Author Manuscript
Author Manuscript
Author Manuscript
Author Manuscript

HHS Public Access
Author manuscript
IEEE Trans Image Process. Author manuscript; available in PMC 2018 April 06.

Published in final edited form as:
IEEE Trans Image Process. 2016 December ; 25(12): 5857–5866. doi:10.1109/TIP.2016.2615809.

Dynamic Programming Using Polar Variance for Image Segmentation

José A. Rosado-Toro [Student Member, IEEE],
Electrical and Computer Engineering, University of Arizona, Tucson, AZ 85721 (jrosado@email.arizona.edu)

María I. Altbach, and
Department of Medical Imaging, University of Arizona, Tucson, AZ 85724 (maltbach@radiology.arizona.edu)

Jeffrey J. Rodríguez [Senior Member, IEEE]
Department of Electrical and Computer Engineering, University of Arizona, Tucson, AZ 85721 (jjrodrig@email.arizona.edu)

Abstract

When using polar dynamic programming (PDP) for image segmentation, the object size is one of the main features used. This is because if size is left unconstrained the final segmentation may include high-gradient regions that are not associated with the object. In this paper, we propose a new feature, polar variance, which allows the algorithm to segment objects of different sizes without the need for training data. The polar variance is the variance in a polar region between a user-selected origin and a pixel we want to analyze. We also incorporate a new technique that allows PDP to segment complex shapes by finding low-gradient regions and growing them. The experimental analysis consisted on comparing our technique with different active contour segmentation techniques on a series of tests. The tests consisted on robustness to additive Gaussian noise, segmentation accuracy with different grayscale images and finally robustness to algorithm-specific parameters. Experimental results show that our technique performs favorably when compared to other segmentation techniques.

Keywords

dynamic programming; image segmentation; polar variance; region growing

I. INTRODUCTION

Automated object segmentation is a necessary step in many image analysis applications. In this manuscript we propose an improved method for extracting a closed contour of an object based on a single user-defined seed point within the object. Kass et al. [1] developed a segmentation technique known as active contours or snakes, which uses external forces such as edge strength and internal forces such as elasticity to evolve an initial contour to a minimum energy state. One of the problems with the technique is its inability to segment complex objects [2]. To address this issue, several models have been developed. Among them are gradient vector flow models [2][3][4][5][6], region-based models[7][8][9][10][11]

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

[12], and level-set models [13][14][15][16][17][18][19]. Although these algorithms can segment complex shapes, the evolution of the curves is constrained by the number of iterations. If the number of iterations is too small, the contour may not evolve to its optimum state, whereas if it is too large and it does not converge quickly the algorithm may take a long time to run.

Another approach for segmenting objects is dynamic programming [20]. Dynamic programming has been used for a series of applications including outlining pectoral muscle boundaries [21], resizing images [22], creating superpixels [23], and generating a closed contour [24]-[26]. To generate a closed contour, Sun and Pallotino [24], Timp and Karssemeijer [25] and Zhang et al. [26] transformed the image space from Cartesian to polar coordinates. By finding a path, as defined by [20], in polar coordinates, they were able to segment features in medical images, such as masses in mammograms [25] and bladders in MR images [26] without needing an iteration parameter. Unfortunately, the algorithm outlined in [24]-[26] consists of two major drawbacks that prevent it from being used as a general-purpose algorithm. First, it requires training data to constrain the size of the object being segmented. The size constraint helps prevent the closed contour from latching onto high-gradient regions that are not associated with the object. The second major drawback is that this algorithm cannot segment shapes for which rays emanating from the selected origin within the shape intersect the shape boundary in more than one point, for example a hand. To allow the algorithm to segment different-sized objects without needing training data, we introduce a preprocessing step where a polar variance image is calculated from the image (Section II-A). To segment complex shapes using the polar representation of the image, techniques such as fast marching techniques [27][28], PDEs [29], or active contours [30] are used to find a path. In this manuscript, we develop a technique that uses dynamic programming to segment complex shapes by growing low-gradient regions of the contour (Section II-E). We provide experimental results comparing our algorithm, which we call polar dynamic programming (PDP), with other state-of-the-art algorithms on a variety of test images.

II. PROPOSED METHOD

The proposed PDP algorithm begins with a simple PDP (SPDP) method to generate a closed contour representing a simple object shape. The first step of the SPDP algorithm is to preprocess the input image to obtain a *polar variance image*. The second step is to calculate the cost function needed by the dynamic programming algorithm. This includes generating the cost function in Cartesian space and later resampling to a uniform polar grid (U_p). The third step is to use dynamic programming to generate a closed contour representing the object shape. The PDP algorithm then extends this contour by applying a localized region growing technique for segmenting complex shapes.

A. Polar Variance Image

To obtain closed contours using dynamic programming, Timp [25] and Zhang [26] had to constrain the size of the final segmentation by using training data. They constrained the size because the closed contour may latch onto high-gradient regions that are not associated with

Author Manuscript

the object. Requiring training data to constrain the size of the final segmentation reduces the generalization of the algorithm; therefore, we introduce a preprocessing step to generate what we define as the polar variance image.

Given an image I, the first step is to normalize the intensities from 0 to 1. We then define each pixel in I as $I(x, y)$ where (x, y) are the coordinates in the uniform grid $G(x, y)$ and is the edge strength of $I(x, y)$. The edge strength with no Gaussian smoothing is generally calculated as

$$G(x,y) = |\nabla I(x,y)|^2, \quad (1)$$

whereas with Gaussian smoothing the edge strength is calculated as

$$G(x,y) = |\nabla\left[G_\sigma(x,y) * I(x,y)\right]|^2. \quad (2)$$

Here, $G_\sigma(x, y)$ is a 2-D Gaussian function with standard deviation σ, * denotes linear convolution, and ∇ is the gradient operator [1]. We assume that $I(x, y)$ is scalar (i.e., a grayscale image), but the derivation can be generalized for higher dimensionality. Introducing an origin (x_o, y_o) allows us to represent $I(x, y)$ and $G(x, y)$ as $f(\theta, \rho)$ and in polar coordinates as follows:

$$\rho = \sqrt{\left(x - x_o\right)^2 + \left(y - y_o\right)^2}, \quad (3)$$

$$\theta = \tan^{-1}\left(\frac{y - y_o}{x - x_o}\right), \quad (4)$$

$$f(\theta,\rho) = I(x,y), \quad (5)$$

$$g(\theta,\rho) = G(x,y). \quad (6)$$

Using the polar representation, we define the polar variance image as

$$V(\theta, \rho) = \int_0^{\rho} \int_{\theta - \gamma}^{\theta + \gamma} \left(f(\phi, r) - \overline{f}(\phi, r)\right)^2 dr\, d\phi, \quad (7)$$

where

$$\overline{f}(\theta, \rho) = \int_0^{\rho} \int_{\theta - \gamma}^{\theta + \gamma} f(\phi, r) dr\, d\phi\,. \quad (8)$$

In these equations, γ is a smoothing parameter. The higher γ is, the more smoothed the polar variance image will be. Note that the polar variance image is cumulative because the integral spans radii from the origin to the pixel being analyzed. Although variance as a feature has been used for segmentation [8] and as a preprocessing technique [27][32], to the best of our knowledge ours is the first segmentation algorithm that calculates a cumulative variance based on polar region coordinates. Fig. 1 shows the results of the polar variance image on two different scenarios. The short-axis cardiac MR image, shown in Fig. 1(a), contains high-gradient regions that are not part of the right ventricle (i.e. the object with the dot). Fig. 1(b) shows an image where rays emanating from an origin (the dot) will intersect the hand boundary in more than one point. Fig. 1(c) shows the polar variance image of Fig. 1(a). Note that the right ventricle has low variance values, whereas high gradient regions that are not associated with the right ventricle have high variance values. Fig. 1(d) shows the polar variance image of Fig. 1(b). Note that the polar variance image does not have low values in the entire hand boundary just in the regions where the rays intersect the shape boundary at only one point.

