Shortest enclosing walks with a non-zero winding number in directed weighted planar graphs: a technique for image segmentation

Alexey Malistov

ELVEES-NeoTek, Russia,
alexey@malistov.ru

Abstract. This paper presents an efficient graph-based image segmentation algorithm based on finding the shortest closed directed walks surrounding a given point in the image. Our work is motivated by the Intelligent Scissors algorithm, which finds open contours using the shortest-path algorithm, and the Corridor Scissors algorithm, which is able to find closed contours. Both of these algorithms focus on undirected, nonnegatively weighted graphs. We generalize these results to directed planar graphs (not necessary with nonnegative weights), which allows our approach to utilize knowledge of the object's appearance. The running time of our algorithm is approximately the same as that of a standard shortest-path algorithm.

1 Introduction

The shortest-paths algorithms are among the most widely used methods for image segmentation [1]. These methods have many applications in the field of medicine [2], optical character recognition systems [3], etc. The shortest-path segmentation is a so-called graph-based method. All graph-based methods represent the image as a weighted graph $G = (V, E)$, where each vertex $v \in V$ corresponds to a pixel in the image, the edges connect neighboring pixels [4]. Weights of the edges depend on the properties of the pixels, for instance, their intensities.

Mortensen and Barret [1] in 1995 was one of the first to present a new segmentation tool known as *Intelligent Scissors*. Their algorithm finds an optimal path from a start pixel to a goal pixel, where each pixel corresponds to a vertex in the graph G. Intelligent Scissors assign weights to edges according to the image gradient: the greater the image gradient, the lower the weight of the edge.

The shortest path from a start pixel s to the same pixel s is trivial. The shortest closed path surrounding a given point H can be considered as a natural generalization of a shortest-paths problem. The shortest closed paths are used as an image segmentation technique in [5–7]. In [5], a new segmentation tool *Corridor Scissors* is presented. Their tool searches for a shortest closed path inside of the corridor marked by a user. The corridor is considered as a ring-shaped graph. This ring-shaped graph is transformed into a lane-shaped graph by cutting along the shortest cut-path. In [7], a similar technique is used for image composition.

Previous works on closed paths are only focused on undirected, nonnegatively weighted graphs. In this paper we generalize these results. We give a direct algorithm for the shortest closed walks with the non-zero winding number n around a given point of the plane. Our algorithm applies to directed graphs with no negative-weight cycles. Any corridor can be emulated by removing vertices and edges outside the corridor. This generalization allows the algorithm to utilize knowledge of the objects appearance. Negative weights can be used to point a set of mandatory edges for the required walks. Directed graphs can use different weights for different directions, which makes it possible to find inner and outer contours (see fig. 1).

In fact, if the graph is undirected and the source vertex s is fixed, then no cut-path is needed. In [8], Provan found a simple algorithm for the shortest enclosing walk in undirected graphs. The thing is that the shortest enclosing walk can be easily produced from the shortest-paths tree. To find the shortest-paths tree Provan uses Dijksta's algorithm [9]. The implementation of Fredman and Tarjan [10] has a running time $O(E + V \log V)$. For the square grid, where $|E| = O(V)$, we obtain $O(V \log V)$. To date, there is an algorithm of the running time $O(V)$ for single-source shortest paths in planar graphs with nonnegative weights [11]. The shortest path in directed planar graphs with negative weights can be found in $O(V \log^2 V)$ time [12].

We recall that directed graphs can be useful to enclose an object if we know a priori color models of the object and background. Object always lies to the left of counterclockwise enclosing contour. Suppose that all pixels have integer coordinates. Let us shift each node in the above graph by $(0.5, 0.5)$. Then each node and each edge will lie between pixels. This aproach is used in [2]. Let O be an object segmented by the shortest closed walk p around the point $H \in O$. Whenever the walk p travels counterclockwise around the point H, pixels of the object O lie immediately to the left of p, in the traversal. Suppose e is an edge, e' is e given in reverse direction. Let ℓ be a pixel that lies immediately to the left of e (to the right of e'), r be a pixel that lies immediately to the right of e (to the left of e'). If the closed walk p contains e, then ℓ is inside of the object O. If the closed walk p contains e', then r is inside of the object O. To distinguish this we must have $w(e) \neq w(e')$ (see fig. 1). Thus, we obtain a directed graph.

