2D Min-Filters with Polygons

Paolo Codenotti and Pedro Felzenszwalb
University of Chicago
{paoloc, pff}@cs.uchicago.edu

1. Introduction

Min-filters are an important operation in image processing. The min-filter of an image I by a shape S associates to every integer translate of S the minimum value that I takes inside the translated shape. Min-filters are commonly used for noise suppression, contrast enhancement, and edge detection [1]. Min-filters are also an important special case of min-convolutions, which have been used in signal processing, pattern recognition, computer vision, and combinatorial optimization [2].

Most applications of min-filters use small filtering elements. This is partly due to the computational burden of using large shapes. Gil and Werman [3] described an algorithm to compute the min-filter of an image by an axis parallel rectangle in constant time per pixel. Here we describe efficient algorithms for computing min-filters by large polygons.

Min-filters are related to convolutions by binary shapes. It was previously known how to compute convolutions by a convex polygon with k sides in $O(k)$ time per pixel [4]. However, computing the min-filter is significantly different from computing convolutions, since the $\min$ operation does not have an inverse like summation does. On the other hand, we can use the fact that $\min$ is an idempotent operator.

Our algorithms are based on the idea of covering a shape with smaller copies of itself for a dynamic programming approach. In general we obtain approximate results that have a simple interpretation: for each translate of a shape S we compute the minimum value that the image I takes inside a shape S' that is close to S. This notion of approximation is similar to the one used in [4] for convolutions.

Let $\overline{\mathbb{R}} = \mathbb{R} \cup \{\infty\}$, and $[n] = \{0, 1, \ldots, n-1\}$. For a shape $S \subseteq \mathbb{R}^2$, let $\mathrm{Int}(S)$ denote the integer points in S. Polygons P will be specified so that their center of mass is at the origin. We use $P[i]$ to denote the translate of P by $i \in \mathbb{R}^2$. That is, $P[i] = P \oplus i = \{j + i \mid j \in P\}$.

Definition 1 *Let P be a polygon, $D = [n] \times [m]$, and f be a function $D \to \overline{\mathbb{R}}$. The min-filter of f by P, $h = f \otimes P$, is another function $D \to \overline{\mathbb{R}}$:*

$$h(i) = \min_{j \in \mathrm{Int}(P[i])} f(j).$$

Here we assume that the value of f outside of D is ∞.

In image processing applications $f(i)$ is the brightness of the pixel centered at i in an n by m image.

An α-approximation of a shape S is a shape $\widetilde{S}$ such that every point that is well inside S (α away from the boundary), is in $\widetilde{S}$, and every point well outside S is not in $\widetilde{S}$.

Definition 2 *Given a shape S, an* α-approximation *of S is a shape $\widetilde{S}$ such that:*

- *$x \in S$, and $d(x, \partial S) \geq \alpha \Rightarrow x \in \widetilde{S}$,*
- *$x \notin S$, and $d(x, \partial S) \geq \alpha \Rightarrow x \notin \widetilde{S}$.*

Where $d(x, y)$ denotes the distance between two objects.

Note that if $\alpha \leq 1/2$ then every pixel that is completely inside S is inside $\widetilde{S}$, and every pixel that is completely outside S is outside of S'.

Let $\mathrm{Int}_\alpha(S)$ be the set of integer points inside an α-approximation of S. An *α-approximate* min-filter of a function f by a polygon P is defined by:

$$(f \otimes_\alpha P)(i) = \min_{j \in \mathrm{Int}_\alpha(P[i])} f(j).$$

Here each entry $(f \otimes_\alpha P)(i)$ is the min over integer points in some α-approximation to $P[i]$. In particular, $\mathrm{Int}_\alpha(P[i])$ may not be a translate of $\mathrm{Int}_\alpha(P[j])$ for $i \neq j$.

2. Results

Our main results are efficient algorithms for computing approximate min-filters of a function $f : D \to \overline{\mathbb{R}}$ by large polygons. All of the running times are linear in the size of D, which corresponds to the number of pixels in an image.

Theorem 1 *For a (non axis-parallel) rectangle R, we can compute $f \otimes_\alpha R$ in $O((\frac{1}{\alpha})^2 \log L)$ time per pixel, where L is the length of the longest side of R.*

Theorem 2 *For a triangle T, we can compute $f \otimes_\alpha T$ in $O(\frac{1}{\sin\gamma}(\frac{1}{\alpha})^2 \log L)$ time per pixel, where L is the length of the longest side and γ is the smallest angle in T.*

Theorem 3 *For a polygon P, we can compute $f \otimes_\alpha P$ in $O((\frac{1}{\alpha})^2 k \log^3 L)$ time per pixel, where L is the length of the longest side and k is the number of sides of P.*

3. The Main Algorithm

Here we describe the algorithm to compute an approximate min-filter by convex polygons with a constant number of sides and all angles at least $\pi/2$. The results can be extended to arbitrary polygons. We can handle small angles by

"cutting corners" and arbitrary polygons (non-convex with many sides) using triangulations.

Our algorithm uses a simple geometric fact: any convex polygon P with k sides can be covered by k smaller polygons $\widetilde{P}_1, \ldots, \widetilde{P}_k$ that are translates of each other. These polygons are similar to P but scaled down by a factor of $2/3$. Each of the polygons $\widetilde{P}_i$ covers a particular corner of P. The idea of our algorithm is to recursively compute the min-filter by P in terms of the min-filter by $\widetilde{P}_i$. However, the polygons $\widetilde{P}_1, \ldots, \widetilde{P}_k$ are generally not centered at integer points. Since each covering polygon will be near but not exactly at an integer point, we use expanded versions of the polygons at each level to ensure proper coverage. This approach introduces a "drift" which can be controlled by computing every min-filter over a finer grid G_c, where c is the distance between neighboring points in the grid.