B. Cost Function

The gradient and variance values are the features that will be optimized using dynamic programming. As a general rule, we want the variance values inside the closed contour to be low, and we want the gradient values along the contour to be high. Therefore, we define the cost function as

$$C(\theta, \rho) = \alpha V(\theta, \rho) + \beta \exp(-g(\theta, \rho)), \quad (9)$$

where α and β are parameters for determining the importance of each feature.

1) Polar Resampling—To use dynamic programming as defined by [20], we require a uniform grid. Due to the discrete nature of I, its polar representation is not uniformly sampled. Therefore, we resample on a uniform polar grid (U_p), which is the lattice comprising points $(\theta_i, \rho_j) = (i\Delta\theta, j\Delta\rho)$ that span the image domain. We derive the sampling intervals ($\Delta\rho$ and $\Delta\theta$) to ensure that the polar grid has sufficient resolution to localize each pixel in the rectangular grid. First, we find the (x, y) coordinates that are farthest from the origin (x_o, y_o). We consider a neighboring pixel (e.g., this could be the pixel at (x - 1, y) if the farthest point is at the upper-right corner). As a general rule, the smaller the angular sampling interval ($\Delta\theta$), the bigger the maximum allowed curvature of the extracted contour. To find the minimum $\Delta\theta$ required to localize every pixel in the image, we find the angular difference between the two neighboring pixels farthest from the origin:

$$\Delta\theta_{min} = \left| \tan^{-1}\left(\frac{y - y_o}{x - x_o}\right) - \tan^{-1}\left(\frac{y - y_o}{x - 1 - x_o}\right) \right|. \quad (10)$$

The $\Delta\theta$ values used for analysis are multiples of $\Delta\theta_{min}$.

The minimum absolute radial difference between pixels with the same angular value is 1 pixel. Therefore a $\Delta\rho \leq 1$ can be used to localize every pixel in the image. For our experiments we used $\Delta\rho = 0.5$.

Unlike active contours, our cost function (9) does not explicitly minimize the curvature of the extracted contour. Instead, the curvature of the contour of the segmented object is based on the value of we select. Fig. 2 shows how the value affects the curvature of the contour.

2) Interpolation—For each (θ_i, ρ_j) ϵ U_P point we compute the corresponding Cartesian coordinates as

$$x = \rho_j \cos(\theta_i), \quad (11)$$

$$y = \rho_j \sin(\theta_i). \quad (12)$$

Then we use bilinear interpolation [33] to generate the cost function ($C_P(\theta_i, \rho_j)$) from the cost function values at the four pixels that surround the point (x, y) in the Cartesian space.

C. Dynamic Programming

After generating a uniformly sampled polar representation of the cost function (C_P), the next step is to use dynamic programming to compute the object contour by finding a path, as defined by [20]. This path, denoted as $\rho_{path}(\theta_i)$, is a function of θ_i because the path will pass through only one radius value for each θ_i. Also, the absolute radial difference between adjacent angles will be $\Delta\rho$ or 0. The general approach is to find the radii that minimize the cumulative sum of the cost function (C_p). To obtain a closed contour, the ρ_{path} value associated with $\theta_i = 0°$ and $\theta_i = 360°$ must be the same. To add this constraint to the dynamic programming, we use the image patching algorithm (IPA). This technique, coined by Sun and Pallotino [24], consists of extending θ_i in a periodic fashion. In their work, Timp and Karssemeijer extends θ_i from [−180°, 180°] to [−540°, 540°]. For these experiments we

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

define our array from [0°, 360°] and in turn extend to [−360°, 720°]. This extended version of the cost function is referred as C_p^*.

Given C_p^* we initialize the cumulative sum M_p as follows. To initialize we let $M_p(-360°, \rho_j) = C_p^*(-360°, \rho_j)$ for each ρ_j. The other M_p values are recursively computed:

$$M_p(\theta_i, \rho_j) = \min\{M_p(\theta_i - \Delta\theta, \rho_j - \Delta\rho), M_p(\theta_i - \Delta\theta, \rho_j), M_p(\theta_i - \Delta\theta, \rho_j + \Delta\rho)\} + C_p^*(\theta_i, \rho_j), \quad (13)$$

for each $\theta_i \in [-360° + \Delta\theta, 720°]$ and for each ρ_j.

Once M_p has been generated, ρ_{path} is initialized as follows:

$$\rho_{path}(720°) = \underset{\rho}{\operatorname{argmin}} M_p(720°, \rho). \quad (14)$$

The rest of the path, for $\theta_i \in [-360°, 720° - \Delta\theta]$, is created as follows:

$$\rho_{path}(\theta_i) = \underset{\rho \in \{\rho_{path}(\theta_i + \Delta\theta), \rho_{path}(\theta_i + \Delta\theta) + \Delta\rho, \rho_{path}(\theta_i + \Delta\theta) - \Delta\rho\}}{\operatorname{argmin}} M_p(\theta_i, \rho). \quad (15)$$

Note that ρ_{path} is defined for $\theta_i \in [-360°, 720°]$, just like M_P. To generate the closed contour, we only keep ρ_{path} for $\theta_i \in [0°, 360° - \Delta\theta]$. To convert to the Cartesian coordinate system, we use (11) and (12) to convert ρ_{path} and θ_i to x and y.

One of the drawbacks of the IPA is that a closed path is not guaranteed[24][25]. Timp and Karssemeijer noted that using IPA generated a closed contour on 98 percent of the data they used for analysis. To ensure a closed contour we use Bresenham's line algorithm [34] to join the (x, y) coordinates associated with $\rho_{path}(0°)$ and $\rho_{path}(360° - \Delta\theta)$. This step allowed our technique to generate a closed contour on all cases.

D. Optimizing for Multiple Values

In case a single $\Delta\theta$ does not suffice, we generate a closed contour using multiple $\Delta\theta$ values (i.e., multiple curvature values), which are multiples of $\Delta\theta_{min}$, as discussed in Section II-B. 1. We select the $\Delta\theta$ value that optimizes a cost function as follows:

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

$$\Delta\theta^* = \underset{\Delta\theta}{\operatorname{argmin}}\left(\alpha\int_{\Omega_1} V(x,y)dxdy + \beta\int_{\Omega_2}\exp(g(x,y))dxdy\right) \tag{16}$$

where Ω_1 is the interior of the extracted closed contour and Ω_2 is the extracted closed contour. We define the union of Ω_1 and Ω_2 as the mask, $\Omega = \Omega_1 \cup \Omega_2$.