2 Shortest Enclosing Walks in Directed Graphs

2.1 Background

Let $G = (V, E)$ be a weighted, directed graph with weight function $w\colon E \to \mathbb{R}$. Suppose $s, t \in G$. By $s \rightsquigarrow t$ denote an arbitrary walk from s to t. A closed walk is a walk such that its first and last vertices are the same. The weight $w(p)$ of walk p is the sum of the weights of the edges of p. Sometimes the word *length* is used instead of weight. But we reserve *length* for the number of edges in the

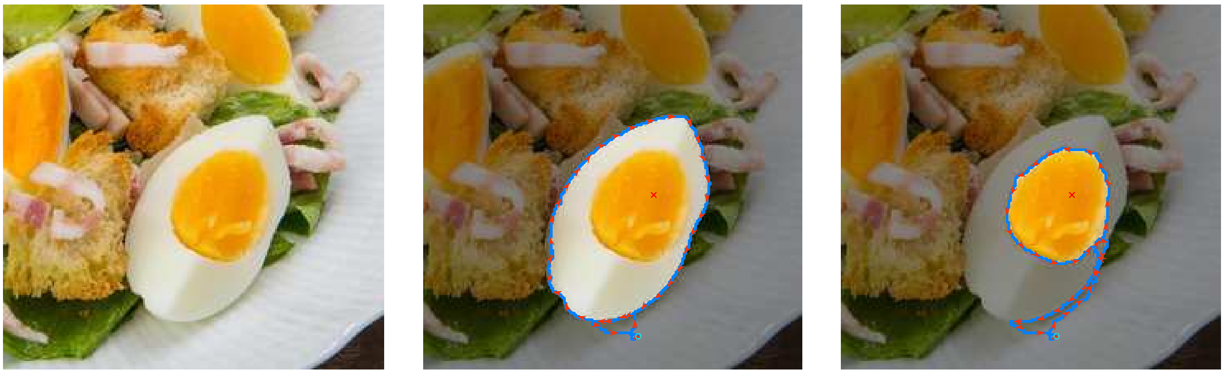

Fig. 1. Shortest enclosing directed walks: left) original image; center) the shortest clockwise walk (winding number $= -1$); right) the shortest counterclockwise walk (winding number $= +1$). Note that the clockwise walk finds the outer contour of the egg, while the counterclockwise walk find the inner contour (that of the yoke).

walk. In the sequel, only walks of finite length are considered. A shortest walk from vertex s to vertex t is any walk $s \rightsquigarrow t$ with minimal weight. Shortest walks are well defined if and only if the graph G contains no negative-weight cycles. It follows that if the graph is undirected, all weights are non-negative. In the sequel, only graphs without negative-weight cycles are considered. Any shortest walk cannot contain a positive-weight cycle. This cycle can be removed to create a walk with a lower weight. We can also remove 0-weight cycles to create a walk with the same weight. Without loss of generality we can assume that all shortest walks have no cycles. Thus all shortest walks are simple paths. In particular, the shortest path from s to s is the path that contains no edges.

For any vertex s there is a *shortest-paths tree* $T_s = (V_s, E_s)$ such that 1) T_s is a subgraph of G, 2) s is the root of T_s, 3) V_s contains vertices reachable from s, 4) for all $v \in V_s$, the path from root s to v in T_s is the shortest path from s to v in G.

Suppose G has a fixed planar embedding in the plane $P = \mathbb{R}^2$ and this embedding is given by some map $f\colon V \to P$. All edges intersect only at endpoints. By (x_v, y_v) we denote the coordinates of vertex v. Suppose that the plane P has a distinguished point H. We will assume that no edge intersects H. In the converse case, remove all those edges. To each pair (u, v), where $u, v \in V$, assign the directed angle $\theta(u, v) = \measuredangle UHV$ where $U = f(u)$, $V = f(v)$. We have $-2\pi < \theta(u, v) \le 2\pi$ and $\theta(v, u) = -\theta(u, v)$. Suppose $e \in E$ is an edge of G. By defenition, put $\alpha(e) = \theta(u, v)$. A *winding angle* of walk $\alpha(p)$ is the sum of the angles $\alpha(e)$ of the edges of p.

Suppose $s, t \in V$; then it is not hard to prove that for any walk $s \rightsquigarrow t$ there exists a unique $n \in \mathbb{Z}$ such that $\alpha(s \rightsquigarrow t) = \theta(s, t) + 2\pi n$. To do this, one can use the polar coordinate system with the origin at H. In particular, for any closed walk $s \rightsquigarrow s$ we get

$$\alpha(s \rightsquigarrow s) = 2\pi n, \quad n \in Z \ . \tag{1}$$

$r(p) = \alpha(p)/2\pi$ is called the *winding number* of walk p. It follows from (1) that the winding number of a closed walk is always integer. If the shortest-paths tree

T_s is fixed, then for all $v \in V_s$ there exists a unique path $q = s \rightsquigarrow v$ in T_s. By $\beta(s, v)$ we denote the winding angle of q.

