Patch-based potentials for interactive contour extraction

Thoraya Ben Chattah, Sébastien Bougleux, Olivier Lézoray, Atef Hamouda

Patch-based potentials for interactive contour extraction

Thoraya Ben Chattah[1,2], Sébastien Bougleux[1], Olivier Lézoray[1], and Atef Hamouda[2]

[1] Normandie Univ, UNICAEN, ENSICAEN, CNRS, GREYC, 14000 Caen, France
{bougleux,olivier.lezoray}@unicaen.fr
[2] University of Tunis El Manar, Faculty of Siences of Tunis
LIPAH-LR11ES14, 2092 Tunis, Tunisia

Presented at the 13th International Symposium on Visual Computing (ISVC)
Las Vegas, November 19-21, 2018 (https://www.isvc.net/)

Abstract. The problem of interactive contour extraction of targeted objects of interest in images is challenging and finds many applications in image editing tasks. Several methods have been proposed to address this problem with a common objective: performing an accurate contour extraction with minimum user effort. For minimal paths techniques, achieving this goal depends critically on the ability of the so-called potential map to capture edges. In this context we propose new patch-based potentials designed to have small values at the boundary of the targeted object. To evaluate these potentials, we consider the livewire framework and quantify their abilities in terms of number of needed seed points. Both visual and quantitative results demonstrated the strong capability of our proposed potentials in reducing the user's interaction while preserving a good accuracy of extraction.

Keywords: Contour extraction · Patch · Minimal paths.

1 Introduction

Despite the high number of research works in image segmentation, interactive contour extraction is still a very challenging image processing problem. In contrast to the traditional image segmentation problem, many real-world applications focus on identifying the pixels belonging to a specific object. Their aim is to precisely delineate the contour of a targeted object. This is at the core of many image editing tasks in photography or medical image analysis: using a selection tool, an object is extracted and the background is removed. This process can be very difficult to do cleanly manually because of the presence of structures with ill-defined borders, and assistive tools can be very beneficial for end-users. For that, it is necessary for the user to provide inputs to ease the object extraction. These inputs can take different forms that are then incorporated into the segmentation process as hard or soft constraints. Since this requires an

interaction with the user, all methods try to obtain accurate segmentation results while reducing the user's effort. There are essentially three methods in the literature to allow the user to specify the targeted object. The first is to label some pixels inside/outside the object [10] and to use a labeling process such as graph-cut based-methods [8]. The second method is to provide a sub-region within the image that contains the object. Bounding boxes have often been considered for that, see [1], and have been popularized by the GrabCut method [15]. The third method is to provide a curve close to the boundary or seed points on the boundary. Providing curves has been extensively employed in level sets and snakes methods [4]. Providing points has been mainly employed with minimal path approaches [5]. In the sequel we will focus on that last type of interaction that we call seed-based interactive contour extraction methods.

The paper is organized as follows. Section 2 provides a review of the minimal paths approach for interactive contour extraction and describes the livewire framework. Section 3 describes our patch-based potentials. They are compared in Section 4 through experiments on ISEG dataset.

2 Seed-based Interactive contour extraction methods

2.1 Minimal paths

Most seed-based interactive contour extraction methods rely on minimal paths. Given an image $f : \Omega \subset \mathbb{R}^2 \to \mathbb{R}_+^n$ and two seeds $p_s, p_e \in \Omega$ located on the contour of an object, the objective is to find a path from p_s to p_e that represents a piece of contour. In this paper, we consider the classical model presented in [5]. A path from p_s to p_e is defined as a parametric curve $\gamma : [0,1] \to \Omega$ so that $\gamma(0) = p_s$ and $\gamma(1) = p_e$, and its length as the cost:

