Proc. of CVPR*, pages 2141–2148. IEEE, 2010.
- [21] Silvio Jamil F Guimarães, Jean Cousty, Yukiko Kenmochi, and Laurent Najman. A hierarchical image segmentation algorithm based on an observation scale. In *International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition*. Springer, 2012.

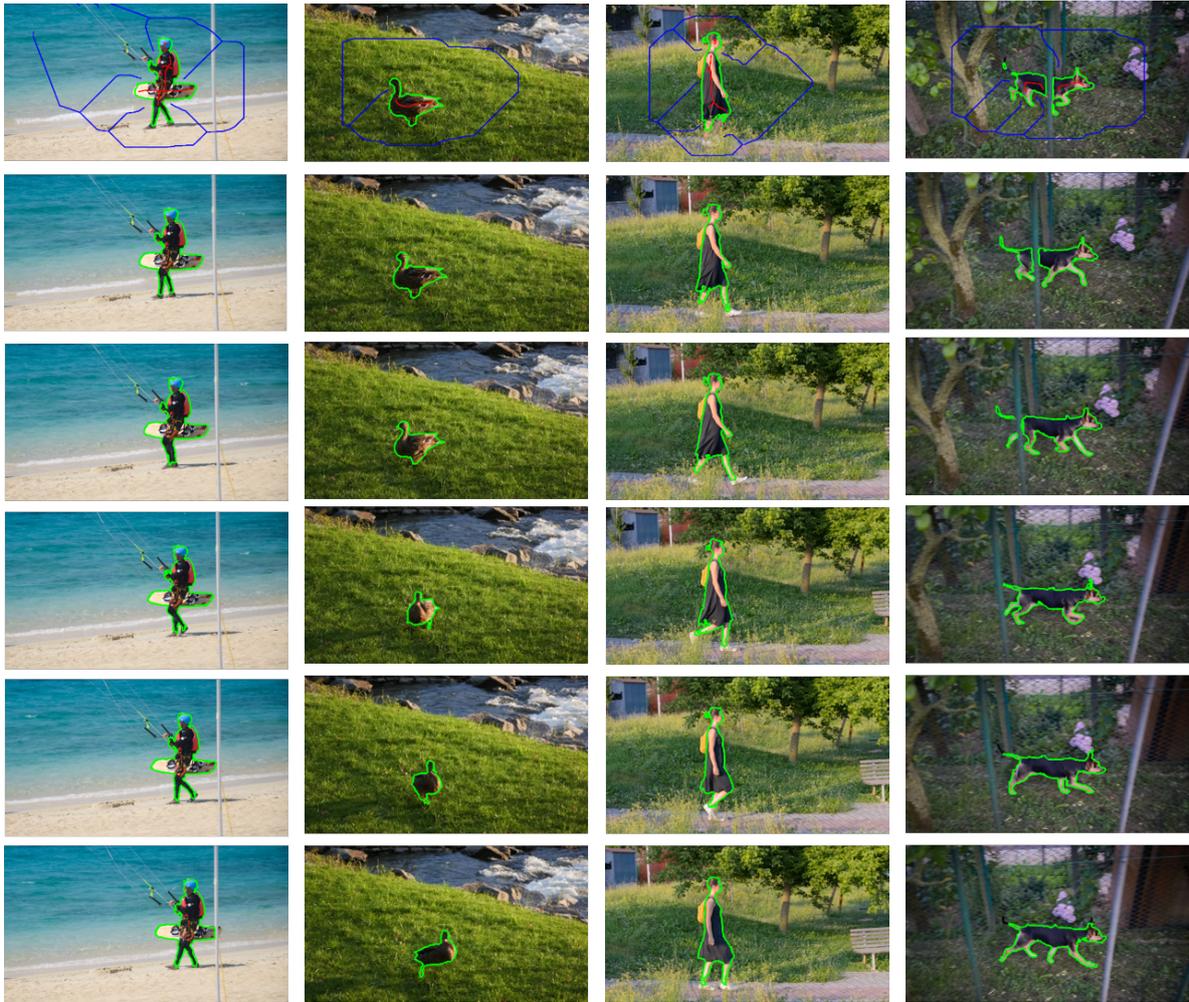


Figure 17: Examples of video segmentation. The scribbles (in blue and red) are given in the frame of the video and the algorithm automatically segments the remained frame.