We recall that the shortest closed walk is a zero-length path. But the shortest closed walk with non-zero winding number is nontrival. In the sequel, we shall focus on the problem of the shortest closed walk of non-zero winding number. Given a weighted digraph $G = (V, E)$ embedded in the plane P, let H be a point of the plane P, let $s \in V$ be a source vertex. We want to find a shortest closed walk $s \rightsquigarrow s$ with a given winding number $n \neq 0$, $n \in \mathbb{Z}$, around H.

2.2 Undirected Graphs

Let p^{-1} be a walk p given in reverse direction. If G is undirected graph, then we obtian $w(p^{-1}) = w(p)$. It follows that there exist a shortest closed walk of winding number $+n$ if and only if there exists a shortest closed walk of winding number $-n$. Moreover, the weights of these walks are the same. In the next subsection we will show that the shortest closed walk with non-zero winding number has winding number ± 1 (see Theorem 5).

Provan [8] gave the first algorithm for finding nontrivial walks in undirected, nonnegatively weighted graphs with a fixed (not necessary planar) embedding in the plane. His algorithm finds a shortest closed walk surrounding a given obstacle O in the plane. This shortest walk has non-zero winding number. But we cannot choose an arbitrary winding number n.

Provan considers a plane embedding that is not necessary planar. For planar graphs, Provan's algorithm gives a closed walk of winding number ± 1.

2.3 Directed Graphs

For directed graphs, $w(p^{-1})$ is not necessarily equal to $w(p)$. Moreover, the shortest closed walk with winding number $+1$ and the shortest closed walk with winding number -1 may be distinct.

Consider a point $F = (x, y, z) \in \mathbb{R}^3$. By definition, put $x(F) = x$, $y(F) = y$, $z(F) = z$. Fix the source vertex s. Take a vertex $v \in V$. Let $\mathcal{H}_v$ be a set given by

$$\mathcal{H}_v = \{h \in \mathbb{R}^3 \mid x(h) = x_v, y(h) = y_v, z(h) = \theta(s, v) + 2\pi n, n \in \mathbb{Z}\} \ .$$

For example, since $\theta(s, s) = 0$, it follows that $\mathcal{H}_s = \{(x_s, y_s, 2\pi n) \mid n \in \mathbb{Z}\}$. Put $\mathcal{V} = \bigcup_{v \in V} \mathcal{H}_v$. Let g be the map from $\mathcal{V}$ to V taking $h \in \mathcal{H}_v$ to v. Consider a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where

$$\mathcal{E} = \big\{(h_1, h_2) \mid h_1, h_2 \in \mathcal{V}, \big(g(h_1), g(h_2)\big) \in E, z(v) = z(u) + \theta\big(g(h_1), g(h_2)\big)\big\} \ .$$

For each edge $(h_1, h_2) \in \mathcal{E}$, we have a weight $w(h_1, h_2) = w\big(g(h_1), g(h_2)\big)$ The graph $\mathcal{G}$ can be embedded in the helicoid (see fig. 2). Consider $S = (x_s, y_s, 0) \in \mathcal{H}_s \subset \mathcal{V}$, where $s \in V$ is the fixed source vertex in G. Clearly, $g(S) = s$. If we take another source vertex s_2, then we get another graph $\mathcal{G}_2$. To simplify the notation we write $\mathcal{G}$ instead of $\mathcal{G}_s$. In the sequel, s is fixed.