$$L(\gamma) = \int_0^1 P(\gamma(s))\|\gamma'(s)\|ds. \tag{1}$$

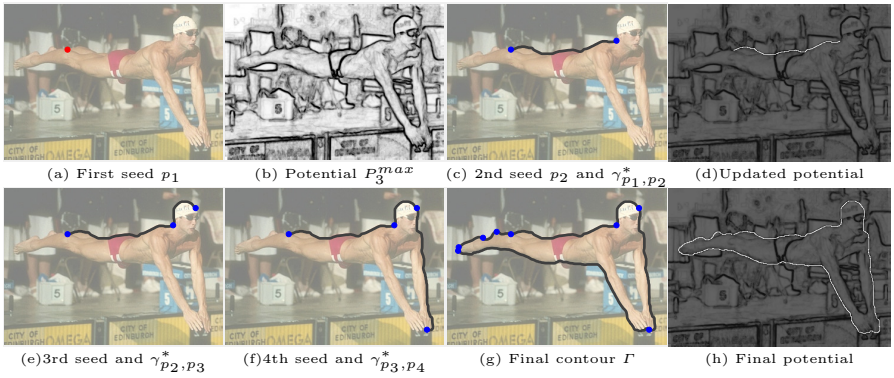
γ' is the derivative of γ. $P : \Omega \to R_+^*$ is a potential function derived from image f that has low values for edges. The length $L(\gamma)$ is smaller when it goes through low values of P and one can search for curves that minimize $L(\gamma)$ given a pair of starting and ending points. The goal is therefore to capture the minimal path γ^*, also called the geodesic path, that globally minimizes the cost function defined in (1):

$$L(\gamma^*) = \min_{\gamma \in \mathcal{A}(p_s,p_e)} L(\gamma) \tag{2}$$

with $\mathcal{A}(p_s, p_e)$ the set of all possible paths joining p_s to p_e. The optimal cost defines the geodesic distance $d(p_s, p_e) = L(\gamma^*)$.

To tackle the minimization problem (2), the geodesic distance map $\mathcal{U}_{\{p_s\}} : \Omega \to \mathbb{R}_+$ from a point p_s to any other point $p \in \Omega$ is considered: $\mathcal{U}_{\{p_s\}}(p) = d(p_s, p) = \min_{\gamma \in \mathcal{A}(p_s,p)} L(\gamma)$. The distance map is solution of the Eikonal equation:

$$\begin{cases} \|\nabla \mathcal{U}_{\{p_s\}}(p)\| = P(p), & \forall p \in \Omega \\ \mathcal{U}_{\{p_s\}}(p_s) = 0 \end{cases} \tag{3}$$



(a) First seed p_1 (b) Potential P_3^{max} (c) 2nd seed p_2 and $\gamma^*_{p_1,p_2}$ (d)Updated potential

(e)3rd seed and $\gamma^*_{p_2,p_3}$ (f)4th seed and $\gamma^*_{p_3,p_4}$ (g) Final contour Γ (h) Final potential

Fig. 1: A livewire interactive contour extraction.

The value $\mathcal{U}_{\{p_s\}}(p)$ can be interpreted as the arrival time at p of a front propagating from a point p_s with speed $1/P$. Fast marching method [16,5] and Dijkstra's algorithm [6] are often used to compute the values of $U_{\{p_s\}}$ in increasing order, so that the process can be stopped when the end point p_e is reached. Then the minimal path between p_s and p_e can be constructed by applying a back-propagation procedure (gradient descent) starting from p_e along the gradient of the map $\mathcal{U}_{p_s}$ until arriving at p_s. See [14] for more details.

An interesting thing to note is that the result curve γ^* depends strongly on the potential functional P which is usually built such as it takes lower values at the desired structure of interest and higher values elsewhere. Some works have suggested the use of new potential functionals or path cost functions in order to better discriminate the discontinuities between objects and background. To this end, and to go beyond classical edge-based energies, some authors have considered more evolved information such as texture, curvature [3], or orientation [2]. We focus on this track of research in this paper.

2.2 Livewire interactive contour extraction framework

Before entering into the details of our proposed potentials, we focus on how interaction can be considered in seed-based minimal paths interactive contour extraction. When points are provided as seeds in minimal path approaches for contour extraction, the employed methods all adopt the same strategy. The user places *seed points* that provides an initial labelling for some pixels of the image. Then, an algorithm is performed to propagate the labels of seeds to unlabeled regions until an optimum criterion is reached. If this strategy is similar from one method to another, the interaction with the user for the providing of seeds differs. A first way to proceed consists in iteratively minimizing an energy functional for a given set of seed points on a contour [5,11]. The main drawback of this type of user interaction is that the user cannot easily add new boundary points on the contour segments that were poorly extracted: this requires a complete

