AMAT: Medial Axis Transform for Natural Images
Supplemental Material

1. Introduction

In this supplemental material we add details and qualitative results that we were not able to include in the main submission due to the limited space. We analytically derive the computational complexity of the various steps of our method, and discuss ways of reducing it. We also present additional qualitative results of medial axes computed with our method, and their respective reconstructions. Finally, in the zip file for the supplemental material we include a video showing the execution of the greedy algorithm on a test image (in normal and fast speed). In the video we visualize the process of selecting disks and covering the image, as well as the medial axis construction and respective radii.

2. Complexity Analysis

First, we remind the reviewer of the notation for some useful quantities to make the supplemental material self-contained. A disk of radius $r_j \in \mathcal{R} = \{1, \ldots, R\}$, centered at point $\mathbf{p}_i \in X^I \subset \mathbb{R}^2$, is denoted as $D_{\mathbf{p}_i, r_j} = D_{ij}$. We also assume that X^I is a universe of N points in total, corresponding to the spatial support of an input RGB image I.

Encodings: Computing the encodings $\mathbf{f}_{ij}$ at all possible locations and scales involves convolving the input image with disk filters of radii $1, \ldots, R$. The complexity of this step is:

$$O(N \cdot 1^2 + N \cdot 2^2 + \ldots + N \cdot R^2) = O(N \sum_{r=1}^{R} r^2) = O(NR(R+1)(2R+1)/6) = O(NR^3). \tag{1}$$

Disk cost: The most demanding step in our method is the computation of the disk costs

$$c_{ij} = \sum_k \sum_l \|\mathbf{f}_{ij} - \mathbf{f}_{kl}\|^2 \quad \forall k, l : D_{kl} \subset D_{ij}. \tag{2}$$

This quantity must be computed at all location-radius combinations, and *for all the contained disks* of radii $r \in \mathcal{R}$. Let $r, r' \in \mathcal{R}$ with $r < r'$ and consider a disk of radius r', centered at point $\mathbf{p}'$, $D_{\mathbf{p}',r'}$. A *fully contained* disk $D_{\mathbf{p},r} \subset D_{\mathbf{p}',r'}$ can be placed anywhere inside the disk $(x - x_{\mathbf{p}'})^2 + (y - y_{\mathbf{p}'})^2 < (r' - r)^2$. It follows that the number of such contained disks for a given radius r' is approximately

$$N_d^{r'} = \sum_{r=1}^{r'} \pi(r' - r)^2 = \pi \sum_{r=1}^{r'} (r')^2 - 2r'r + r^2 = k(r')^3, \tag{3}$$

where k is a constant factor. Summing over all possible radii $r' \in \mathcal{R}$ we have that the total number of disks at $\mathbf{p}$ is

$$N_d = \sum_{r'=1}^{R} N_d^{r'} = \sum_{r'=1}^{R} k(r')^3 = \frac{k(R(R+1))^2}{4} \sim R^4. \tag{4}$$

This is the (approximate up to a constant factor) number of contained disks and respective encodings that are used to compute c_{ij} at all possible scales $r \in \mathcal{R}$, at a single point in the image. Repeating this process for all N points in the image domain yields a total complexity of $O(NR^4)$.

Greedy approximation algorithm: At each step of the greedy algorithm we cover at most $O(R^2)$ pixels in X^I, but we also have to update costs for all partially and entirely covered disks, at all scales. The greedy algorithm has complexity $O(R^2 N \sum_{r=1}^{R} r^2) = O(NR^5)$.

2.1. Reducing Complexity

The $O(NR^4)$ complexity derived in the previous section can be reduced through a combination of the following: i) rewriting Equation (2) in an equivalent form that is amenable to efficient computation using convolutions; ii) using cumulative sums to avoid redundant computations. Since we will release our code, we do not elaborate further on i).

We provide an intuitive explanation for ii): Suppose we use $\sum_{D_{ij}} \mathbf{f}$ to denote the sum of a quantity $\mathbf{f}$ within a disk area D_{ij} of radius r_j. Naively calculating this quantity for a disk of radius $r_l > r_j$ involves unnecessary recomputations of $\mathbf{f}$ in the area $D_{ij} \subset D_{il}$. We can avoid

these redundant computations, observing that

$$\sum_{D_{ij}} \mathbf{f} = \sum_{j=1}^{R} \sum_{C_{ij}} \mathbf{f}, \tag{5}$$

where C_{ij} is the *circle* of radius r_j, centered at $\mathbf{p}_i$. Using the relationship in Equation (5) we reduce the complexity of computing feature encodings at all locations from $O(NR^3)$ to $O(NR^2)$ and the complexity of disk cost computation from $O(NR^4)$ to $O(NR^3)$.

Using a shape other than disks in our approach allows us to reduce complexity even more. For example, one can consider using rectangles or squares to approximate or replace disks, as in [1, 4]. In that scenario, pre-computing an integral image would allow us to use a fixed number of operations per image point, further reducing the complexity of encoding and disk cost computation to $O(NR)$ and $O(NR^2)$, respectively. Similarly, the complexity of the greedy algorithm can be reduced to $O(NR^3)$. We aim to explore this in future work.

3. Qualitative results

In the following pages we illustrate more examples of qualitative results. In Figure 1 we are showing results for the task of medial axis extraction. From left to right we are showing a) the input image; b) the extracted medial axes; c) the medial connected components after grouping; and d) the ground-truth medial axes in BMAX500.

In column b) the color of the medial axes represents the encoding of the respective disk-shaped region in the input image, and we use black color to denote non-medial points. Note that dark regions in the original image result in axes with low contrast w.r.t. to non-medial points; please magnify and view in color.