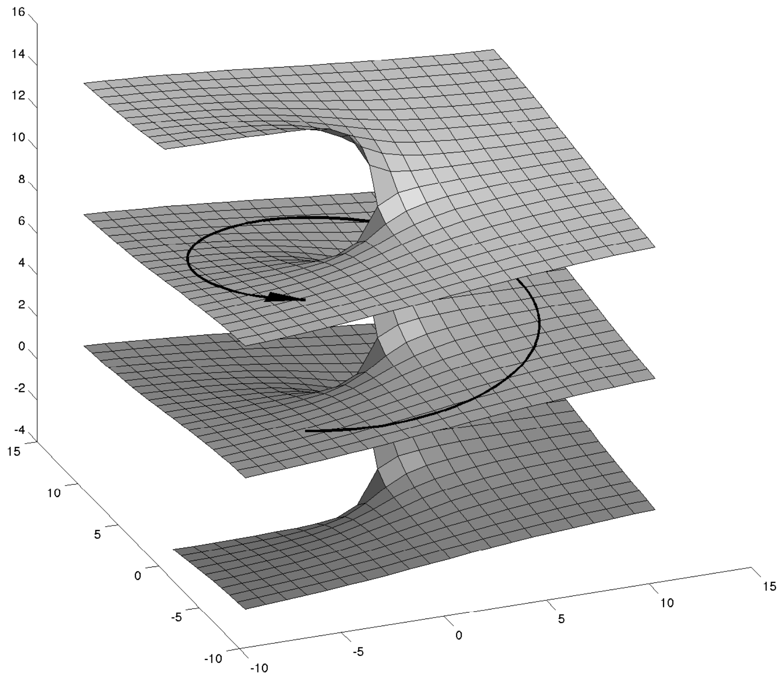

Fig. 2. Helicoid graph.

Lemma 1. *Let W_s be a set of walks from s in the graph G, $\mathcal{W}_S$ be a set of walks from S in the graph $\mathcal{G}$. Then W_s is isomorphic to $\mathcal{W}_S$ with respect to weight.*

Proof. Let $p = \langle sv_1 \dots v_\ell \rangle$ be a walk in G. Consider the path $P = \langle SV_1 \dots V_\ell \rangle$ in the graph $\mathcal{G}$, where $V_i = \big(x_{v_i}, y_{v_i}, \alpha(sv_1v_2 \dots v_i)\big)$, $\alpha(sv_1v_2 \dots v_i)$ is the winding angle of the subwalk $s \rightsquigarrow v_i$. Since $\alpha\big(sv_1 \dots v_iv_{i+1}\big) - \alpha\big(sv_1 \dots v_i\big) = \theta(v_i, v_{i+1})$, we get $(V_i, V_{i+1}) \in \mathcal{E})$ and $w(V_i, V_{i+1}) = w(v_i, v_{i+1})$. It follows that $w(P) = w(p)$.

Now let $P = \langle SV_1V_2 \dots V_\ell \rangle$ be a walk in $\mathcal{G}$. Simalarly, consider the walk $p = \langle sv_1v_2 \dots v_\ell \rangle$, where $v_i = g(V_i)$. The walk p corresponds to P and has the same weight. This completes the proof of Lemma 1. $\square$

Let I be the isomorphism from Lemma 1, W_s^n be the set of all closed walks in G from s with winding number $n \neq 0$ around H. Then $I(W_s^n)$ is the set of all open walks in $\mathcal{G}$ from $S = (x_s, y_s, 0)$ to $S_n = (x_s, y_s, 2\pi n)$. Thus, the shortset closed walk problem in graph G is equivalent to the shortset path problem in graph $\mathcal{G}$. No well-known algorithm can be started because the graph $\mathcal{G}$ is infinite. Our aim is to find some subgraph $\mathcal{G}'$ of $\mathcal{G}$ that is finite. Then we can use any well-known shortest-path algorithm.

The first approach is to remove all vertices $h \in \mathcal{V}$ such that $|z(h)| < 4\pi|V|$. This is a good idea because the walks of a length more than $2|V|$ are not very useful. The shortcoming of this method is that the number of vertices in $\mathcal{G}$ is $|\mathcal{V}| = \Theta(|V|^2)$. This method is not optimal. We will show that there is a better way to find a finite subgraph.

2.4 A Finite Subgraph of $\mathcal{G}$

We recall that $\beta(s,v)$ is the winding angle of the unique path q in the shortest-paths tree T_s. For each vertex $v \in G$ there exists a unique vertex $h \in \mathcal{V}$ such that $h = (x_v, y_v, \beta(s,v))$. Now we shall give the following definitions.

Definition 1. *Suppose $v \in V$. Then the vertex $(x_v, y_v, \beta(s,v)) \in \mathcal{V}$ is called the shortest representative of v in the helicoid graph $\mathcal{G}$ with respect to the source vertex s.*

Since s is fixed, "with respect to the source vertex s" will be omitted.

Definition 2. *We say that the vertex $h = (x_v, y_v, z) \in \mathcal{V}$ has a tier m and write $\operatorname{tier}(h) = m$ if $z = \beta(s,v) + 2\pi m$, where $v = g(h) \in V$ is the corresponding vertex in G.*

Clearly, all shortest representatives in $\mathcal{G}$ have the tier 0.