E. Localized Region Growing

The simple PDP algorithm (SPDP) presented in Sections II-A to II-D can segment simple shapes for which rays emanating from an origin within the shape intersect the shape boundary in one point without the need for training data. However, this technique may generate undersized contours in the case of more complex shapes. To segment complex shapes, we grow the extracted closed contour (Ω_2) using the localized region growing procedure shown in Fig. 3. The procedure begins by localizing weak sub-contours (Fig. 3, line 2). A weak sub-contour is a region in Ω_2 that has a gradient magnitude less than or equal to the mean gradient magnitude in Ω_1. A size constraint can be used to potentially reduce the number of analyzed weak sub-contours. For our experiments we do not use a size constraint. For each weak sub-contour, we run SPDP with a new origin located on the weak sub-contour to produce a temporary mask (Ω^{temp}) (Fig. 3, lines 5-7). This temporary mask is refined to include regions that do not overlap with Ω, thus forming a new mask (Ω^{new}) (Fig. 3, lines 8-13). If the total length of the weak sub-contours in the new mask is less than the length of the weak sub-contour we are growing, we keep the new mask (Fig. 3, lines 14-18). The portion of the weak sub-contour that overlaps with the new mask is marked as closed (Fig. 3, line 19-20). Finally, new weak sub-contours are added (Fig. 3, line 23). This is repeated until all the weak sub-contours have been examined. Fig. 4 shows an example where the localized region growing is used to grow the closed contour to include the thumb.

F. Complexity Analysis

The complexity of the algorithm is expressed as $O\left(r\left(n^2 + \alpha(3vu)\right)\right)$. n is the number of elements in the image, u and v are the height and width of the polar representation of the cost function that is used with dynamic programming. $u = \left(\frac{\rho_{max}}{\Delta\rho} + 1\right)$, where ρ_{max} is the Euclidean distance between the origin and the farthest point in the image; $v = \left(\frac{3(360°)}{\Delta\theta} + 1\right)$ (see Section II-C). α are the number of angles used (Section II-D) and r are the number of low-gradient regions that will be grown (Section II-F). $O(n^2)$ is the complexity associated with calculating the variance image, and $O(3vu)$ is the complexity associated with finding a path using IPA. Note that in the case where $\alpha > 1$, the upper bound complexity can be found by using the widest v (i.e., when $\Delta\theta = \Delta\theta_{min}$).

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

III. RESULTS AND ANALYSIS

We compare the PDP algorithm to several active contour segmentation algorithms [5][6],[8],[12],[19],[27] using a variety of tests. The first test measures the robustness to additive Gaussian noise. The second test measures segmentation accuracy using different grayscale images. The third test shows the robustness to algorithm-specific parameters. The final test tests the robustness of PDP to changes in origin selection.

The algorithms' performance is measured using the Dice metric [35] and the average border positioning error (BPE) [36]. The Dice metric finds the mutual overlap between the mask generated by the automated algorithm and the mask generated by manual segmentation (i.e., ground truth). The average BPE gives the mean difference (in pixels) between the automated and manual contours. An automated mask that perfectly matches the manual mask will yield an average Dice value of 1 and an average BPE of 0 pixels.

The images used for these experiments are roughly100×100. The experiments were carried out in a MacBook Pro with 2.4 GHz Intel Core i5 processor and 4 GB RAM.

A. Parameter Selection

To achieve the best performance, the proper set of parameters needs to be used. Table I shows the algorithm-specific parameters that will be tuned to compare the algorithms. The ranges used for the parameters are the ones specified by the authors [5][6][12][19] in their work. For [27] we introduce the parameter $\Delta\theta$. Experimental results showed that the technique is susceptible to changes in the polar representation of the image; therefore similar values to the ones used for PDP are used. In [12], the authors note that the Chan-Vese algorithm [8] is just a specific formulation (zero-degree polynomial) of the generalized Legendre polynomial framework. Therefore, we rename the algorithm using the zero-degree polynomial ($m = 0$) as the Chan-Vese (CV) algorithm. Higher-degree polynomials are defined as the Legendre level-sets (L2S) algorithm. The algorithms described in [19],[5][6] and [27] are defined as distance-regularized level-set evolution (DRLSE), Poisson inverse gradient (PIG) and globally optimal geodesic active contour (GOGAC), respectively.

The PDP method uses the following parameters: γ, $\Delta\rho$, $\Delta\theta$, α, β, σ. For the images used in this paper (100 × 100 pixels), we use $\gamma = 5°$ and $\Delta\rho = 0.5$ pixels. To further reduce our search space, we let $\beta = 1 - \alpha$. Even though (7) allows the PDP algorithm to handle multiple curvatures (see Section II-D), for simplicity we use only one curvature value for the parameter selection. Note that $\Delta\theta$ the values shown in Table I are multiples of $\Delta\theta_{min}$. For the images used in this paper, $\Delta\theta_{min} = 0.1°$.

Aside from the parameters, the L2S, DRLSE, and CV algorithms require a mask, whereas PDP and GOGAC require an origin, (x_o, y_o). PIG is able to generate the mask in an automated fashion.

B. Sensitivity to Additive Gaussian Noise

To evaluate the performance of PDP under additive Gaussian noise, we add zero-mean Gaussian noise to a simulated crescent-shaped object, shown in Fig. 5(a), using 11 different signal-to-noise ratio (SNR) levels in the 0–10 dB range. The SNR [37] is calculated in terms of the mean intensity values as

$$\mathrm{SNR} = 10\log_{10}\left(\frac{|\mu_{\mathrm{object}} - \mu_{\mathrm{background}}|}{\sigma_{\mathrm{background}}}\right)\mathrm{dB}. \quad (17)$$

To ensure that the results are statistically significant, we used 100 noise realizations at each SNR level. Fig. 5(b) and 5(c) show an instantiation of the 5 dB and the 0 dB cases. To find the parameters for the different algorithms, we randomly selected an image from each of the 11 noise levels and selected the set of parameters (shown in Table I) that gave the best mean Dice metric. Fig. 5(a) shows the initial contour used for L2S, DRLSE, and CV and the origin (shown as a dot) required for PDP and GOGAC. The results (i.e., average Dice and average BPE) for all the algorithms for different SNR levels are shown in Fig. 6. At low SNRs (e.g., 0-1), GOGAC outperforms all techniques, and for higher SNRs CV and L2S outperform all the other techniques. One explanation for the low performance for CV and L2S at low SNR is that they do not smooth the input image as a preprocessing step and thus can be influenced by high noise. Conversely, the absence of smoothing allows these algorithms to generate good results at higher SNR levels. One explanation for the low performance of PDP at low SNR levels is the gradient image. At low SNR levels there are multiple high gradient regions associated with noisy pixels and thus the technique may get latched into edges associated with noisy pixels. Increasing σ will mitigate these problems.

C. Grayscale Images

To evaluate the performance of the algorithms with different grayscale images we used the non-medical and medical images shown in Fig. 7. The first row shows the initialization contour for L2S, DRLSE, CV, as well as the initial origin (dot) used for PDP and GOGAC. The other rows show the closed contours obtained with the PDP, L2S, DRLSE, PIG, GOGAC, and CV algorithms. In Fig. 7(e) and 7(f) the algorithms are outlining the right ventricle and left atrium on cardiac MR images, respectively. The parameters used for the different algorithms are those that gave the best Dice metric. The Dice metric and average BPE for the images are shown in Table II. The proposed algorithm is able to segment high curvature objects with well defined edges similar to L2S or CV (e.g., Fig. 7(a), 7(b), and 7(c)), while also being able to segment lower curvature objects with lower contrast (e.g., Fig. 7(d), 7(e), and 7(f)), similar to the PIG and GOGAC algorithms.