In order to make sure we make good progress at each level of recursion we need to carefully choose how to expand $\widetilde{P}_i$. In particular neither a global scaling nor the Minkowski sum of $\widetilde{P}_i$ with a small shape have the right properties.

Definition 3 (α-expansion) *The* α-expansion *of a convex polygon P is the (convex) polygon $\widehat{P}$ constructed as follows. For each side s of P, look at the line l_s parallel to s at distance α from s and outside the polygon P. The α-expansion $\widehat{P}$ is the "interior" of these lines (see Figure 1(a)).*

Lemma 1 *Let $\widehat{P}$ be the α-expansion of a convex polygon P. Let ℓ be the length of any side of P, and ℓ' the length of the corresponding side of $\widehat{P}$. Let γ and θ be the angles incident to ℓ (see Figure 1(a)). Then $\ell' = \ell + \alpha t$, where*

$$t = \frac{1+\cos\gamma}{\sin\gamma} + \frac{1+\cos\theta}{\sin\theta}.$$

In particular, if both angles are at least $\pi/2$ then $\ell' \leq \ell + 2\alpha$. Moreover if all angles in P are at least $\pi/2$, then $\widehat{P}$ is a $\sqrt{2}\alpha$-approximation to P.

The α-expansion of a polygon P is another polygon with the same number of sides and the same angles, containing all points within α of P. Moreover, the sides don't grow by much, as described by Lemma 1.

Consider a convex polygon P with covering polygons $\widetilde{P}_1, \ldots, \widetilde{P}_k$ as described above. Let $\widehat{P}_i$ be the c-expansion of $\widetilde{P}_i$. Let P_i be the translate of $\widehat{P}_i$ such that the center of mass of P_i is the closest point to the center of mass of $\widehat{P}_i$ in the grid G_c. Note that P_i is a c-approximation to $\widehat{P}_i$. Moreover, the covering polygons P_i are congruent and centered at grid points. Figure 1(b) illustrates the construction.

Lemma 2 *Let $P' = \bigcup_i P_i$, then $P \subseteq P'$ and P' is a $(1+\sqrt{2})c$-approximation to P.*

Now we give the algorithm for computing $f \otimes_\alpha P$. We construct a sequence of polygons $P^0, P^1, P^2, \ldots P^s$, where

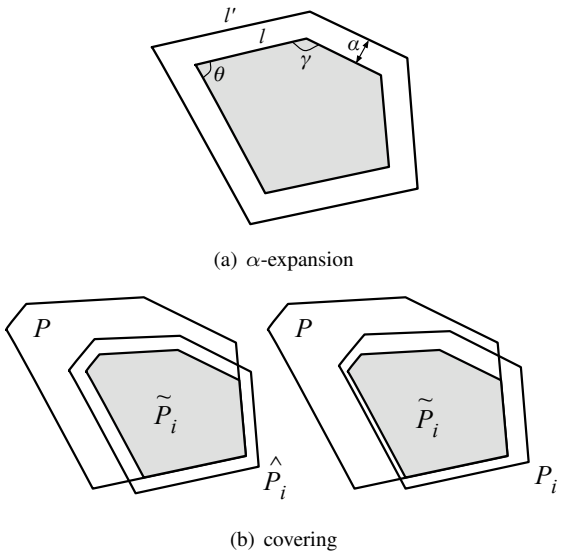

Figure 1: (a) A polygon and its α-expansion. The length of each side grows according to α and the angles adjacent to that side. (b) A polygon P, one of its covering polygons $\widetilde{P}_i$, the α-expansion $\widehat{P}_i$ and the translation with center of mass at an integer point P_i.

$P^0 = P$ and P^{q+1} has the shape of the covering polygons P_i^q for P^q. Let y_i^q be center of P_i^q. Let g^s be the (exact) min-filter of f by P^s. We will compute g^s by exhaustive search. We then compute an approximate min-filter of f by P^q starting with $q = s-1$ down to $q = 0$ using the recursion:

$$g^q(x) = \min_i \{g^{q+1}(x + y_i^q)\}. \tag{1}$$

The running time to compute the base case is $O((kl)^2)$ per point in G_c, where l is the maximum length of a side of P^s. The running time for every other level is $O(k)$ per point in G_c. If we look at P^q as q increases we see that long sides get smaller by a factor close to 2/3, and short sides remain relatively short. By Lemma 1 we have $l \leq L(2/3)^s + 6c$, where L is the longest side of P. Now we choose $s = \log L$, and $c = \alpha/((1+\sqrt{2})\log L)$. With this choice of parameters, the analysis above proves the running time claimed in Theorem 3, and Lemma 2 can be used to show that the the algorithm correctly computes an α-approximate min-filter.

References

[1] P. Maragos. *Morphological Filtering for Image Enhancement and Feature Detection*. Elsevier Academic Press, 2005.

[2] L. Babai and P. Felzenszwalb. Computing rank convolutions with a mask. Technical report, University of Chicago, 2006.

[3] J. Gil and M. Werman. Computing 2-d min, median, and max filters. *IEEE Transactions on pattern analysis and machine intellingence*, 15:504–507, 1993.

[4] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. Piatko, R. Silverman, and A. Y. Wu. Approximating large convolutions in digital images. In *Proc. SPIE Vol. 3454, Vision Geometry VII*, pages 216–227, October 1998.