Shape-based Image Representation (Part II)

Alexandre Xavier Falcão

Institute of Computing - UNICAMP

afalcao@ic.unicamp.br

Shape-based image representation

- Image superpixels and objects may be represented by their contours, skeletons, and salience points.

Shape-based image representation

- Image superpixels and objects may be represented by their contours, skeletons, and salience points.
- Contours have internal and external skeletons.

Shape-based image representation

- Image superpixels and objects may be represented by their contours, skeletons, and salience points.
- Contours have internal and external skeletons.
- The concave and convex saliences of a contour are related to its external and internal skeletons, respectively.

Shape-based image representation

- Image superpixels and objects may be represented by their contours, skeletons, and salience points.
- Contours have internal and external skeletons.
- The concave and convex saliences of a contour are related to its external and internal skeletons, respectively.
- Salience points of superpixels are strongly related to feature points for image matching.

Agenda

- Multiscale contours.
- Multiscale skeletons.
- Salience points.

Multiscale contours

- Let $\mathcal{S} \subset \mathcal{B}$ be a set of contour pixels extracted from a border set $\mathcal{B}$, such that $L_{ct}(p)$ is known for $p \in \mathcal{S}$ (previous lecture).

Multiscale contours

- Let $\mathcal{S} \subset \mathcal{B}$ be a set of contour pixels extracted from a border set $\mathcal{B}$, such that $L_{ct}(p)$ is known for $p \in \mathcal{S}$ (previous lecture).

- Let $(D_I, \mathcal{A}_r)$ be a non-weighted graph for the binary image $\hat{I} = (D_I, I)$ and adjacency relation $\mathcal{A}_r$.

Multiscale contours

- Let $\mathcal{S} \subset \mathcal{B}$ be a set of contour pixels extracted from a border set $\mathcal{B}$, such that $L_{ct}(p)$ is known for $p \in \mathcal{S}$ (previous lecture).
- Let $(D_I, \mathcal{A}_r)$ be a non-weighted graph for the binary image $\hat{I} = (D_I, I)$ and adjacency relation $\mathcal{A}_r$.
- Let f_{edt} be a connectivity function defined as

$$
\begin{aligned}
f_{edt}(\langle p \rangle) &= \begin{cases} 0 & \text{if } p \in \mathcal{S}, \\ +\infty & \text{otherwise.} \end{cases} \\
f_{edt}(\pi_p \cdot \langle p, q \rangle) &= \left(\| q - R(p) \|_2 \right)^2,
\end{aligned}
$$

where $R(p) \in \mathcal{S}$ is the root pixel of π_p.

Multiscale contours

- Let $\mathcal{S} \subset \mathcal{B}$ be a set of contour pixels extracted from a border set $\mathcal{B}$, such that $L_{ct}(p)$ is known for $p \in \mathcal{S}$ (previous lecture).

- Let $(D_I, \mathcal{A}_r)$ be a non-weighted graph for the binary image $\hat{I} = (D_I, I)$ and adjacency relation $\mathcal{A}_r$.

- Let f_{edt} be a connectivity function defined as

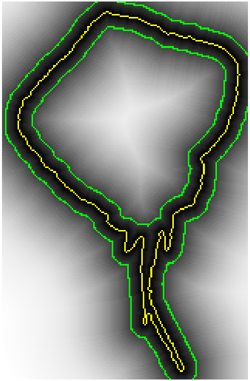
$$
\begin{aligned}
f_{edt}(\langle p \rangle) &= \begin{cases} 0 & \text{if } p \in \mathcal{S}, \\ +\infty & \text{otherwise.} \end{cases} \\
f_{edt}(\pi_p \cdot \langle p, q \rangle) &= (\|q - R(p)\|_2)^2,
\end{aligned}
$$

  where $R(p) \in \mathcal{S}$ is the root pixel of π_p.

- For some $r \geq \sqrt{2}$, the minimization of the cost map $V(p) = \min_{\pi_p \in \Pi}\{f_{edt}(\pi_p)\}$ assigns to each pixel $p \in D_I$ the squared Euclidean distance to its closest pixel $R(p) \in \mathcal{S}$.

Multiscale contours

This transformation is named Euclidean distance transform of $\mathcal{S}$.