- [22] Varun Gulshan, Carsten Rother, Antonio Criminisi, Andrew Blake, and Andrew Zisserman. Geodesic star convexity for interactive image segmentation. In *Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE, 2010.
- [23] Xiaoming Huang and Yujin Zhang. Water flow driven salient object detection at 180 FPS. *Pattern Recognition Letters*, 76:95–107, 2018.
- [24] Ma Jun, G Cheng, W Yixin, A Xingle, G Jiantao, Y Ziqi, Z Minqing, L Xin, D Xueyuan, C Shucheng, et al. Covid-19 ct lung and infection segmentation dataset. *Zenodo*, Apr, 20, 2020.
- [25] Michael Kass, Andrew Witkin, and Demetri Terzopoulos. Snakes: Active contour models. *International Journal on Computer Vision (IJCV)*, 1(4):321–331, 1988.
- [26] Nicolas Märki, Federico Perazzi, Oliver Wang, and Alexander Sorkine-Hornung. Bilateral space video segmentation. In *Proc. of CVPR*, pages 743–751, 2016.
- [27] Laurent Najman and Michel Schmitt. Geodesic saliency of watershed contours and hierarchical segmentation. *Trans. on PAMI*, 18(12):1163–1173, 1996.
- [28] Minh Ôn Vũ Ngoc, Nicolas Boutry, Jonathan Fabrizio, and Thierry Géraud. A minimum barrier distance for multivariate images with applications. *Computer Vision and Image Understanding (CVIU)*, page 102993, 2020.
- [29] Minh Ôn Vũ Ngoc, Jonathan Fabrizio, and Thierry Géraud. Saliency-based detection of identity documents captured by smartphones. In *Processing of the IAPR International Workshop on Document Analysis Systems (DAS)*, pages 387–392, 2018.
- [30] Minh Ôn Vũ Ngoc, Jonathan Fabrizio, and Thierry Géraud. Document detection in videos captured by smartphones using a saliency-based method. In *Proc. of ICDARW*. IEEE, 2019.

- [31] Federico Perazzi, Jordi Pont-Tuset, Brian McWilliams, Luc Van Gool, Markus Gross, and Alexander Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 724–732, 2016.
- [32] Federico Perazzi, Oliver Wang, Markus Gross, and Alexander Sorkine-Hornung. Fully connected object proposals for video segmentation. In *Proc. of ICCV*, pages 3227–3234, 2015.
- [33] Benjamin Perret, Jean Cousty, Silvio Jamil F Guimaraes, and Deise S Maia. Evaluation of hierarchical watersheds. *Trans. on TIP*, 2017.
- [34] Benjamin Perret, Jean Cousty, Jean Carlo Rivera Ura, and Silvio Jamil F Guimarães. Evaluation of morphological hierarchies for supervised segmentation. In *Proc. of ISMM*. Springer, 2015.
- [35] Jordi Pont-Tuset, Federico Perazzi, Sergi Caelles, Pablo Arbeláez, Alex Sorkine-Hornung, and Luc Van Gool. The 2017 davis challenge on video object segmentation. *arXiv preprint arXiv:1704.00675*, 2017.
- [36] Brian L Price, Bryan Morse, and Scott Cohen. Geodesic graph cut for interactive image segmentation. In *Proc. of CVPR*. IEEE, 2010.
- [37] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. "grabcut" interactive foreground extraction using iterated graph cuts. *ACM transactions on graphics (TOG)*, 2004.
- [38] Jianping Shi, Qiong Yan, Li Xu, and Jiaya Jia. Hierarchical image saliency detection on extended cssd. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(4):717–729, 2015.
- [39] Pierre Soille. Constrained connectivity for hierarchical image partitioning and simplification. *Trans. on PAMI*, 30(7):1132–1145, 2008.
- [40] Robin Strand et al. The minimum barrier distance. *Journal of CVIU*, 2013.
- [41] Robin Strand et al. The minimum barrier distance: A summary of recent advances. In *Proceedings of the International Conference on Discrete Geometry for Computer Imagery (DGCI)*, volume 10502 of *LNCS*, pages 57–68. Springer, 2017.
- [42] Yi-Hsuan Tsai, Ming-Hsuan Yang, and Michael J Black. Video segmentation via object flow. In *Proc. of CVPR*, pages 3899–3908, 2016.
- [43] Wei-Chih Tu, Shengfeng He, Qingxiong Yang, and Shao-Yi Chien. Real-time salient object detection with a minimum spanning tree. In *Proc. of CVPR*, 2016.
- [44] Guotai Wang, Maria A Zuluaga, Wenqi Li, Rosalind Pratt, Premal A Patel, Michael Aertsen, Tom Doel, Anna L David, Jan Deprest, Sébastien Ourselin, et al. Deepigeos: a deep interactive geodesic framework for medical image segmentation. *Trans. on PAMI*, 2018.
- [45] Jianming Zhang, Stan Sclaroff, Zhe Lin, Xiaohui Shen, Brian Price, and Radomir Mech. Minimum barrier salient object detection at 80 FPS. In *Proc. of ICCV*, 2015.