Algorithm 1 Livewire contour extraction framework

Potential function P, Seed set $\mathcal{S} = \emptyset$, extracted contour Γ
Select seed point p_1
Compute geodesic distance map $\mathcal{U}_{\{p_1\}}$ with P
$p_s \leftarrow p_1$, $\mathcal{S} \leftarrow \mathcal{S} \cup \{p_s\}$
repeat
  Select p_e
  Backtrack to determine the shortest curve $\gamma^*_{p_s,p_e}$ from $\mathcal{U}_{\{p_s\}}$
  $S \leftarrow S \cup \{p_e\}$, $\Gamma \leftarrow \Gamma \cup \gamma^*_{p_s,p_e}$
  Update Potential $P(q) \leftarrow +\infty$, $\forall\ q \in \gamma^*_{p_s,p_e} \setminus \{p_s, p_e\}$
  Compute geodesic distance map $\mathcal{U}_{\{p_e\}}$ with P
  Backtrack to determine the shortest curve $\gamma^*_{p_e,p_1}$ from $\mathcal{U}_{\{p_e\}}$
  $p_s \leftarrow p_e$
until The user is satisfied with the extracted contour
$\Gamma = \Gamma \cup \gamma^*_{p_{i+1},p_1}$

re-computation of the solution. A second more popular way to proceed is to let the user place new seed points after each label propagation step.

The user provides starting seed points and the optimal boundary is extracted. Then, the user can add another seed point if he is not satisfied. In that case, the previously extracted boundary is frozen and a new optimal path is computed to the new seed point. This process is repeated until the user decides to close the contour. These methods do belong to the livewire framework. They require an ordered sequence of seed points (provided one after the other) to extract the object's contour. The advantage of this type of interaction is that the user can control and predict the final result. The interaction should be minimal in the sense that few seeds should be used and the largest optimum path between them should be obtained. Algorithm 1 resumes the livewire interactive contour extraction framework and Figure 1 illustrates this interactive process to extract one contour with one of the potentials (P_3^{max}) we propose thereafter. Significant examples of the livewire framework include intelligent scissors methods [13], their variations with the on-the-fly extension [7], the G-wire extension [9] and the riverbed algorithm [12].

3 Patch-based Potentials

In this section, we introduce new patch-based potentials for livewire contour extraction. Generally speaking, a contour is a curve Γ that separates optimally two regions $\mathcal{R}^+$ and $\mathcal{R}^-$ that have different features (e.g., color, texture). Our proposal consists in using patches as texture descriptors to build a potential map that has small values especially for edges pixels.

3.1 Notations

For a color image f, the color at a pixel $p_i = (x_i, y_i)^T$ is given by $f(p_i) \in \mathbb{R}^3$. A patch is a rectangular region of size $(2w+1)^2$ around a pixel and is defined by the vector $\mathcal{P}(p_i) = (f(p_i + t), \forall t \in [-w, w]^2)^T \in \mathbb{R}^{3(2w+1)^2}$. The difference between two patches is provided by the L_2-norm $d(\mathcal{P}(p_i), \mathcal{P}(p_j)) = \|\mathcal{P}(p_i) - \mathcal{P}(p_j)\|_2$.

Given $t_{\theta,\delta}$ a vector of translation δ according to the angle θ with the horizontal line, we can define the left and right pixels at a distance δ of a pixel p_i on a line of angle θ with the horizon as: $p_i^{l,\theta,\delta} = p_i + t_{\theta,\delta}$ and $p_i^{r,\theta,\delta} = p_i - t_{\theta,\delta}$. In the sequel, we will consider lines of angle $\theta \in \{0°, 45°, 90°, 135°\}$.