When $\mathcal{S}$ contains a single contour, it creates in the connectivity map V multiscale contours (iso-contours) by subsequent exact dilations and erosions of $\mathcal{S}$ [1].

The IFT algorithm for f_{edt} can execute the EDT in $O(|D_I|)$ for small values of $r \geq \sqrt{2}$.

Euclidean distance transform (EDT)

The IFT algorithm for f_{edt} can execute the EDT in $O(|D_I|)$ for small values of $r \geq \sqrt{2}$.

Input:	Binary image $\hat{I} = (D_I, I)$, seed set $\mathcal{S}$, and adjacency relation $\mathcal{A}_r$.
Output:	Connectivity map V.
Auxiliary:	Priority queue $\mathcal{Q}$ based on bucket sort, root map R, and variable *tmp*.

Euclidean distance transform (EDT)

The IFT algorithm for f_{edt} can execute the EDT in $O(|D_I|)$ for small values of $r \geq \sqrt{2}$.

Input:	Binary image $\hat{I} = (D_I, I)$, seed set $\mathcal{S}$, and adjacency relation $\mathcal{A}_r$.
Output:	Connectivity map V.
Auxiliary:	Priority queue $\mathcal{Q}$ based on bucket sort, root map R, and variable *tmp*.

For contours and surfaces, $r = \sqrt{2}$ and $r = \sqrt{3}$ are usually enough, respectively.

Euclidean distance transform (EDT)

For each $p \in D_I$, do.

  Set $V(p) \leftarrow +\infty$.

  If $p \in \mathcal{S}$ then set $V(p) \leftarrow 0$ and $R(p) \leftarrow p$.

  Insert p in $\mathcal{Q}$.

While $\mathcal{Q} \neq \emptyset$, do.

  Remove p from $\mathcal{Q}$ such that $p = \arg\min_{q \in \mathcal{Q}}\{V(q)\}$.

  For each $q \in \mathcal{A}_r(p)$ such that $V(q) > V(p)$, do.

    Set $tmp \leftarrow (\|q - R(p)\|_2)^2$.

      If $tmp < V(q)$, then.

        Set $V(q) \leftarrow tmp$ and $R(q) \leftarrow R(p)$.

Euclidean distance transform (EDT)

For each $p \in D_I$, do.
    Set $V(p) \leftarrow +\infty$.
    If $p \in \mathcal{S}$ then set $V(p) \leftarrow 0$ and $R(p) \leftarrow p$.
    Insert p in $\mathcal{Q}$.
While $\mathcal{Q} \neq \emptyset$, do.
    Remove p from $\mathcal{Q}$ such that $p = \arg\min_{q \in \mathcal{Q}}\{V(q)\}$.
    For each $q \in \mathcal{A}_r(p)$ such that $V(q) > V(p)$, do.
        Set $tmp \leftarrow (\|q - R(p)\|_2)^2$.
            If $tmp < V(q)$, then.
                Set $V(q) \leftarrow tmp$ and $R(q) \leftarrow R(p)$.

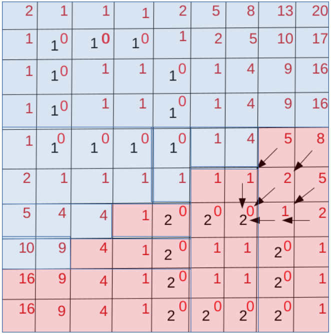
Note that, we may use the values in I to constrain computation inside an object, outside it, and/or up to a threshold $V(p) \leq T$.

Euclidean distance transform (EDT)

	1^{0}	1^{0}	1^{0}					
	1^{0}			1^{0}				
	1^{0}			1^{0}				
	1^{0}	1^{0}	1^{0}	1^{0}				
				2^{0}	2^{0}	2^{0}		
				2^{0}			2^{0}	
				2^{0}			2^{0}	
				2^{0}	2^{0}	2^{0}	2^{0}	

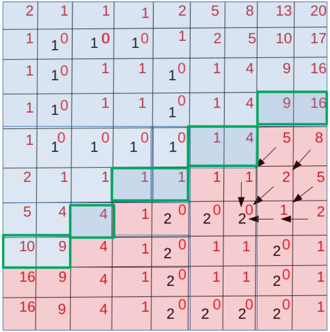
The example shows two contours with pixel labels 1 or 2 (black) and initial costs 0 (red). The cost of the remaining pixels is $+\infty$.