Note that for the PIG algorithm we had to input the number of objects present in the image. In the horse and short-axis right ventricle image, we indicated to look for two objects, whereas for the four-chamber left atrium we indicated to look for four objects. Once the algorithm had generated multiple contours, we kept the contour with the best Dice metric.

The main limitation of the PDP algorithm is shown in Fig. 8, which is a schematic of the tail region of the horse shown in Fig. 7(c). Given a weak sub-contour, shown as a thick black line in Fig. 8(a), the localized region growing (Section II-E) finds a new origin, shown in Fig. 8(b), and generates a new mask. To add the new mask (Ω^n) to the prior mask (Ω), the

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

Author Manuscript

Author Manuscript

Author Manuscript

Author Manuscript

number of low-gradient pixels in the new extracted closed contour must be less than the number of pixels in the weak sub-contour. Since the PDP segmentation (Sections II-A to II-D) finds the optimum path given that the rays emanating from the new origin within the shape intersect the shape boundary in one point, the new mask is not the entire tail, but a subsection of the tail, shown in Fig. 8(c). Since the weak gradient region of the new mask, shown as a thick black line in Fig. 8(c), is bigger than the weak sub-contour, the weak sub-contour is marked as closed and thus the algorithm fails to outline the horse tail. Note that the segmentation generated by GOGAC for Fig. 7(c) is very similar to the one generated by PDP.

The evaluation of the grayscale images using a suboptimal implementation of the PDP in MATLAB took 10-70 seconds; similar MATLAB implementations of L2S, DRLSE, PIG, GOGAC and CV took 1-7 seconds, 20-49 seconds, 0.3-1 seconds, 0.5-7 seconds and 0.5-5 seconds respectively.

D. Parameter Robustness

Aside from the segmentation accuracy, we also evaluated the robustness of the algorithm-specific parameters shown in Table I. We selected the parameters that generated the masks shown in Fig. 7 and changed them by ±1, ±5, ±10 and ±15 percent. An algorithm that is robust to its parameters will yield similar results even when the parameter has been changed by as much as percent. This analysis was performed only on the grayscale images where the technique yielded a Dice metric equal or higher than 0.8, as seen in Table II. For PDP, and GOGAC all 6 images where used; for PIG and DRLSE 5 of 6 images were used; for CV 4 of 6 images were used and for L2S 3 of 6 images were used.

Table III shows the mean difference between the performance metrics given the change in parameters. These changes were done with some constraints (shown in Table III) to generate valid values. For example, in PDP and DRLSE there can be smoothing ($\sigma = 1.5$) or no smoothing ($\sigma = 0$). Non-positive σ values are not valid; therefore, in cases where a non-smoothed gradient ($\sigma = 0$) generated the best result, $\sigma = 0.01, 0.05, 0.1, 0.15$ were used instead.

Positive values in Table III show performance degradation, meaning that the masks generated with the new parameters yielded a smaller Dice metric or higher average BPE than the ones generated with the initial parameter. A negative value (i.e., improvement) means that using the new parameter generated a mask with either a higher Dice metric or a lower average BPE than the initial parameter. There are cases where the Dice metric gives a positive value while the average BPE shows a negative value. This is because we are using the parameters that optimize the Dice metric, not the average BPE.

As a general rule, the value of α should be selected as follows. To segment an object with one mode (e.g., Fig. 7(e)) the value of α should be in the 0.6-0.9 range. When segmenting an object that has more than one mode, but its gradient magnitude is high compared to the background (e.g., Fig. 7(d)), the value of α should be in the 0-0.3 range. For high-curvature objects (e.g., Fig. 7(b)), a $\sigma\theta$ value close to $\sigma\theta_{min}$ should be used. For low-curvature objects, $\sigma\theta$ should be a multiple of $\sigma\theta_{min}$. When the gap between the edges is small (e.g., Fig. 7(c)),

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

no smoothing should be applied (i.e., $\sigma = 0$). When dealing with noisy images, a value of $\sigma > 0$ should be used.

The L2S algorithm was not affected by changes to λ_1 and λ_2; for v the degradation was fairly small. A change of ±15% in v produced a degradation of less 5% than in the mean Dice metric and a bias of less than 2 pixels for average BPE. For L2S the polynomial order (m) turned out to be a sensitive parameter in this set of images. Unlike most of the parameters shown in Table I, m could only have two possible values, 2 or 3. We quantified the effect on the performance by using the incorrect polynomial order (e.g., $m = 2$ instead of $m = 3$ for L2S). Using this approach we observe a degradation of 20% in the mean Dice metric for selecting the wrong polynomial order.

DRLSE was sensitive to variations on all its parameters. PIG was sensitive to variations on all its parameters, except α. The main problem of the algorithm was the inability to generate an initial closed contour that was on the object we wanted to segment. This is one of the main drawbacks of using a purely automated method for creating the mask. GOGAC was robust to most algorithm-specific parameters with the exception of exponential on gradient strength (p). Selecting the incorrect exponential of gradient strength resulted in a degradation of 3% in the mean Dice metric. CV was robust to changes in λ_1, λ_2 and v. A change of ±15% in v resulted in less than a in degradation in the Dice metric and a bias of less than 0.01 pixel.

E. Robustness to Initialization

Aside from the algorithm-specific parameters, we tested the robustness of the PDP to initialization (i.e., origin). This test focused on two images, the hand (Fig. 7(b)) and the horse (Fig. 7(c)) using the parameters that yielded the results shown in Fig. 7. Fig. 9 shows the location of the initialization points as well as the Dice metric, average BPE and convergence time. For the hand image (Fig. 9(a)) 241 initializations were used. From the 241 used initializations, 170 (i.e., green dots) yielded comparable Dice and average BPE metrics reported in Table II and shown in Fig. 7(b). Most of these initializations resided in the central part of the hand. The other 71 initializations (i.e., red dots in Fig. 9(a)) failed to segment the object and thus, have a lower Dice and higher average BPE. This is because when the origin resides on one of the fingers, it finds an initial segmentation (SPDP) that outlines the finger. When GROW tries to grow the low-gradient regions (Section II.E), the total length of the weak sub-contours in the new mask is more than the length of the weak sub-contour, therefore the new mask is rejected and the technique stops growing, thus failing to outline the hand. This explains why there is a drop in the average convergence time.

For the horse image (Fig. 9(b)), 86 different origins were used. Out of the 86 different initializations, the 39 that covered the body of the horse yielded similar results to the ones reported in Table II and shown in Fig. 7(c). The failing cases happened when the initializations fell in the tail, the hind leg, the face or the front legs. Similar to Fig. 9(a) there is a drop in both Dice and average convergence time, and an increase in the average BPE.

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

IV. CONCLUSION

We developed an algorithm that uses polar dynamic programming to outline complex shapes. By introducing the polar variance image, we did not have to constrain the size of the object for correct boundary delineation, something that previous implementations of polar dynamic programming [24]-[26] were not able to accomplish. The proposed algorithm is able to segment high curvature objects, while also segmenting low-gradient objects. We showed that our technique performed favorably when compared to other segmentation algorithms. In future work we will extend the algorithm to segment 3D objects.

Acknowledgments

This work was supported in part by the National Institute of Health under grant T32-HL007955-15.

Biographies

José A. Rosado-Toro received his Ph.D. degree in Electrical and Computer Engineering with a minor in Biomedical Engineering from the University of Arizona in 2016.