3.2 Pixel-surround potentials

We propose two first potentials that consider only the pixels surrounding a current pixel p_i in a 8-neighborhood. A pixel can be considered as lying on an edge if the patch at its left is different from the patch at its right. If we compute distances between these left and right patches, a high value means a high probability of contour. We therefore define a patch difference centered on p_i as: $d_c^{\theta,\delta}(p_i) = d\left(\mathcal{P}(p_i^{l,\theta,\delta}), \mathcal{P}(p_i^{r,\theta,\delta})\right)$. However, the edge is not necessarily oriented along the y axis and we have to test all the possible orientations in a 8-neighborhood to define the potential: $P_1(p_i) = \max_\theta d_c^{\theta,1}(p_i)$ with $\delta = 1$ since we consider only the surrounding pixels. This first potential is illustrated in Figure 2 (first row, first column). Since a pixel on an edge is also not supposed to be similar to its surrounding pixels, we can enhance this potential by computing upwind left or right differences between the pixel and its neighbors. We define $d_l^{\theta,\delta}(p_i) = d\left(\mathcal{P}(p_i), \mathcal{P}(p_i^{l,\theta,\delta})\right)$ and $d_r^{\theta,\delta}(p_i) = d\left(\mathcal{P}(p_i), \mathcal{P}(p_i^{r,\theta,\delta})\right)$. Then, we can aggregate all these distances along a direction and take the maximum on all directions:

$$P_2^{AF}(p_i) = \max_\theta AF(d_s^{\theta,1}(p_i)) \tag{4}$$

with $d_s^{\theta,1}(p_i) = \{d_l^{\theta,1}(p_i), d_c^{\theta,1}(p_i), d_r^{\theta,1}(p_i)\}$ and AF an aggregation function among the min, the max and the mean. This second potential is illustrated in Fig. 2 (first row, second column) and Fig. 3.

3.3 Band-based potentials

The drawback of the previous potentials is that they are operating on a small neighborhood to discriminate edge from non-edge pixels. It might be much more beneficial to study a larger neighborhood. Indeed if there is a texture edge at pixel p_i it is natural to expect that shifted patches to the left are different from shifted patches to the right. To take this into account we generalize the previous potential P_2 to operate on a longer line and to consider pixels that are much farther than a distance $\delta = 1$. To that aim, we consider a band of ns patches on a side of the line and to avoid an overlap between the patch at p_i and its left and right patches, we consider pixels at a distance $\delta \in [\epsilon, \epsilon + ns]$. For each possible

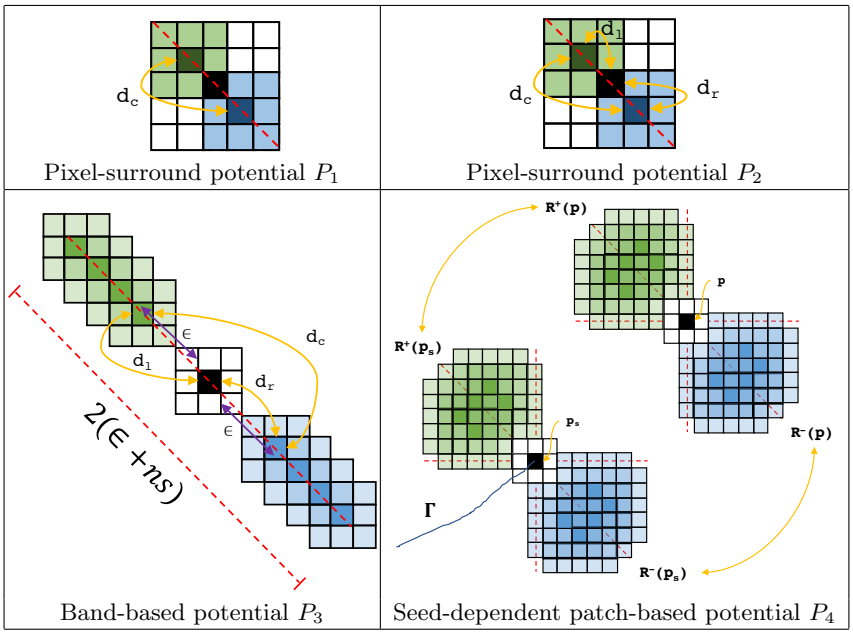


Fig. 2: Computation principles of the proposed potentials.

set of distances (left d_l, right d_r or center d_c), we keep only the minimum on a side. Then we can aggregate all these distances between the band of patches along a direction and take the maximum on all directions:

$$P_3^{AF}(p_i) = \max_{\theta} AF\left(\left\{\min_{\delta\in[\epsilon,\epsilon+ns]}, d_s^{\theta,\delta}(p_i), \forall s \in \{l,c,r\}\right\}\right) \tag{5}$$

This third potential is illustrated in Fig. 2 (second row, first column) and Fig. 3.