Euclidean distance transform (EDT)



The EDT algorithm propagates the minimum cost to all pixels (red). It can be easily modified to propagate the contour labels in L_{ct} (colored regions) and the optimum-path tree of each contour pixel (arrows) in a predecessor map P.

Euclidean distance transform (EDT)



A skeleton by influence zones (SKIZ) can be defined by pixels p (green) such that it exits a 4-neighbor $q \in \mathcal{A}_1(p)$ with $L_{ct}(q) > L_{ct}(p)$.

Multiscale skeletons

- Each contour is related to its internal and external skeletons (medial axes)— point sets with at least two equidistant pixels on the contour.

Multiscale skeletons

- Each contour is related to its internal and external skeletons (medial axes)— point sets with at least two equidistant pixels on the contour.
- By painting a ball of radius $\sqrt{V(p)}$ for each skeleton pixel p, one can reconstruct the shape. Filtered skeletons imply filtered shapes.

Multiscale skeletons

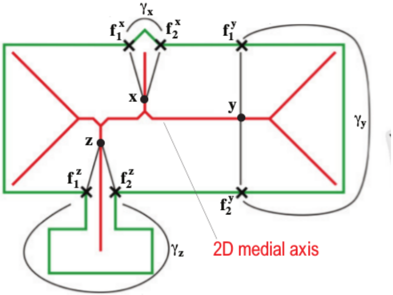
- Each contour is related to its internal and external skeletons (medial axes)— point sets with at least two equidistant pixels on the contour.
- By painting a ball of radius $\sqrt{V(p)}$ for each skeleton pixel p, one can reconstruct the shape. Filtered skeletons imply filtered shapes.
- From each contour in $\mathcal{S}$, one can create internal and external multiscale skeletons which are one-pixel-wide and connected in all scales.

Multiscale skeletons

- Let n_k be the number of pixels in a contour $\mathcal{S}_k \in \mathcal{S}$, $k = \{1, 2, \ldots, K\}$.

Multiscale skeletons

- Let n_k be the number of pixels in a contour $\mathcal{S}_k \in \mathcal{S}$, $k = \{1, 2, \ldots, K\}$.
- A multiscale skeleton $\hat{S} = (D_I, S)$ is an image where $S(p)$ is the maximum geodesic length $\gamma(f_1, f_2)$ between $f_1, f_2 \in \mathcal{S}_k$ — feature (root) points equidistant to $p \in D_I \backslash \mathcal{S}$ [2].



Multiscale skeletons

- By labeling contour pixels $p_i \in \mathcal{S}_k$, $i \in [1, n]$, with the geodesic length $L_{px}(p_i)$ equal to $f_{geo}(\pi^*_{p_1 \rightsquigarrow p_i})$ (previous lecture), one can directly obtain $\gamma(f_1, f_2)$ between any pair of feature points $(f_1, f_2) \in \mathcal{S}_k$.

$$
\begin{aligned}
\gamma(f1, f2) &= \min\{\Delta, n_k - \Delta\}, \\
\Delta &= |L_{px}(f_2) - L_{px}(f_1)|.
\end{aligned}
$$

Multiscale skeletons

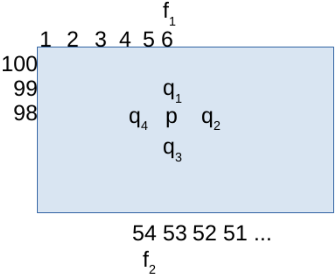
- By labeling contour pixels $p_i \in \mathcal{S}_k$, $i \in [1, n]$, with the geodesic length $L_{px}(p_i)$ equal to $f_{geo}(\pi^*_{p_1 \rightsquigarrow p_i})$ (previous lecture), one can directly obtain $\gamma(f_1, f_2)$ between any pair of feature points $(f_1, f_2) \in \mathcal{S}_k$.

$$\begin{aligned} \gamma(f1, f2) &= \min\{\Delta, n_k - \Delta\}, \\ \Delta &= |L_{px}(f_2) - L_{px}(f_1)|. \end{aligned}$$

- As shown next, the module $|L_{px}(f_2) - L_{px}(f_1)|$ is not even required for multiscale skeletonization.