He is currently working as a postdoctoral fellow for the Physiological Sciences Department at the University of Arizona. His research interests include image processing, with applications in biomedicine, artificial intelligence and other areas.

María I. Altbach is an Associate Professor in the Department of Medical Imaging, University of Arizona. She obtained her Ph.D. in Chemistry from the Louisiana State University.

Her research interest is in medical imaging, mainly in the field of magnetic resonance imaging.

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

Jeffrey J. Rodríguez (SM'90) is an Associate Professor, Director of Graduate Studies in the Department of Electrical and Computer Engineering and a faculty member of the Biomedical Engineering Interdisciplinary Program at the University of Arizona. He received a Ph.D. degree in electrical engineering from the University of Texas at Austin in 1990.

Current research involves signal/image/video processing and analysis, combined with pattern recognition techniques, to develop novel solutions to interdisciplinary problems in biology, medicine, homeland security, and consumer electronics.

REFERENCES

[1]. Kass M, Witkin A, Terzopoulos D. Snakes: Active contour models. Int. J. Comput. Vis. Jan.1987 1(4):321–331.

[2]. Xu C, Prince JL. Snakes, shapes, and gradient vector flow. IEEE Trans on Image Process. Mar. 1998 7(3):359–369.

[3]. Xu C, Prince JL. Generalized gradient vector flow external force for active contours. Signal Process. Dec.1998 71(2):131–139.

[4]. Ray N, Acton ST. Motion gradient vector flow: An external force for tracking rolling leukocytes with shape and size constrained active contours. IEEE Trans. Med. Imag. Dec.2004 23(12):1466–1478.

[5]. Li B, Acton ST. Active contour external force using vector field convolution for image segmentation. IEEE Trans on Image Process. Aug.2007 16(8):2096–2106.

[6]. Li B, Acton ST. Automatic active model initialization via Poisson inverse gradient. IEEE Trans on Image Process. Aug.2008 17(8):1406–1420.

[7]. Mumford D, Shah J. Optimal approximations by piecewise smooth functions and associated variational problems. Commun. Pure Appl. Math. Oct.1989 42(5):577–685.

[8]. Chan T, Vese L. Active contour without edges. IEEE Trans on Image Process. Feb.2001 10(2): 266–277.

[9]. Li C, Kao C, Gore JC, Ding Z. Minimization of region-scalable fitting energy for image segmentation. IEEE Trans on Image Process. Oct.2008 17(10):1940–1949.

[10]. Lankton S, Tannenbaum A. Localizing region-based active contours. IEEE Trans on Image Process. Nov.2008 17(11):2029–2039.

[11]. Barbosa D, Dietenbeck T, Schaerer J, D'hooge J, Friboulet D, Bernard O. B-Spline Explicit Active Surfaces: An efficient framework for real-time 3D region-based segmentation. IEEE Trans on Image Process. Jan.2012 21:241–251.

[12]. Mukherjee S, Acton ST. Region based segmentation in presence of intensity inhomogeneity using Legendre polynomials. IEEE Signal Process. Lett. Mar.2015 22(3):298–302.

[13]. Osher S, Sethian J. Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations. J. Comput. Phys. Nov.1988 79(1):12–49.

[14]. Caselles V, Catte F, Coll T, Dibos F. A geometric model for active contours in image processing. Numer Math. Dec.1993 66(1):1–31.

[15]. Malladi R, Sethian JA, Venturi BC. Shape modeling with front propagation: A level set approach. IEEE Trans. Pattern Anal. Mach. Intell. Feb.1995 17(2):158–175.

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

[16]. Sethian J. Level Set Methods and Fast Marching Methods. Cambridge Univ. Press; Cambridge, U.K.: 1999.

[17]. Osher S, Fedkiw R. Level Set Methods and Dynamic Implicit Surfaces. Springer-Verlag; New York: 2002.

[18]. Li C, Xu C, Gui C, Fox MD. Level-set evolution without re-initialization: a new variational formula. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR). Jun.2005 1:430–436.

[19]. Li C, Xu C, Gui C, Fox MD. Distance regularized level set evolution and its application to image segmentation. IEEE Trans on Image Process. Dec.2010 19(12):3243–3254.

[20]. Sonka M, Hlavac V, Boyle R. Image Processing, Analysis and Machine Vision. Fourth. Cengage Learning; Stanford CT: 2015. 206–210. Sect. 6.2.5

[21]. Rosado-Toro JA, Barr T, Galons JP, Marron MT, Stopeck A, Thomson C, Thompson P, Carroll D, Wolf E, Altbach MI, Rodriguez JJ. Automated breast segmentation of fat and water MR images using dynamic programming. Academic Radiology. Feb.2015 22(2):139–148. [PubMed: 25572926]

[22]. Avidan S, Shamir A. Seam carving for content-aware image resizing. ACM Trans. Graph. Jul. 2007 26(3) article 10.

[23]. Moore AP, Prince SJD, Warrell J, Mohammed U, Jones G. Superpixel lattices. Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Jun.2008 1:1–8.

[24]. Sun C, Pallotino S. Circular shortest path in images. Pattern Recognit. Mar.2003 36(3):709–719.

[25]. Timp S, Karssemeijer N. A new 2D segmentation method based on dynamic programming applied to computer aided detection in mammography. Med. Phys. May; 2004 31(5):958–971. [PubMed: 15191279]

[26]. Zhang Y, Matuszewski BJ, Shark LK, Moore CJ. A novel medical image segmentation method using dynamic programming. IEEE Int. Conf. Med. Inform. Visualization – BioMed. Visualization (MediVis ’07). Jul.2007 :69–74.

[27]. Appleton B, Talbot H. Globally optimal geodesic active contours. J. Math. Imaging Vision. Jul. 2005 23(1):67–86.

[28]. Peyre G. The numerical tours of signal processing – advanced computational signal and image processing. IEEE Computing Sci. Eng. Jun.2011 13(4):94–97.

[29]. Appleton B, Talbot H. Globally minimal surfaces by continuous maximal flows. IEEE Trans. Pattern Anal. Mach. Intell. Jan.2006 28(1):106–118. [PubMed: 16402623]

[30]. Collewet C. Polar snakes: a fast and robust parametric active contour model. IEEE Int. Conf. on Image Process. Nov.2009 16:3013–3016.

[31]. Chang DC, Wu WR. Image contrast enhancement based on a histogram transformation of local standard deviation. IEEE Trans. Med. Imag. Aug.1998 17(4):518–531.

[32]. Bocher PK, McCloy KR. The fundamentals of average local variance – part I: detecting regular patterns. IEEE Trans on Image Process. Feb.2006 15(2):300–310.

[33]. Handbook of Computer Vision Algorithms in Image Algebra. 2nd. CRC Press Inc.; Boca Raton, FL: 2000. 313

[34]. Bresenham JE. Algorithm for computer control of a digital plotter. IBM Syst J. 4(1):25–30.

[35]. Dice LR. Measures of the amount of ecologic association between species. Ecology. Jul.1945 26(3):297–302.

[36]. Sonka M, Hlavac V, Boyle R. Image Processing, Analysis and Machine Vision. Fourth. Cengage Learning; Stanford CT: 2015. 238–239. Sect. 6.5.1.2

[37]. Fiete RD, Tantalo T. Comparison of snr image quality metrics for remote sensing systems. Opt. Eng. Apr.2001 40(4):574–585.