3.4 Seed-dependent patch-based potential

All the previous potentials we proposed can be considered as blind as they do not take into account the previously positioned seeds to compute the potential. To overcome this, we design a seed-dependent patch-based potential. Given a seed p_s, it is expected that the regions $\mathcal{R}^+(p_s)$ and $\mathcal{R}^-(p_s)$ on its both sides will be similar to the other regions for the next points of the contour. However, for the first seed $p_s = p_1$ (Algorithm 1), we have no exact idea of the orientation to define the regions it separates. To cope with this, we search for the angle θ_s that maximizes the differences between the set of patches of the two regions. These set of patches are contained within 45° to the band defined by the angle θ to the horizon. We consider only 5 patches of size 5×5 with minimum overlap for

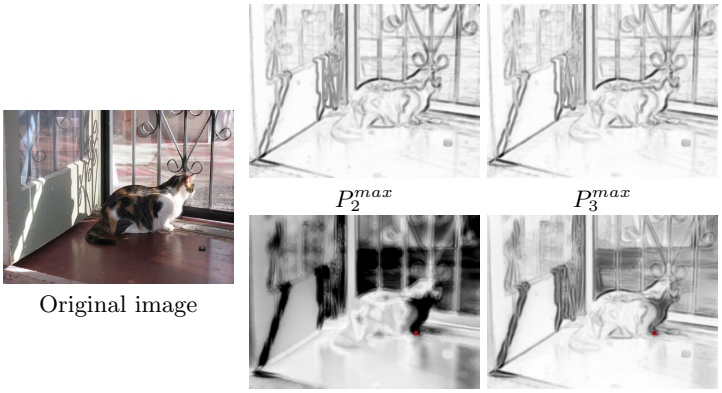


Fig. 3: Potentials maps with 5×5 patches, $AF = \max$ for P_2 and P_3, $\epsilon = 2$ and $ns = 5$ for P_3, and $\alpha = 0.3$ for P_5.

each region. Finally for p_1,

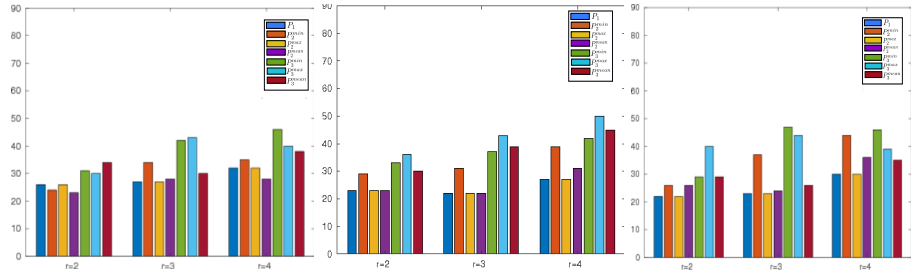
$$\theta_1 = \arg\max_{\theta} \max_{\substack{q_1 \times q_2 \\ q_1 \in \mathcal{R}_\theta^+(p_1) \\ q_2 \in \mathcal{R}_\theta^-(p_1)}} d(\mathcal{P}(q_1), \mathcal{P}(q_2)) \tag{6}$$

This provides two regions $\mathcal{R}_{\theta_1}^+(p_1)$ and $\mathcal{R}_{\theta_1}^-(p_1)$ according to the retained orientation θ_1. For any other seed $p_s \neq p_1$, the regions $\mathcal{R}_{\theta_s}^+(p_s)$ and $\mathcal{R}_{\theta_s}^-(p_s)$ are defined from the normal $N_s = (\gamma_{s-1,s}^{*\prime}(p_s))^\perp / \|\gamma_{s-1,s}^{*\prime}(p_s)\|$ to the shortest path $\gamma_{s-1,s}^*$ at p_s by the angle $\theta_s = \arg\min_\theta \angle(N_s, p_s p_s^{l,\theta,1})$.