Multiscale skeletons

You may propagate $L_{px}(R(p))$ in the EDT algorithm to every pixel p, such that $L_{px}(R(p)) = L_{px}(p)$.



Delta = | 54 – 6 | = 48
nk = 100
gamma(f_1,f_2) = min(48,100-48) = 48
S(p) = 48 (importance of p)

$L_{px}(p) = 6$, $L_{px}(q_1) = 6$, $L_{px}(q_2) = 7$

$L_{px}(q_3) = 53$, $L_{px}(q_4) = 54$

$\text{Delta}_i = L_{px}(q_i) - L_{px}(p)$, i = 1,2,3,4

$S(p) = \text{Max}\{ \min_{i=1,2,3,4} \{ \text{Delta}_i , 100-\text{Delta}_i \}\} = 48$

Multiscale skeletons

1. For each $p \in D_I \backslash \mathcal{S}$, do.
2. Set $S(p) \leftarrow 0$.
3. For each $q \in \mathcal{A}_1(p)$ do.
4. If $L_{ct}(R(q)) > L_{ct}(R(p))$, then.
5. Set $S(p) \leftarrow +\infty$.
6. Else if $L_{ct}(R(q)) = L_{ct}(R(p))$, then.
7. Set $\Delta \leftarrow L_{px}(R(q)) - L_{px}(R(p))$.
8. Set $S(p) \leftarrow \max\{S(p), \min_{k=L_{ct}(R(p))}\{\Delta, n_k - \Delta\}\}$.

- Let $R(p)$ be the feature point of p after EDT computation.

Multiscale skeletons

1. For each $p \in D_I \backslash \mathcal{S}$, do.
2. Set $S(p) \leftarrow 0$.
3. For each $q \in \mathcal{A}_1(p)$ do.
4. If $L_{ct}(R(q)) > L_{ct}(R(p))$, then.
5. Set $S(p) \leftarrow +\infty$.
6. Else if $L_{ct}(R(q)) = L_{ct}(R(p))$, then.
7. Set $\Delta \leftarrow L_{px}(R(q)) - L_{px}(R(p))$.
8. Set $S(p) \leftarrow \max\{S(p), \min_{k=L_{ct}(R(p))}\{\Delta, n_k - \Delta\}\}$.

- Let $R(p)$ be the feature point of p after EDT computation.
- Lines 4 and 5 define the SKIZ among multiple contours.

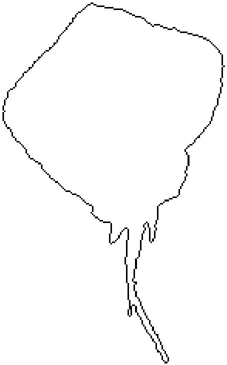
Multiscale skeletons

1. For each $p \in D_I \backslash \mathcal{S}$, do.
2. Set $S(p) \leftarrow 0$.
3. For each $q \in \mathcal{A}_1(p)$ do.
4. If $L_{ct}(R(q)) > L_{ct}(R(p))$, then.
5. Set $S(p) \leftarrow +\infty$.
6. Else if $L_{ct}(R(q)) = L_{ct}(R(p))$, then.
7. Set $\Delta \leftarrow L_{px}(R(q)) - L_{px}(R(p))$.
8. Set $S(p) \leftarrow \max\{S(p), \min_{k=L_{ct}(R(p))}\{\Delta, n_k - \Delta\}\}$.

- Let $R(p)$ be the feature point of p after EDT computation.
- Lines 4 and 5 define the SKIZ among multiple contours.
- Lines 7 and 8 define the importance $S(p)$ based on the geodesic length between feature points of p and its 4-adjacents q.

Multiscale shape representation: Skeletons

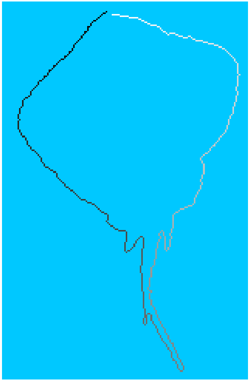
By thresholding, $S(p) \geq T \; \forall p \in D_I$, at a given scale value $T > 0$, one obtains an one-pixel-wide and connected skeleton. Higher is T, more simplified are the skeletons.