Minh Ôn Vũ Ngọc received the Ing. Degree from HCMUT (VietNam) in 2015, the M.Sc degree in Signal and Image Processing from IMT Atlantique-Bretagne, in 2016 and the Ph.D. degree in Computer Science from Sorbonne Université, Paris in 2020. He is currently working at EPITA, Paris. His research interests include Image Processing and Machine learning.

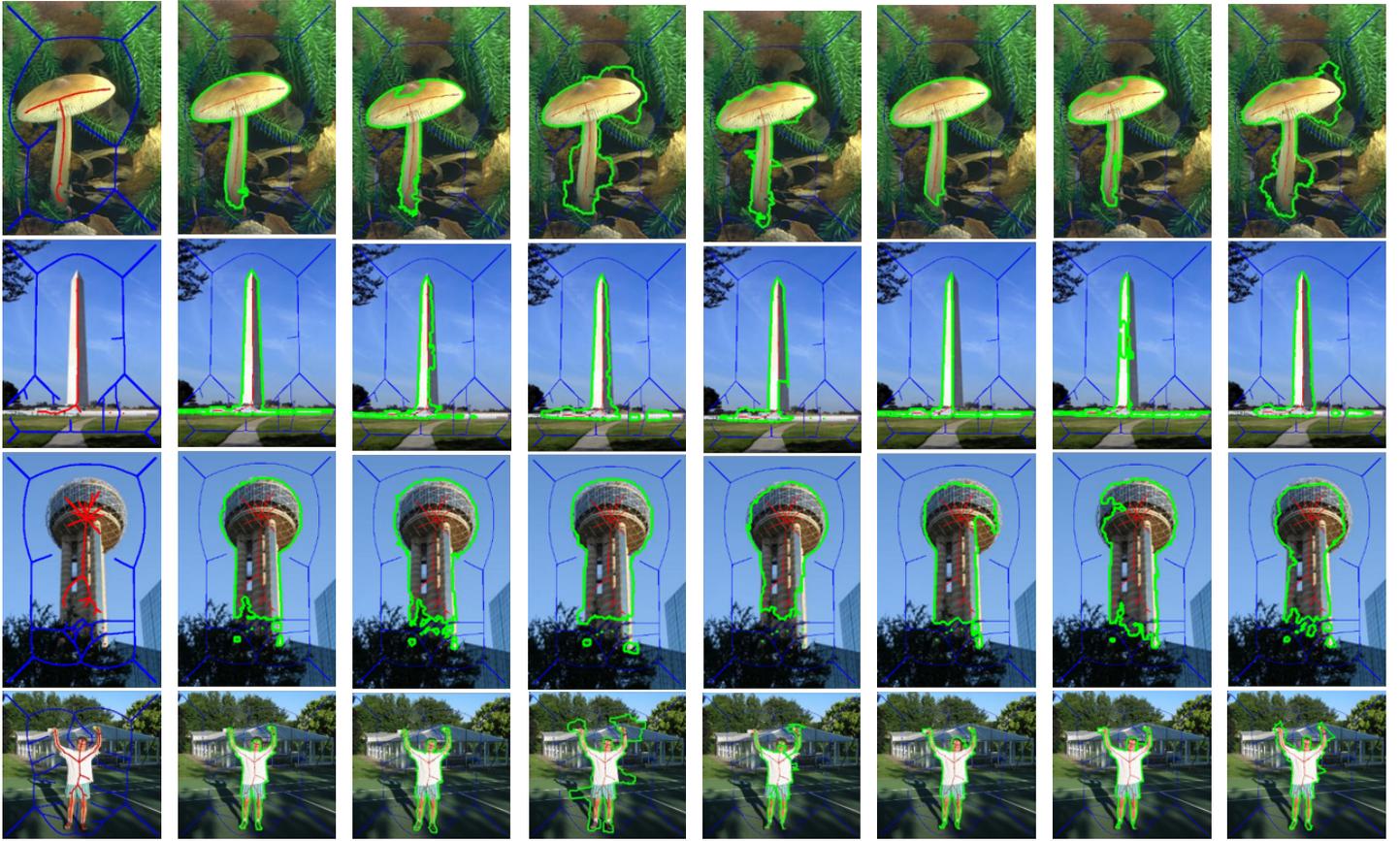
Edwin Carlinet received the Ing. degree from EPITA, the MVA M.Sc degree from ENS-Cachan and the Ph.D. degree from Univ. Paris-Est in 2015. He is currently a lecturer at EPITA. His research interests include high performance computing, in particular, the optimization of IP and Mathematical Morphology related algorithms.