Then, for each point p_i of the image, we compute all the pairwise distances between the regions $\mathcal{R}_{\theta_s}^+(p_s)$ and $\mathcal{R}_\theta^+(p_i)$ (and similarly for $\mathcal{R}^-$) for different orientations θ around p_i. We retain only the minimum value that accounts for a very similar configuration between p_s and p_i. This can be formulated by:

$$P_4(p_i) = \min_\theta \left(\min_{\substack{q_1 \times q_2 \\ q_1 \in \mathcal{R}_\theta^+(p_i) \\ q_2 \in \mathcal{R}_{\theta_s}^+(p_s)}} d(\mathcal{P}(q_1), \mathcal{P}(q_2)) + \min_{\substack{q_1 \times q_2 \\ q_1 \in \mathcal{R}_\theta^-(p_i) \\ q_2 \in \mathcal{R}_{\theta_s}^-(p_s)}} d(\mathcal{P}(q_1), \mathcal{P}(q_2)) \right) \tag{7}$$

This potential is illustrated in Figure 2 (second row, second column) and Figure 3 (the red point shows the seed p_s). Contrary to the other potentials, it must be computed at each iteration of Algorithm 1.



(a) (b) (c)

Fig. 4: Number of times where a potential gives the minimum % of seeds (a) $w = 1$; (b) $w = 2$; (c) $w = 3$.

4 Experiments

4.1 User simulation for contour extraction evaluation

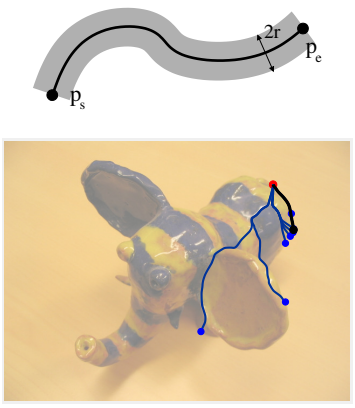
As introduced in [17], to reliably evaluate the effectiveness of approaches for contour extraction, one should ideally eliminate the user bias, while obtaining quality measures of the result. To cope with this, user agents have to be considered. These agents should be able to simulate the human user behavior through the addition of seeds close to the objects boundary. To do so, they have to evaluate the optimum-boundary segment being computed from a previously selected seed point to a virtual mouse position, seeking for the longest possible segment with minimum acceptable error. A new seed is then added when the error is too high and the agents iterate until closing the contour, just like real users. To achieve that, we have designed our own user agent dedicated to the livewire framework. A starting point p_s close to the ground truth object contour is selected using some criterion. In our case, the point has to be contained in a tube $\mathcal{T}$ of radius r on both sides of the contour (Fig. 5b). The virtual mouse position of the user is then simulated by finding a point p_e that satisfies three criteria. First it has to be located in tube $\mathcal{T}$. Second, the length of the path γ_{p_s,p_e} must be the largest one among all possible paths joining p_s and p_e. Third, all points of the curve γ_{p_s,p_e} must be in the tube $\mathcal{T}$. To do so, we perform a dichotomous search along the ground truth object contour (Fig. 5b). To close the contour, the last step of our simulation algorithm consists of trying to find a path γ_{p_s,p_e} that fulfills the third criterion. If it is not the case, point p_e is considered as the new seed starting point of the next iteration, as this is the case in the livewire framework (see Algorithm 1). The final contour being in a tube $\mathcal{T}$ of radius r on both sides of the contour, the contour extraction error is proportional to the radius of the tube: the larger r, the higher the contour extraction error.

4.2 Results

To demonstrate the ability of the proposed potentials to recover closed boundaries of objects in color images, we have carried out experiments on ISEG [8].

Potential	Radius	Mean % seeds	Mean Jaccard index
P_E	2	4.58	0.96
	3	2.96	0.95
	4	2.31	0.94
P_3^{max}	2	3.93	0.97
	3	2.67	0.95
	4	2.12	0.94
P_4	2	4.42	0.95
	3	3.13	0.93
	4	2.52	0.91
P_5	2	3.67	0.97
	3	2.38	0.95
	4	1.86	0.94

(a)



(b)

Fig. 5: (a) Quantitative comparison between potentials. (b) Tube of radius r surrounding a ground truth curve and process to find the next seed from the current one (red dot). The black curve is finally retained.