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

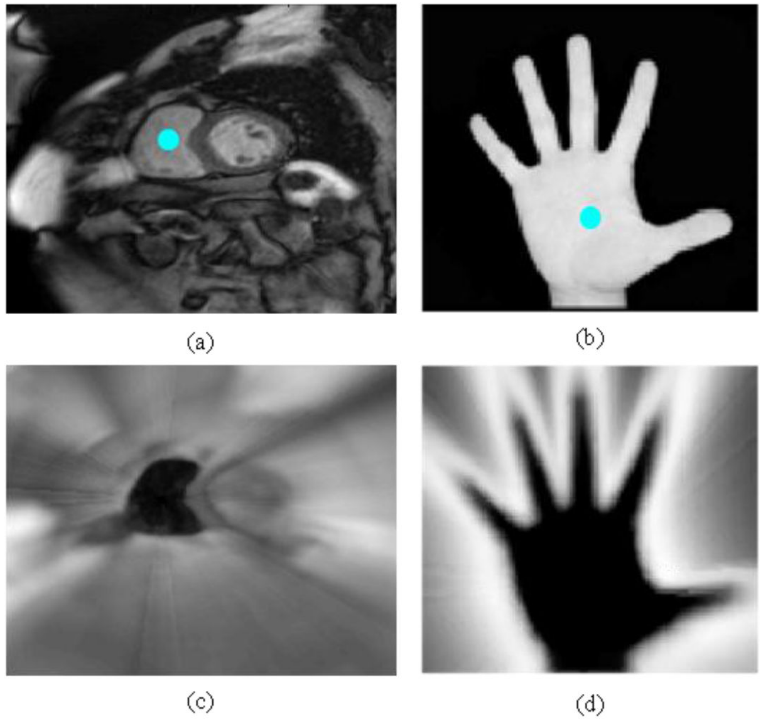


Fig. 1.
Grayscale and polar variance images. (a) Short-axis cardiac MR image and (b) hand image. (c-d) Polar variance images of (a) and (b) respectively. The dot denotes the object we wish to outline (right ventricle in (a), the entire hand in (b)) and is used to calculate the polar variance image. Black in the polar variance images represents low variance.

Author Manuscript

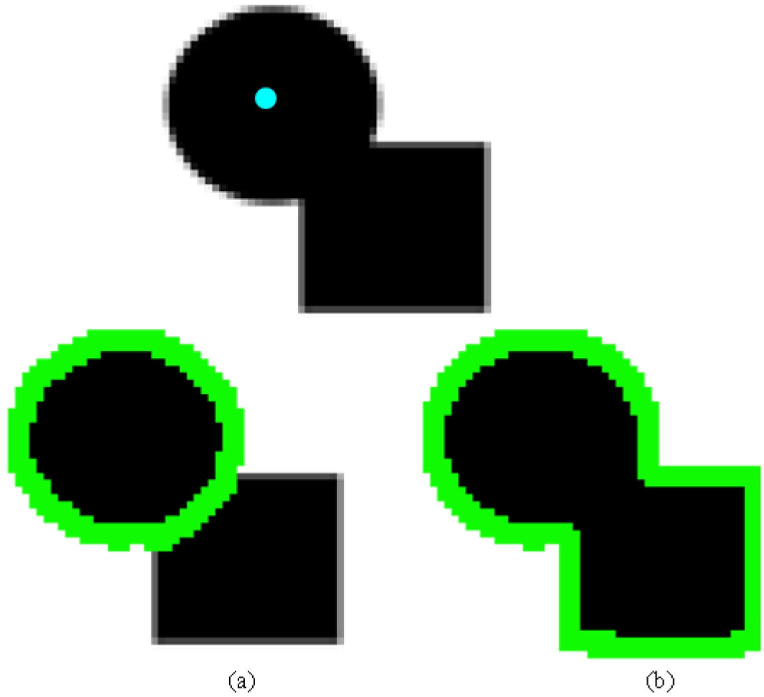


Fig. 2.

Assessing how $\Delta\theta$ affects closed contour. For this example, α and β equal 0.5. The dot is the origin used for PDP. Results using a $\Delta\theta$ of (a) 5° and (b) 0.1°. Closed contours have been dilated for viewing purposes.

procedure $GROW(\Omega, I, G, \gamma, \alpha, \beta, \Delta\rho, \Delta\theta)$

Input Ω is the initial segmentation calculated using the user selected origin x_o, y_o. Ω is the union of Ω_1 and Ω_2, where Ω_1 is the region inside the closed contour and Ω_2 is the extracted contour. I is the intensity image and G is the gradient image. γ is the smoothing parameter for the polar variance image (default value 5°). α and β are the weight values for the cost function (default values 0.5). $\Delta\rho$ is the radial sampling interval (default value 0.5) and $\Delta\theta$ is the angular sampling interval (default value is a multiple of $\Delta\theta_{min}$).

Output Ω is the grown region.

1. threshold = mean of G in Ω_1
2. weakEdges = $(G <= threshold) \cap \Omega_2$
3. Initialize closedEdges as a zero matrix the size of G
4. **while** sum(weakEdges(:)) > 0 **do**
5. $[x_n, y_n]$ = Point of the largest connected component of
6. weakEdges that is the farthest away from a strong edge.
7. $\Omega_{temp} = SPDP(I, G, x_n, y_n, \gamma, \alpha, \beta, \Delta\rho, \Delta\theta)$ \\ Sections II:A-D
8. $\Omega_1^{new} = \Omega_1^{temp} \cap \Omega^C$ \\ Ω^C is complement of Ω
9. $\Omega_2^{new} = \Omega_2^{temp} \cap \Omega^C$ \\ Ω^C is complement of Ω
10. $\Omega^{new} = \Omega_1^{new} \cup \Omega_2^{new}$
11. Ω^{new} is the connected component from Ω^{new} that is closest to
12. $[x_n, y_n]$.
13. $\Omega_2^{new} = \Omega^{new} \cap \Omega_2^{new}$ \\Redefinition based on Ω^{new} from 11-12.
14. weakEdgesNew = $(G <= threshold) \cap \Omega_2^{new}$
15. **if** the number of nonzero pixels in weakEdgesNew is smaller than the
16. largest connected component of weakEdges **then**
17. $\Omega = \Omega \cup \Omega^{new}$
18. **end if**
19. Take largest connected component of weakEdges that overlaps with
20. Ω^{new} and add it to the closedEdges
21. threshold = mean of G inside the closed contour Ω_1
22. weakEdgesTemp = $(G <= threshold) \cap \Omega_2$
23. weakEdges = (weakEdgesTemp $\cup$ weakEdges) $\cap$ closedEdges^C
24. **end while**
25. **return** Ω

end procedure

Fig. 3.

Pseudo code of localized region-growing (Section II-E).

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

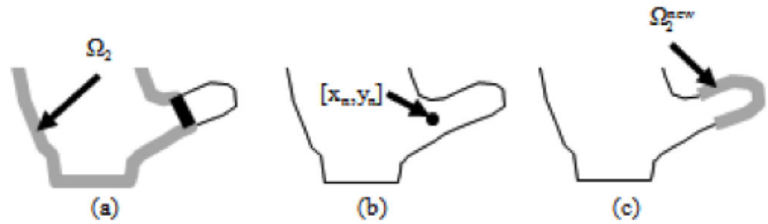


Fig. 4.