Note that connected skeletons are guaranteed only for $\mathcal{A}_{\sqrt{2}}$ in the EDT algorithm.

Multiscale shape representation: Skeletons

By thresholding, $S(p) \geq T \ \forall p \in D_I$, at a given scale value $T > 0$, one obtains an one-pixel-wide and connected skeleton. Higher is T, more simplified are the skeletons.

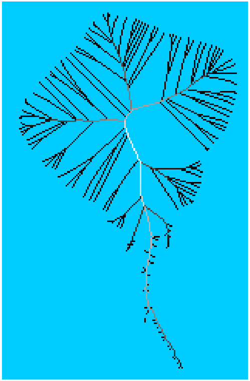


Note that connected skeletons are guaranteed only for $\mathcal{A}_{\sqrt{2}}$ in the EDT algorithm.

By thresholding, $S(p) \geq T \ \forall p \in D_I$, at a given scale value $T > 0$, one obtains an one-pixel-wide and connected skeleton. Higher is T, more simplified are the skeletons.

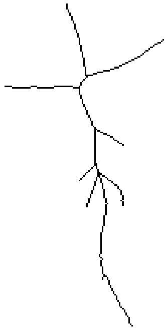
Note that connected skeletons are guaranteed only for $\mathcal{A}_{\sqrt{2}}$ in the EDT algorithm.

By thresholding, $S(p) \geq T \ \forall p \in D_I$, at a given scale value $T > 0$, one obtains an one-pixel-wide and connected skeleton. Higher is T, more simplified are the skeletons.



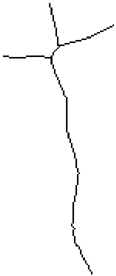
Note that connected skeletons are guaranteed only for $\mathcal{A}_{\sqrt{2}}$ in the EDT algorithm.

By thresholding, $S(p) \geq T \ \forall p \in D_I$, at a given scale value $T > 0$, one obtains an one-pixel-wide and connected skeleton. Higher is T, more simplified are the skeletons.



Note that connected skeletons are guaranteed only for $\mathcal{A}_{\sqrt{2}}$ in the EDT algorithm.

By thresholding, $S(p) \geq T \; \forall p \in D_I$, at a given scale value $T > 0$, one obtains an one-pixel-wide and connected skeleton. Higher is T, more simplified are the skeletons.



Note that connected skeletons are guaranteed only for $\mathcal{A}_{\sqrt{2}}$ in the EDT algorithm.

By thresholding, $S(p) \geq T \; \forall p \in D_I$, at a given scale value $T > 0$, one obtains an one-pixel-wide and connected skeleton. Higher is T, more simplified are the skeletons.

Note that connected skeletons are guaranteed only for $\mathcal{A}_{\sqrt{2}}$ in the EDT algorithm.

An example with multiple contours.

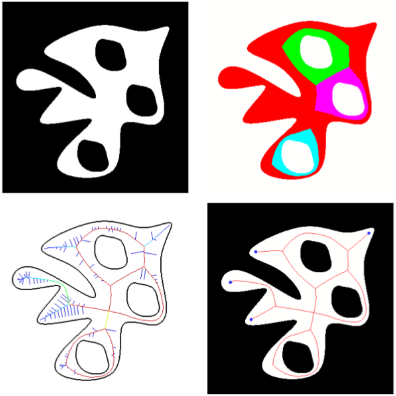


Figure from [2].

3D surface skeletons can be obtained by the direct extension of the geodesic length on surfaces as importance measure [2].

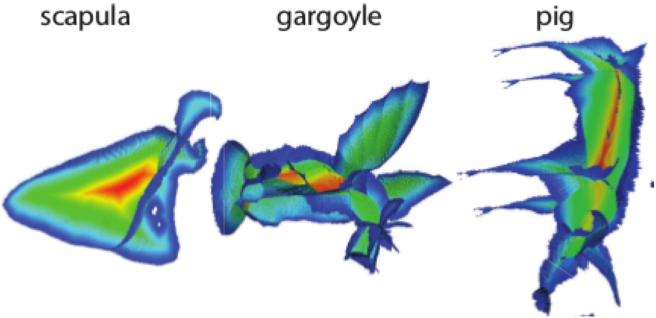


Figure from [2].