We ran Algorithm 1 for all the potentials with the user agent behavior. To have a fair comparison between the compared potentials, the first starting point is always the same on a given image. The criteria used to compare two potentials will be: i) the average of the ratio between the number of seeds and the length of the ground truth contour (the lesser the better, this will be denoted by % of seeds), ii) the Jaccard index that measures the similarity between the extracted contour and the ground truth. Since we use patch-based potentials, the first question that we can ask is what is the best patch size. We have considered three different patch sizes $w \in \{1, 2, 3\}$ that correspond to 3×3, 5×5 and 7×7 patches. For band-based potentials (i.e., P_3), the length of the band is set to $ns = 5$. For potentials that rely on an aggregation function (P_2 and P_3), we consider the min, max and mean aggregation functions. This means that we have three different configurations for each one of these potentials and this will be specified with an upperscript. Figure 4 shows an histogram that counts, on the ISEG dataset, the number of times that a potential gives the minimum % of seeds for different patch sizes. We do not show the Jaccard index since its average values are very close for all potentials (this was attended since we use simulated user agents). From these results, we can see that the best results are mostly always obtained with band-based potentials (with P_3^{max}). This shows the interest of going farther than the pixel surround and the proposed patch-based potentials have the ability to better delineate the object contour with less seeds when considering 5×5 patches. In the sequel we have retained this size.

Once this best patch size has been fixed, we compare the best potential we have proposed and tested so far, namely P_3^{max}, with the classical pixel-edge potential map [11] based on gradient magnitude and defined as $P_E = \mu +$

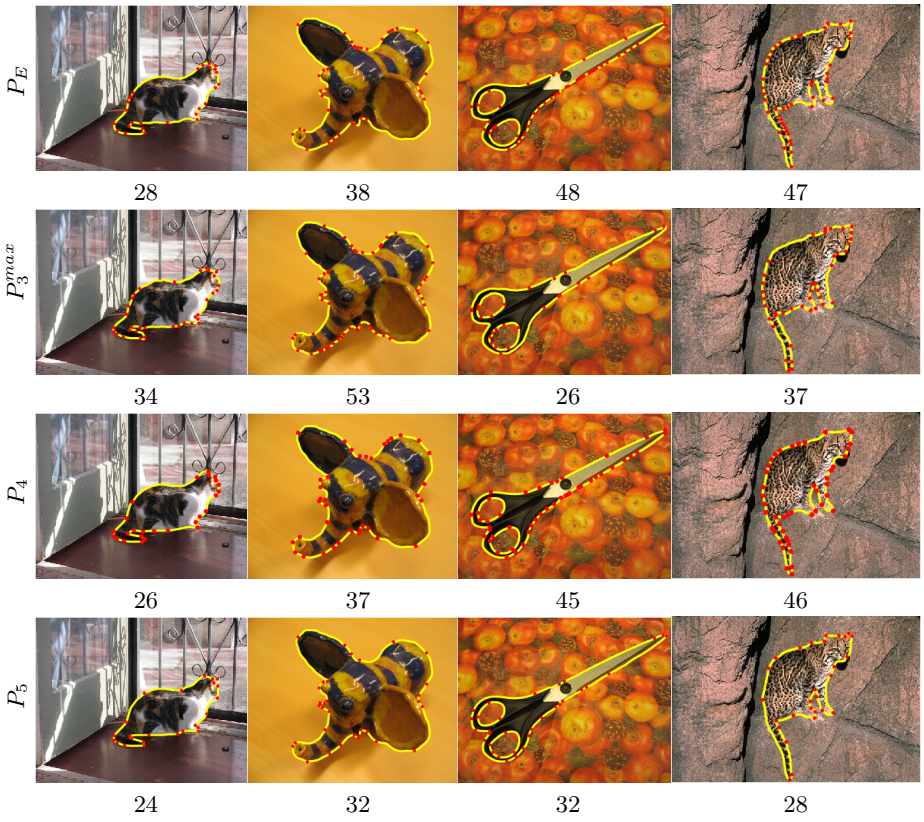


Fig. 6: Contour extraction results with different potentials (the number of required seed points is provided below each result).