Theorem 1 (intermediate value). *For any walk $p = h_1 \rightsquigarrow h_2$ in $\mathcal{G}$ the tier takes any value between $\operatorname{tier}(h_1)$ and $\operatorname{tier}(h_2)$ at some vertex of p.*

Proof. Let (u', v') be an edge of the walk p. Put $u = g(u')$, $v = g(v')$. Then $z(u') = \beta(s,u) + 2\pi \cdot \operatorname{tier} u'$, $z(v') = \beta(s,v) + 2\pi \cdot \operatorname{tier} v'$. We recall that $z(v') - z(u') = \theta(u,v)$. It follows that

$$z(v') - z(u') = \theta(u,v) = \beta(s,v) - \beta(s,u) + 2\pi \cdot (\operatorname{tier} v' - \operatorname{tier} u'),$$

$$\operatorname{tier} v' - \operatorname{tier} u' = \frac{\beta(s,u) + \theta(u,v) - \beta(s,v)}{2\pi} = r(s \rightsquigarrow u \to v \rightsquigarrow s),$$

where $s \rightsquigarrow u \to v \rightsquigarrow s$ is the closed curve which contains the shortest path $s \rightsquigarrow u$, the edge $u \to v$, and the reverse of the shortest path $s \rightsquigarrow v$. This curve has no self-intersections because $s \rightsquigarrow u$ and $s \rightsquigarrow v$ are the paths in the tree T_s. Thus, the winding number $r(s \rightsquigarrow u \to v \rightsquigarrow s) \in \{0, \pm 1\}$. It follows that either the tiers of the neighboring vertices u', v' are the same or $\operatorname{tier}(v') - \operatorname{tier}(u') = \pm 1$.

Theorem 2 (tiers). *Suppose $h \in \mathcal{V}$, $\operatorname{tier}(h) \geq 0$. Then there exists a shortest path p from S to h such that for any vertex u of p we have $\operatorname{tier}(u) \geq 0$.*

Proof. For $\operatorname{tier}(h) = 0$, by definition (2), the shortest paths induced by the shortest-paths tree T_s contain only vertices of the tier 0. For $\operatorname{tier}(h) > 0$, assume the converse. Then some shortest path $\langle S \ldots u \ldots h \rangle$ contains u such that $\operatorname{tier}(u) < 0$. By Theorem 1, there exists a vertex v between u and h such that $\operatorname{tier}(v) = 0$. Replace $\langle S \ldots u \ldots v \rangle$ with the shortest path $S \rightsquigarrow v$ that contains only vertices with the tier 0. Repeating this operation we obtain the shortest path p from S to h such that for any vertex u of p we have $\operatorname{tier}(u) \geq 0$. This contradiction proves the theorem. $\square$

Similarly, there exists a shortest path p from the vertex S to a vertex $h \in \mathcal{V}$, $\operatorname{tier}(h) \leq 0$, such that for any vertex u of p we have $\operatorname{tier}(u) \leq 0$.

Corollary 1. *If we want to find a shortest closed walk with the winding number $n > 0$, then any vertex h with $\mathrm{tier}(h) < 0$ can be removed. If we want to find a shortest closed walk with the winding number $n < 0$, then any vertex h with $\mathrm{tier}(h) > 0$ can be removed.*

Consider a vertex $v \in V$ in the input graph and the corresponding set of vertices $\mathcal{H}_v \subset \mathcal{V}$ in the graph $\mathcal{G}$. We can sort all vertices of $\mathcal{H}_v$ by their shortest distance from S. Evidently, the shortest of the shortest paths from the vertex S to the set $\mathcal{H}_v$ is the path to a vertex $h \in \mathcal{H}_v$ such that $\mathrm{tier}(h) = 0$.

Definition 3. *For $R \geq 0$, we shall say that a vertex $h \in \mathcal{H}_v$ has a rank R and write $\mathrm{rank}(h) = R$ if the shortest distance from S to h is the $(R+1)$-th minimum of the shortest distances from S to the set $\mathcal{H}_v$.*

Theorem 3 (ranks). *Suppose $h \in \mathcal{V}$.Then there exists a shortest path p from S to h such that for any vertex u of p it follows that $\mathrm{rank}(u) \leq \mathrm{rank}(h)$.*