Jonathan Fabrizio received the PhD degree from Université Pierre et Marie Curie - Paris VI in 2004. He is currently assistant-professor at EPITA Research Laboratory (LRE), Paris, France. His research interests include image processing and pattern recognition.

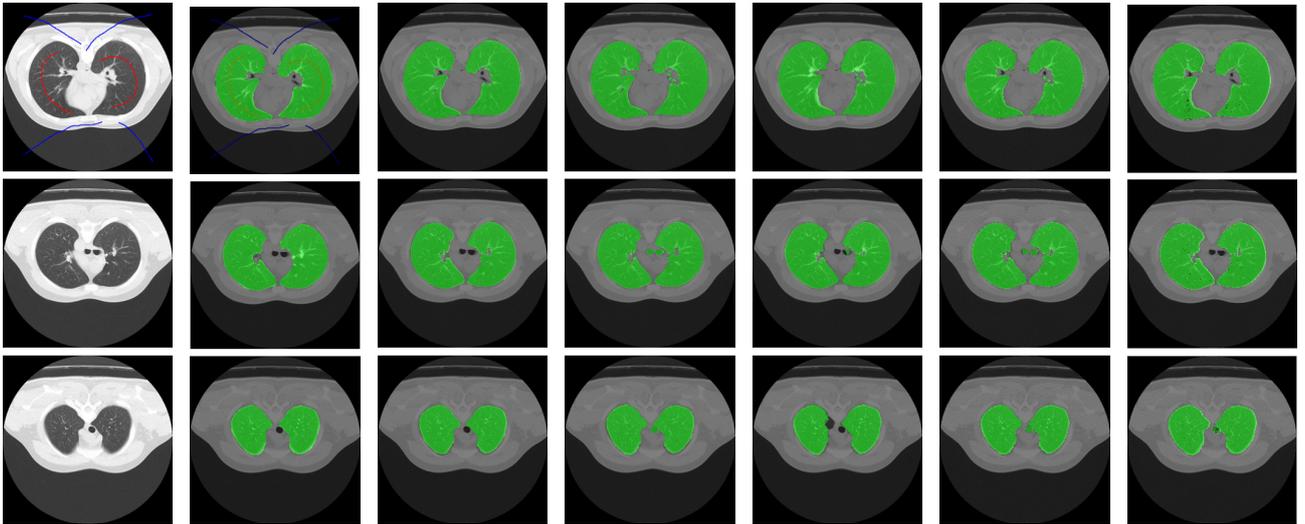
Thierry Géraud received the PhD degree in signal and image processing from Télécom ParisTech, in 1997, and the Habilitation à Diriger les Recherches from the Université Paris-Est, in 2012. He is one of the main authors of the Olena platform, dedicated to image processing and available as free software under the GPL licence. His research interests include image processing, pattern recognition, software engineering, and object-oriented scientific computing. He is currently working with EPITA Research Laboratory (LRE), Paris, France.

A Additional results

We provide here more results of our algorithm on 2D and 3D images (Fig. 18).



(a) Image. (b) Dahu GC. (c) MBD. (d) Geo GC. (e) Geo SP. (f) Grab-cut. (g) Graphcut. (h) MSD.



(i) Image. (j) GT. (k) DahuGC. (l) MBD. (m) Geodesic. (n) GeoGC. (o) Graphcut.

Figure 18: Interactive segmentation results on Grabcut and Weizmann dataset (top) and Lung segmentation of patient C. (bottom)