For column c) we are showing the grouped medial connected components after using our agglomerative scheme on the AMAT's output. We use (random) color coding to highlight different groups. Note that using the criterion of similarity in terms of both scale-space proximity and appearance proximity, the resulting groups correspond to meaningful, distinct ares in the image; *e.g.* a single connected component for the sky and another one for the person's skirt in Figure 1, third row. Since there are multiple segmentations available for each image in BSDS500, in column d) we are showing the medial axis ground-truth of the segmentation that results in the best GT-SKEL reconstruction quality.

In Figure 2 we compare reconstruction results obtained by AMAT and the three baselines we described in the paper (MIL, GT-SEG, GT-SKEL). Again, we can see that with AMAT we can recover more detail than with the alternatives. As a sidenote, one can also observe that the output of our algorithm resembles an artistic rendering of the input image, such as the ones illustrated in [2, 3].

References

[1] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. Contour detection and hierarchical image segmentation. *TPAMI*, 2011.

[2] B. Gooch, G. Coombe, and P. Shirley. Artistic vision: painterly rendering using computer vision techniques. In *International Symposium on Non-photorealistic animation and rendering*, 2002.

[3] J. Hays and I. Essa. Image and video based painterly animation. In *Proceedings of the 3rd international symposium on Non-photorealistic animation and rendering*, 2004.

[4] S. Tsogkas and I. Kokkinos. Learning-based symmetry detection in natural images. In *ECCV*, 2012.

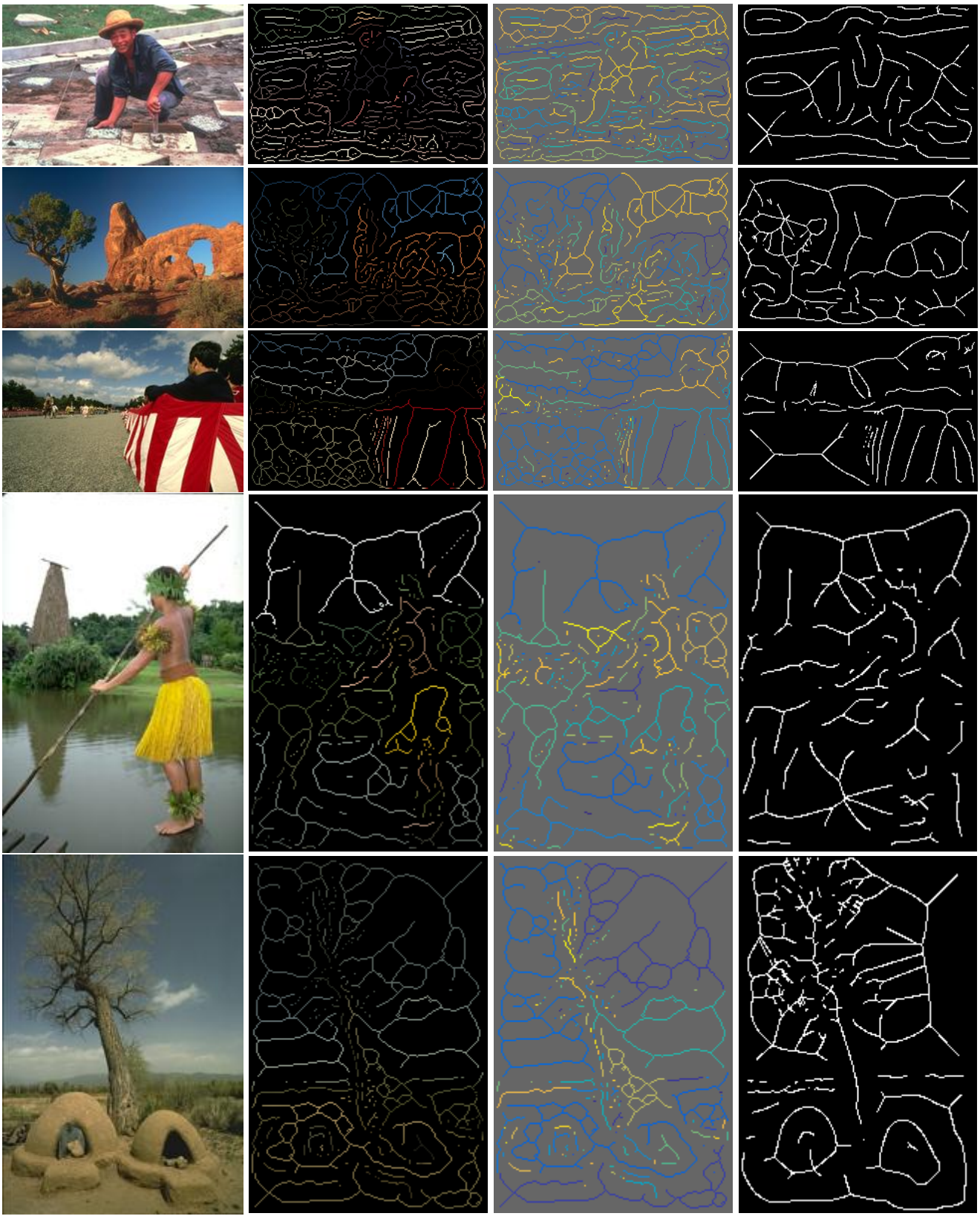

Figure 1: **Medial axes**. From left to right: Input image, AMAT medial axes (axis color indicates the respective encodings $\mathbf{f}$, and black is used for unused points), connected components of medial points (color-coded), ground-truth skeletons.

Figure 2: **Image reconstruction**. From left to right: Input image, MIL [4], GT-seg, GT-skel, AMAT.