Proof. Assume the converse. Then some shortest path $p = \langle S \dots u \dots h\rangle$ contains u such that $\mathrm{rank}(u) > R$, where $R = \mathrm{rank}(h)$. Consider a set $U = \{u_k\}_{k=0}^{R}$ such that $u_k \in \mathcal{V}$, $\mathrm{rank}(u_k) = k$, and $g(u_k) = g(u)$. We can replace $\langle S \dots u\rangle$ with $S \rightsquigarrow u_k$. Then there exist $R+1$ walks $\langle S \dots u_k \dots\rangle_{k=0}^{R}$ with a lower weight. All end nodes h_k are distinct because $\forall i \neq j\ \ \mathrm{tier}(h_i) \neq \mathrm{tier}(h_j)$. Since $\forall k\ \ \mathrm{tier}(u_k) \neq \mathrm{tier}(u)$, we get $\forall k\ \ \mathrm{tier}(h_k) \neq \mathrm{tier}(h)$. We have that $w(p)$ is not the $(R+1)$-th minimum of the shortest distances from S to $\mathcal{H}_{g(h)}$. This contradicts Definition 3. The theorem is proved. $\square$

Definition 4. *Suppose $\mathcal{H}_v^+ = \{h \in \mathcal{H}_v \mid \mathrm{tier}(h) > 0\}$, $\mathcal{H}_v^- = \{h \in \mathcal{H}_v \mid \mathrm{tier}(h) < 0\}$. We shall say that a vertex $h \in \mathcal{H}_v^+$ has a category $K > 0$ and write $\mathrm{cat}(h) = K$ if the shortest distance from S to h is the K-th minimum of the shortest distances from S to the set $\mathcal{H}_v^+$. We shall say that a vertex $h \in \mathcal{H}_v^-$ has a category $K < 0$ if the shortest distance from S to h is the $|K|$-th minimum of the shortest distances from S to the set $\mathcal{H}_v^-$. If $\mathrm{tier}(h) = 0$, we put $\mathrm{cat}(h) = 0$.*

Using Theorem 2, Theorem 3 and Definition 4, we get the following theorem.

Theorem 4 (categories). *Suppose $h \in \mathcal{V}$. Then there exists a shortest path p from S to h such that for any vertex u of p it follows that $|\mathrm{cat}(u)| \leq |\mathrm{cat}(h)|$.*

Theorem 5 (Main). $\forall h \in \mathcal{V}\ \ \mathrm{tier}(h) = \mathrm{cat}(h)$.

Proof. The cases $\mathrm{cat}(h) \geq 0$ and $\mathrm{cat}(h) \leq 0$ are equivalent. Without loss of generality it can be assumed that $k = \mathrm{cat}(h) \geq 0$. The proof is by induction over k. For $k = 0$, there is nothing to prove. For $k > 0$, assume the converse. Then there exist $h \in \mathcal{V}$ such that $\mathrm{cat}(h) = k$ and $\mathrm{tier}(h) \neq k$. Let $h_m \in \mathcal{H}_{g(h)}^+$ be a vertex such that $\mathrm{tier}(h_m) = m$. By the inductive assumption, $\forall m < k\ \ \mathrm{cat}(h_m) = \mathrm{tier}(h_m) = m$. If $\mathrm{tier}(h) < k$, then $\mathrm{cat}(h) = \mathrm{cat}(h_{\mathrm{tier}(h)}) = \mathrm{tier}(h)$. Thus, $\mathrm{tier}(h) > k$. Consider the shortest path $p = S \rightsquigarrow h$. Also, consider the first vertex v in p such that $\mathrm{tier}(v) > k$. Clearly, $\mathrm{cat}(v) = k$. If $\mathrm{cat}(v) < k$, then $\mathrm{cat}(v) = \mathrm{tier}(v)$. Let u be the predecessor of v. By Theorem 4, $\text{cat}(u) \leq \text{cat}(h) = k$. If $\text{cat}(u) < k$, then $\text{tier}(u) = \text{cat}(u) < k$. Thus, $\text{tier}(v) - \text{tier}(u) \geq 2$. This contradicts Theorem 1. It follows that $\text{cat}(u) = \text{tier}(u) = k$ and $\text{tier}(v) = k + 1$. Let u_{k-1} be the vertex such that $\text{cat}(u_{k-1}) = k - 1$ and $g(u_{k-1}) = g(u)$. By the inductive assumption, $\text{tier}(u_{k-1}) = k - 1$. Replace the subpath $S \rightsquigarrow u$ with the shorter path $S \rightsquigarrow u_{k-1}$ ($\text{cat}(u_{k-1}) < \text{cat}(u)$). Then we obtain the shorter path from S to a new vertex v' such that $\text{tier}(v') = k$ and $g(v') = g(v)$. It follows that $\text{cat}(v') = k$ and $\text{cat}(v) \neq k$.

Corollary 2. *If we want to find a shortest closed walk with the winding number n, then any vertex h with $|\,\text{tier}(h)| > n$ can be removed.*

2.5 An Algorithm For Finding Shortest Enclosing Walks

Corollary 1 and corollary 2 give the following algorithm. Suppose s is the fixed source vertex, n is the winding number.