Schematic of the thumb region of the hand shown in Fig. 1(b). The thin black line represents the high gradient region and the white background is the low gradient region. (a) The gray thick line shows part of the high gradient extracted contour of the original mask. The black thick line is the weak sub-contour. (b) The dot is the new origin positioned in the middle of the weak sub-contour shown in (a). (c) The gray thick line is the high gradient extracted contour that does not intersect with the original mask, (a).

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

Author Manuscript
Author Manuscript
Author Manuscript
Author Manuscript

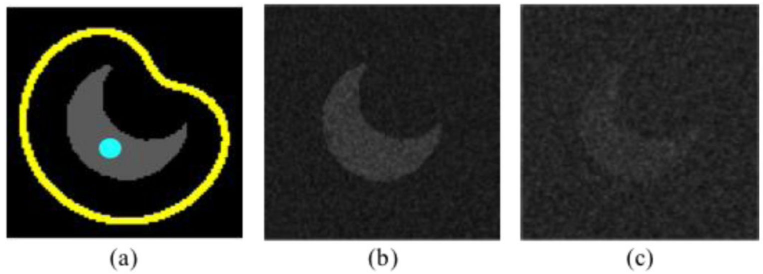


Fig. 5.

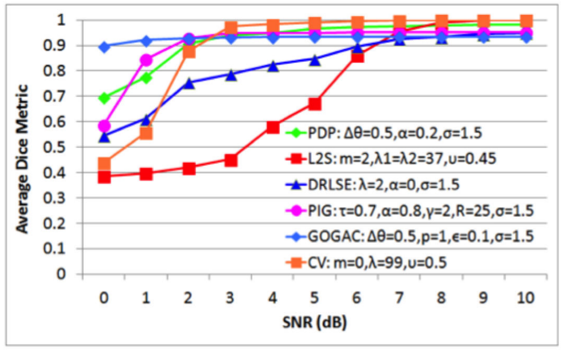
Simulated images with SNR = (a) ∞ , (b) 5 and (c) 0 dB. Outline in (a) is the initial closed contour used for L2S, DRLSE and CV; dot is the origin location used for PDP and GOGAC. Closed contour has been dilated for viewing purposes.

Author Manuscript

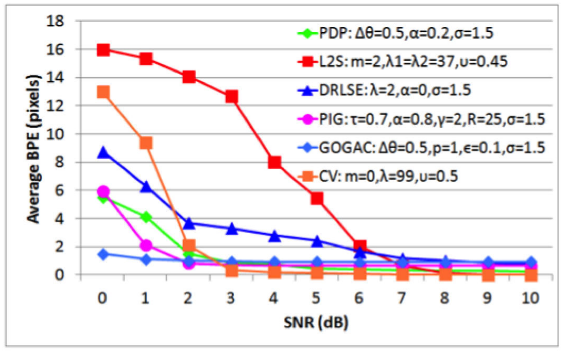
Author Manuscript

Author Manuscript

Author Manuscript



(a)



(b)

Fig. 6.

Noise sensitivity performance analysis for (a) average Dice performance and (b) average border positioning error (BPE) in pixels. One hundred noisy images were created for each SNR level by adding Gaussian noise to the image shown in Fig. 5(a).

Author Manuscript

Author Manuscript

Author Manuscript

Author Manuscript

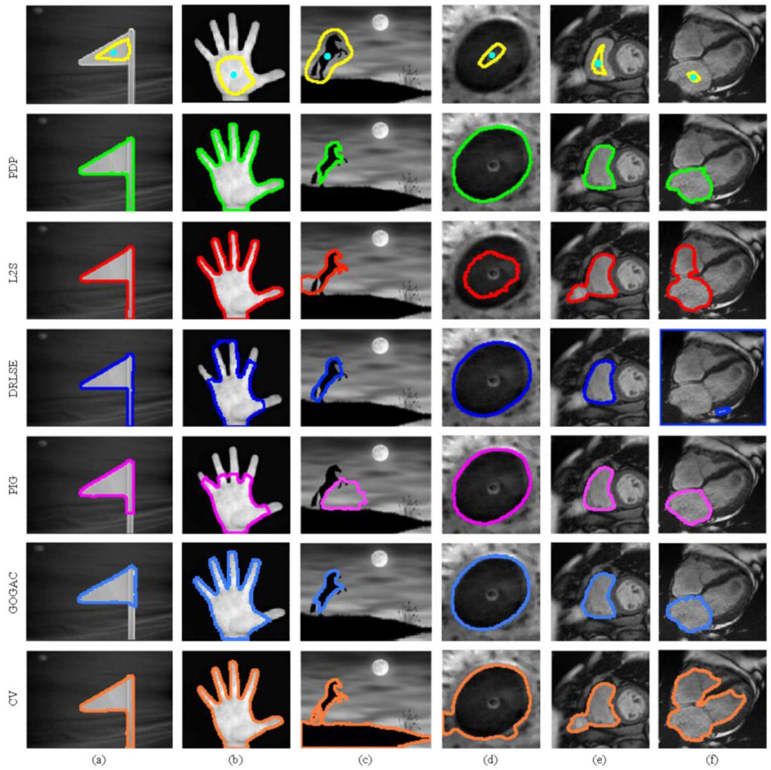


Fig. 7.
Non-medical and medical images. From left to right (a) flag, (b) hand, (c) horse, (d) retina, (e) short-axis and (f) four chamber cardiac image. Upper row shows the initialization for the different algorithms. Short-axis cardiac images (e), have been cropped for display purposes. The image used for analysis is shown in Fig. 1(a). Closed contours have been dilated for viewing purposes.

Author Manuscript

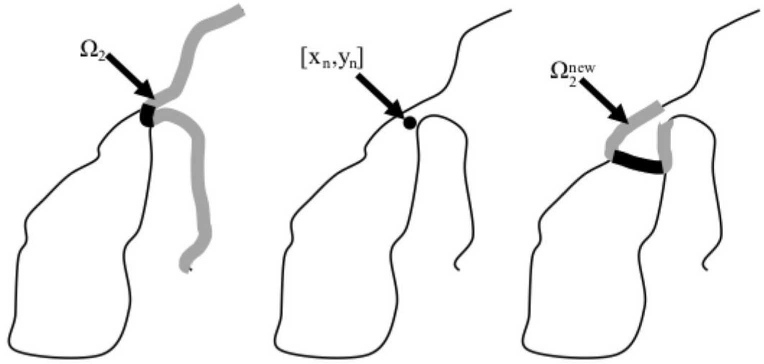


Fig. 8.

Schematic of the tail region of the horse shown in Fig. 7(c). The thin black line represents the high gradient region and the white background is the low gradient region. (a) The gray thick line shows part of the high gradient extracted contour of the original mask. The black thick line is the weak sub-contour. (b) The dot is the new origin positioned in the middle of the weak sub-contour shown in (a). (c) The gray thick line is the high gradient extracted contour that does not intersect with the original mask, (a).