$\max\left(0, \frac{||\nabla f(p_i||}{\max||\nabla f(p)||}\right)$ with $\mu = 0.2$. Whatever the agent configuration (different values of r), the potential P_3^{max} performs always better than the baseline P_E potential in terms of both criteria (% of seeds and Jaccard index). It is important to note that a difference of 0.1% on the average % of seeds corresponds to 1.5 points). This is shown in Fig. 5a. However, if the results are in favor of potential P_3^{max}, this is not the case for potential P_4. But for some images, P_4 performs better. This is illustrated in Figure 6 where less seeds are needed for the first image. Since no general rule can be devised to choose between potentials P_3^{max} and P_4, we choose to combine them with $P_5 = \alpha P_3^{\max} + (1-\alpha)P_4$. To choose the best α, we have tested values within $[0, 1]$ and retained the one that minimizes the % of seeds. We obtained a value of $\alpha = 0.3$. With this new potential P_5, one can see that we overpass by far the classical pixel-edge potential as well as all the patch-based potential we considered. In addition, this is true whatever the considered radius r of the user agent, which enforces the robustness of our proposed seed-dependent patch-based potential P_5.

5 Conclusion

In this paper new patch-based potentials have been proposed for minimal path contour extraction. Their interest has been assessed in terms of the number points needed to perform the contour extraction, so that the user interaction is potentially minimized. To eliminate the user bias in the interactive process, a dedicated user agent has been conceived. Results have shown that patch-based potentials can overcome the classical pixel-based ones and improve object contour extraction.

References

1. Blake, A., Rother, C.: Interactive image segmentation using an adaptive GMMRF model. In: ECCV (2004)
2. Bougleux, S., Peyré, G., Cohen, L.D.: Anisotropic geodesics for perceptual grouping and domain meshing. In: ECCV. vol. II, pp. 129–142 (2008)
3. Chen, D., Mirebeau, J.M., Cohen, L.D.: Global minimum for curvature penalized minimal path method. In: BMVC. pp. 86–86 (2015)
4. Cohen, L.D.: On active contour models and balloons. CVGIP-IMAG UNDERSTAN **53**(2), 211–218 (1991)
5. Cohen, L.D., Kimmel, R.: Global minimum for active contour models: A minimal path approach. IJCV **24**(1), 57–78 (1997)
6. Dijkstra, E.W.: A note on two problems in connexion with graphs. NUMER MATH pp. 269–271 (1959)
7. Falcão, A.X., Udupa, J.K., Miyazawa, F.K.: An ultra-fast user-steered image segmentation paradigm: live wire on the fly. IEEE T MED IMAGING **19**(1), 55–62 (2000)
8. Gulshan, V., Rother, C., Criminisi, A., Blake, A., Zisserman, A.: Geodesic star convexity for interactive image segmentation. In: CVPR. pp. 3129–3136 (2010)
9. Kang, H.W.: G-wire: A livewire segmentation algorithm based on a generalized graph formulation. PATTERN RECOGN LETT **26**(13), 2042–2051 (2005)
10. Liu, D., Xiong, Y., Shapiro, L., Pulli, K.: Robust interactive image segmentation with automatic boundary refinement. In: ICIP. pp. 225–228 (2010)
11. Mille, J., Bougleux, S., Cohen, L.D.: Combination of piecewise-geodesic paths for interactive segmentation. IJCV **112**(1), 1–22 (2015)
12. Miranda, P.A.V., Falcão, A.X., Spina, T.V.: Riverbed: A novel user-steered image segmentation method based on optimum boundary tracking. IEEE T IMAGE PROCESS **21**(6), 3042–3052 (2012)
13. Mortensen, E.N., Barrett, W.A.: Intelligent scissors for image composition. In: Proceedings of the 22Nd Annual Conference on Computer Graphics and Interactive Techniques. pp. 191–198 (1995)
14. Peyré, G., Pechaud, M., Keriven, R., Cohen, L.: Geodesic methods in computer vision and graphics. Foundations and Trends in Computer Graphics and Vision **5**(3-4), 197–397 (2010)
15. Rother, C., Kolmogorov, V., Blake, A.: Grabcut: Interactive foreground extraction using iterated graph cuts. ACM Trans. Graph. **23**(3), 309–314 (2004)
16. Sethian, J.A.: A fast marching level set method for monotonically advancing fronts. Proceedings of the National Academy of Sciences **93**(4), 1591–1595 (1996)
17. Spina, T.V., Falcão, A.X.: Robot users for the evaluation of boundary-tracking approaches in interactive image segmentation. In: ICIP. pp. 3248–3252 (2014)