1. Find the shortest-paths tree T_s.
2. For all $u \in V$ calculate $\beta(s, u)$.
3. Create the finite subgraph $\mathcal{G}' = (\mathcal{V}', \mathcal{E}')$ of $\mathcal{G}$ such that $h \in \mathcal{V}'$ if and only if $\text{sign}(\text{tier}(h)) = \text{sign}(n)$, $|\,\text{tier}(h)| \leq n$, or $\text{tier}(h) = 0$.
4. Find the shortest path from $(x_s, y_s, 0)$ to $(x_s, y_s, 2\pi n)$ in $\mathcal{G}'$.

The running time of our algorithm is $O(nT)$, where T is the running time of the shortest-path algorithm in steps 1 and 4. In particular, for $n = 1$ the running time is $O(T)$.

3 Evaluation

Experimental results are presented in fig. 3. We put the center point H inside the object. Also we put the start point S somewhere outside object and run our algorithm looking for the shortest enclosing walk surrounding the point H. Note that the start point S can be placed far from enclosing object. We use some pictures from The Berkeley Segmentation Dataset [13].

Consider the picture with the bear. Note that the segmentation region is approximately the same for the different start points. Now consider the picture with flowers. We put the center point H inside the left flower. Start point are placed in the background. The shortest clockwise and counterclockwise walks are different. First one helps to find the inner contour inside the left flower. Second one helps to cut flowers.

The special line marked “Tier cut” in the pictures cuts the edges where the tier of the vertices changes. If edge intersects the tier-cut-line, then this edge has vertices with different tiers (see the proof of Theorem 1).