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

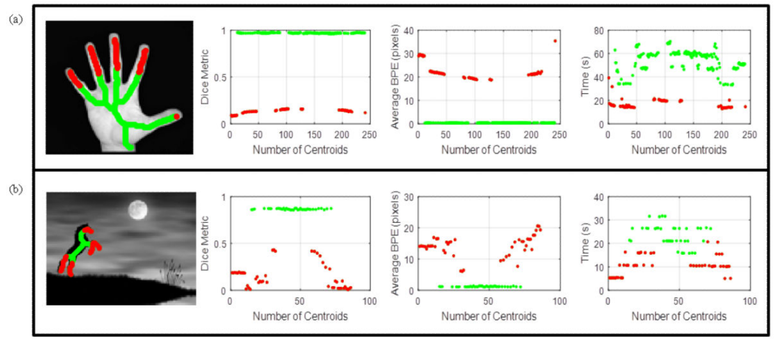


Fig. 9.

Robustness of PDP to different origin locations in (a) hand and (b) horse. Green dots are cases where the performance is comparable to the closed contours shown in Fig. 7. Red dots are cases where the performance is considerably lower than the closed contours shown in Fig. 7.

Author Manuscript
Author Manuscript
Author Manuscript
Author Manuscript

TABLE I

Algorithm-Specific Parameters

Technique	Parameters	Description	Values
PDP	$\Delta\ \theta$	allowed curvature	0.1,0.5,1,2,3, 4,5
	α	variance image weight	From 0 to 0.9, $\Delta = 0.1$
	σ	Gaussian kernel smoothing	0,1.5
L2S	m	Legendre polynomial order	2,3
	λ_1,λ_2	regularization constraints (λ_1=λ_2)	From 1 to 100, $\Delta = 2$
	v	contour smoothness constraint	From 0.05 to 0.6, $\Delta = 0.05$
DRLSE	λ	line integral coefficient	From 0 to 10, $\Delta = 1$
	α	area coefficient shrinks)	From −10 to
	σ	Gaussian kernel smoothing	0,1.5
PIG	τ	gradient edge threshold	From 0.1 to 0.8, $\Delta = 0.1$
	α	curvature weight	From 0 to 1, $\Delta = 0.1$
	γ	VFK decay coefficient	From 1.5 to 3, $\Delta = 0.5$
	R	VFK radii, based on the size of the image (N is the size of the image)	N/8,N/7,N/6, N/5,N/4,N/3, N/2
	σ	Gaussian kernel smoothing	1.5
GOGAC	$\Delta\ \theta$	Allowed curvature	0.1,0.5,1
	p	exponential on gradient strength	1,2
	ϵ	arc-length penalty term	0.001, 0.01, 0.1, 1, 10
	σ	Gaussian kernel smoothing	0,1.5
CV	λ_1,λ_2	regularization constraints (λ_1=λ_2)	From 1 to 100, $\Delta = 2$
	v	contour smoothness constraint	From 0.05 to 0.6, $\Delta = 0.05$

Notation from original papers, and Δ is the uniform increment.

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

TABLE II

Performance of Segmentation Algorithms

Technique	Flag		Hand		Horse		Retina		Short-axis right ventricle		Four-chamber left atrium	
	Dice	**BPE**	**Dice**	**BPE**	**Dice**	**BPE**	**Dice**	**BPE**	**Dice**	**BPE**	**Dice**	**BPE**
PDP	0.9631	0.5213	0.9491	0.7190	0.8733	1.1024	0.9947	0.2172	0.9543	0.7172	0.9549	0.8186
L2S	0.9399	0.8163	0.9762	0.3451	0.7761	1.7970	0.5694	13.3977	0.8701	2.4988	0.7351	5.6670
DRLSE	0.9159	1.1163	0.8662	2.4240	0.8587	1.1323	0.9895	0.4297	0.9371	0.9655	0.1964	37.1450
PIG	0.9018	2.0213	0.8326	4.2837	0.1119	8.3903	0.9983	0.0685	0.9394	0.9195	0.9335	1.2021
GOGAC	0.8825	3.9817	0.9195	1.2235	0.8458	1.1863	0.9931	0.2756	0.9314	1.0767	0.9518	0.8833
CV	0.9471	0.7340	0.9741	0.3798	0.2618	21.4422	0.9324	2.4891	0.8484	2.7902	0.5187	12.0072

Quantitative performance for images shown in Fig. 7.

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

TABLE III

Robustness to Changes in Algorithm-Specific Parameters

Technique	Parameters	Constraints	±1%		±5%		±10%		±15%	
			ΔDice	ΔBPE	ΔDice	ΔBPE	ΔDice	ΔBPE	ΔDice	ΔBPE
PDP	$\Delta \theta$	$\Delta\theta>0$	0.0059	0.1048	0.0076	0.1278	0.0076	0.1261	0.0084	0.1438
	α	$0 \leq \alpha < 1$	0.0015	0.0375	0.0032	0.0620	0.0149	0.4247	0.0157	0.4227
	σ	$\sigma>0$	0	−0.0004	0.0045	0.0595	0.0045	0.0607	0.0113	0.1567
L2S	m	if $m_{old}=2$ $m_{new}=3$	0.1810	3.1189	0.1810	3.1189	0.1810	3.1189	0.1810	3.1189
		if $m_{old}=3$ $m_{new}=2$								
	λ_1,λ_2	$\lambda_1,\lambda_2>0$	0	0	0	0	0	0	0	0
	v	$0 \leq v < 1$	0.0003	0.0031	0.0001	0.0015	0.0003	0.0035	0.0004	0.0048
DRLSE	λ	$\lambda>0$	0.0359	1.7321	0.0453	2.3018	0.0999	3.997	0.2402	8.0538
	α	if $\alpha_{old}>0$, $\alpha_{new}>0$	0.0379	2.0672	0.0598	2.4183	0.1819	4.3407	0.2158	5.1481
		if $\alpha_{old}<0$, $\alpha_{new}<0$								
	σ	$\sigma>0$	0.1118	4.6195	0.1585	7.385	0.1136	5.2672	0.1283	5.8768
PIG	T	$0 \leq \tau < 0.8$	0.0793	4.3448	0.1025	9.1074	0.1691	8.7681	0.2097	15.0553
	α	$0 \leq \alpha \alpha 1$	0.0018	0.0691	0.0040	0.1397	0.0026	0.1051	0.0026	0.0941
	γ	$\gamma>0$	0.0068	0.1474	0.0882	4.5255	0.0931	4.6330	0.0970	4.6336
	R	$R>0$	0.0814	4.4366	0.0824	4.3897	0.0859	4.4776	0.0923	4.5874
	σ	$\sigma>0$	0.1681	8.6817	0.1997	9.2172	0.1799	8.3331	0.1667	8.7000
GOGAC	$\Delta \theta$	$\Delta\theta>0$	0.0013	0.0231	0.0022	0.0403	0.0013	0.0263	−0.0002	−0.0010
	p	if $p_{old}=1$ $p_{new}=2$	0.0268	0.7652	0.0268	0.7652	0.0268	0.7652	0.0268	0.7652
		if $p_{old}=2$ $p_{new}=1$								
	ϵ	$\epsilon>0$	0	0	0	0	0	0	0	0
	σ	$\sigma>0$	0	0.0001	0	0.0001	0	0.0022	0	0.0022
CV	λ_1,λ_2	$\lambda_1,\lambda_2>0$	0	0	0	0	0	0	0	0
	v	$0 \leq v \leq 1$	0.0014	0.0226	0.0011	−0.0149	−0.0016	−0.0047	0.0013	0.0078

Author Manuscript

Change in Dice metric and when the parameters used in Fig. 7 are altered by *±1,5,10,15%.*

Author Manuscript
Author Manuscript
Author Manuscript