3D surface skeletons can be obtained by the direct extension of the geodesic length on surfaces as importance measure [2].

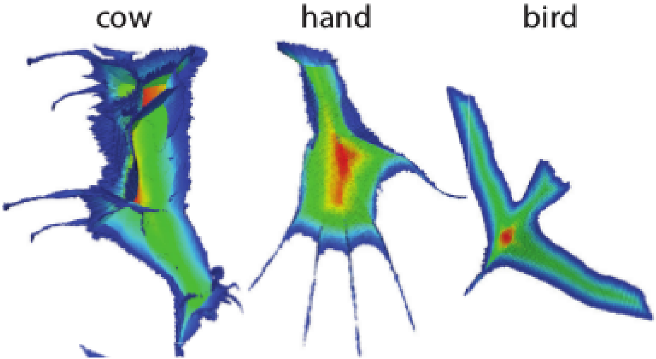
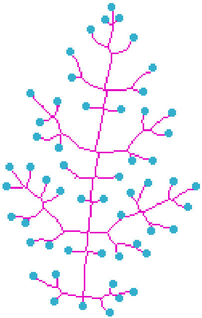


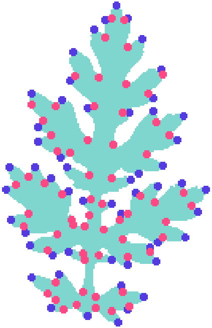
Figure from [2].

Salience points



Terminal points of the internal and external skeletons can be directly related to convex and concave salience points on the contour, respectively [3].

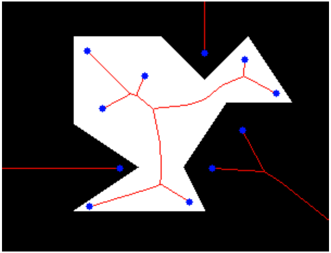
Salience points



Terminal points of the internal and external skeletons can be directly related to convex and concave salience points on the contour, respectively [3].

Salience points

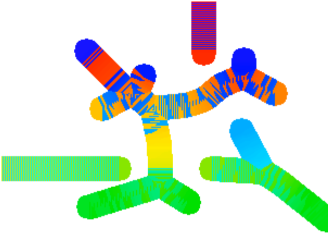
Let $\hat{Sk} = (D_I, Sk)$ be a binary image, where $Sk(p) = 1$ when $p \in D_I$ is a skeleton point and $Sk(p) = 0$ otherwise. Terminal points of the skeleton can be defined as pixels $p \in D_I$ with exactly one $q \in \mathcal{A}_{\sqrt{2}}(p) \backslash \{p\}$ such that $Sk(p) = Sk(q) = 1$.



Such a definition might fail if exists $q_1, q_2 \in \mathcal{A}_{\sqrt{2}}(p) \backslash \{p\}$, such that $Sk(p) = Sk(q_1) = Sk(q_2) = 1$, $q_1 \neq q_2$, and $(q_1, q_2) \in \mathcal{A}_1$.

Salience points

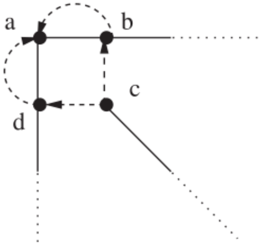
An alternative is to slightly change the scale threshold used to obtain the skeleton. Another alternative is to compute the influence zones of the skeleton points within a small distance r (e.g., 10) to it.



The area A and aperture angle θ of the influence zones are higher for terminal points. By measuring $A = \frac{\theta r^2}{2}$, it is safe to select points with θ above a threshold (e.g., $\theta > 70^o$).

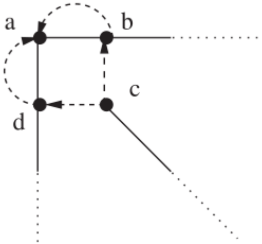
Salience points

Now, to determine which salience point a on a contour of perimeter n corresponds to a terminal point c of the skeleton, we must determine if the root $R(c) = b$ or $R(c) = d$.