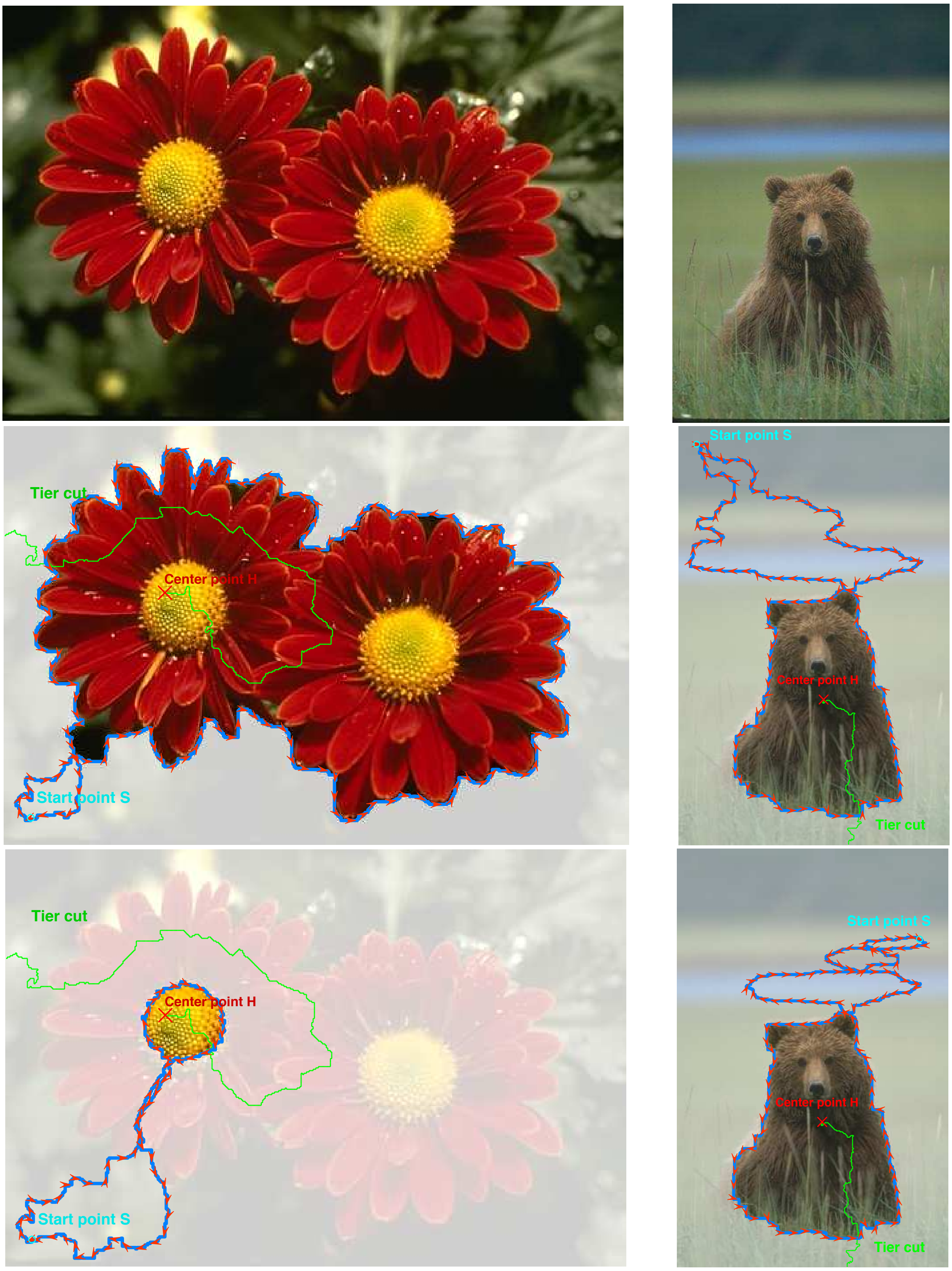

Fig. 3. Experimental results: original image and a segmentation based on the shortest enclosing walk (clockwise and counterclockwise). The center point H is inside the object considered. Note that the start point S is far from enclosing object. "Tier cut" is the line where tier changes (see the proof of Theorem 1). Pictures are from The Berkeley Segmentation Dataset [13].

4 Conclusion and future work

We have presented a new segmentation algorithm which is based on finding a shortest enclosing walk in the directed planar graphs. This walk surrounds a given point of the plane and has a given winding number $n \neq 0$. Our approach generalizes previous works which search for either open simple walks or closed walks in undirected graphs with nonnegative weights. The method runs in $O(nT)$ time, where T is the running time of a regular shortest-path algorithm. For $n = 1$, the running time is $O(T)$.

For future work, we plan to extend our approach to a number of points we want to surround. Also, we want to use an image pyramid representation to assign the weights of the edges according to the information about pixels that does not immediately lie to the left or to the right of closed walks.

References

1. Mortensen, E.N., Barrett, W.A.: Intelligent scissors for image composition. In: Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques. SIGGRAPH '95, New York, NY, USA, ACM (1995) 191–198
2. Falcao, A., Udupa, J., Miyazawa, F.: An ultra-fast user-steered image segmentation paradigm: live wire on the fly. Medical Imaging, IEEE Transactions on **19** (2000)
3. Tse, J., Jones, C., Curtis, D., Yfantis, E.: An OCR-independent character segmentation using shortest-path in grayscale document images. Sixth International Conference on Machine Learning and Applications (ICMLA 2007) (2007) 142–147
4. Felzenszwalb, P.F., Huttenlocher, D.P.: Efficient graph-based image segmentation. Int. J. Comput. Vision **59** (2004) 167–181
5. Farin, D., Pfeffer, M., de With, P., Effelsberg, W.: Corridor scissors: a semi-automatic segmentation tool employing minimum-cost circular paths. In: Image Processing, 2004. ICIP '04. Intnl. Conf. on. Volume 2. (2004) 1177–1180 Vol.2
6. Farin, D., N., P.H.: Shortest circular paths on planar graphs. In: In 27 th Symposium on Information Theory in the Benelux. (2006) 117–124
7. Jia, J., Sun, J., Tang, C.K., Shum, H.Y.: Drag-and-drop pasting. ACM SIGGRAPH (2006) 631
8. Provan, J.S.: Shortest enclosing walks and cycles in embedded graphs. Inf. Process. Lett. **30** (1989) 119–125
9. Dijkstra, E.: A note on two problems in connexion with graphs. Numerische mathematik (1959) 269–271
10. Fredman, M.L., Tarjan, R.E.: Fibonacci heaps and their uses in improved network optimization algorithms. J. ACM **34** (1987) 596–615
11. Klein, P., Rao, S., Rauch, M., Subramanian, S.: Faster shortest-path algorithms for planar graphs. In: Proceedings of the Twenty-sixth Annual ACM Symposium on Theory of Computing. STOC '94, New York, NY, USA, ACM (1994) 27–37
12. Klein, P.N., Mozes, S., Weimann, O.: Shortest paths in directed planar graphs with negative lengths: A linear-space $O(n \log^2 n)$-time algorithm. ACM Trans. Algorithms **6** (2010) 30:1–30:18
13. Martin, D., Fowlkes, C., Tal, D., Malik, J.: A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In: 8th Int'l Conf. Computer Vision. Volume 2. (2001) 416–423