Salience points

Now, to determine which salience point a on a contour of perimeter n corresponds to a terminal point c of the skeleton, we must determine if the root $R(c) = b$ or $R(c) = d$.



Let $q \in \mathcal{A}_1(c)$ be the pixel used to set $S(c)$ in the multiscale skeletonization algorithm. Then, either

(1) $R(q) = d$ and $R(c) = b$, or

(2) $R(q) = b$ and $R(c) = d$.

Salience points

- For clockwise-labeled contour pixels,
  (1) If $R(c) = b$, then a is the point with $L_{px}(a)$ obtained from $L_{px}(b) - \frac{S(c)}{2}$.
  (2) If $R(c) = d$, then a is the point with $L_{px}(a)$ obtained from $L_{px}(d) + \frac{S(c)}{2}$.

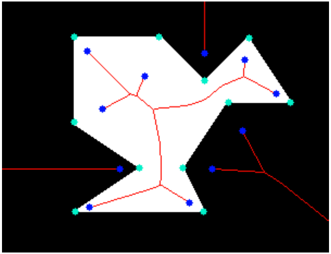
Salience points

- For clockwise-labeled contour pixels,
  (1) If $R(c) = b$, then a is the point with $L_{px}(a)$ obtained from $L_{px}(b) - \frac{S(c)}{2}$.
  (2) If $R(c) = d$, then a is the point with $L_{px}(a)$ obtained from $L_{px}(d) + \frac{S(c)}{2}$.
- For $\Delta = L_{px}(q) - L_{px}(c)$, case (1) occurs when $\Delta \geq n - \Delta$, and case (2) occurs otherwise.

Salience points

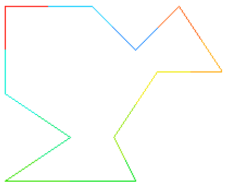
- For clockwise-labeled contour pixels,
  (1) If $R(c) = b$, then a is the point with $L_{px}(a)$ obtained from $L_{px}(b) - \frac{S(c)}{2}$.
  (2) If $R(c) = d$, then a is the point with $L_{px}(a)$ obtained from $L_{px}(d) + \frac{S(c)}{2}$.
- For $\Delta = L_{px}(q) - L_{px}(c)$, case (1) occurs when $\Delta \geq n - \Delta$, and case (2) occurs otherwise.
- We use a sign $s \in \{-1, 1\}$ to indicate (1) and (2), and find a by computing $\delta \leftarrow L_{px}(R(c)) + s\frac{S(c)}{2}$ as the point with
  - $L_{px}(a) = \delta$, when $\delta \in [1, n]$,
  - $L_{px}(a) = \delta - n$, when $\delta > n$, and
  - $L_{px}(a) = \delta + n$, when $\delta \leq 0$.

Salience points



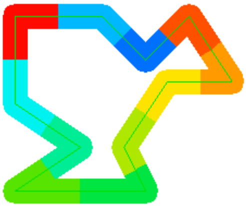
Once salience points on the contour are determined, their influence zones on the contour define segments, which have influence zones using a small distance (e.g., 10) higher outside than inside the shape when they are convex, and the other way around when they are concave.

Salience points



Once salience points on the contour are determined, their influence zones on the contour define segments, which have influence zones using a small distance (e.g., 10) higher outside than inside the shape when they are convex, and the other way around when they are concave.

Salience points



Once salience points on the contour are determined, their influence zones on the contour define segments, which have influence zones using a small distance (e.g., 10) higher outside than inside the shape when they are convex, and the other way around when they are concave.

[1] A.X. Falcão, L.F. Costa, and B.S. Cunha.
Multiscale skeletons by image foresting transform and its application to neuromorphometry.
Pattern Recognition, 35(7):1571 – 1582, 2002.

[2] A.X. Falcão, C. Feng, J. Kustra, and A.C. Telea.
Chapter 2 - multiscale 2d medial axes and 3d surface skeletons by the image foresting transform.
In P.K. Saha, G. Borgefors, and G.S. di Baja, editors, *Skeletonization*, pages 43 – 70. Academic Press, 2017.

[3] R. da S. Torres and A.X. Falcão.
Contour salience descriptors for effective image retrieval and analysis.
Image and Vision Computing, 25(1):3 – 13, 2007.