The Medial Feature Detector: Stable Regions from Image Boundaries

Yannis Avrithis and Konstantinos Rapantzikos
National Technical University of Athens
{iavr,rap}@image.ntua.gr

Abstract

We present a local feature detector that is able to detect regions of arbitrary scale and shape, without scale space construction. We compute a weighted distance map on image gradient, using our exact linear-time algorithm, a variant of group marching for Euclidean space. We find the weighted medial axis by extending residues, typically used in Voronoi skeletons. We decompose the medial axis into a graph representing image structure in terms of peaks and saddle points. A duality property enables reconstruction of regions using the same marching method. We greedily group regions taking both contrast and shape into account. On the way, we select regions according to our shape fragmentation factor, favoring those well enclosed by boundaries—even incomplete. We achieve state of the art performance in matching and retrieval experiments with reduced memory and computational requirements.

1. Introduction

Most successful region detectors like SIFT [9], SURF [2], or *Hessian-affine* [13], are based on *region intensity* and center-surround operations in scale space, inspired by biological vision. They can estimate both location and scale but only a crude approximation of local shape. MSER [11] is also based on region intensity and can adapt to arbitrary shape. *Boundary*-based methods, may be a popular way towards regions under perceptual criteria [1] or shape-based object detection [4], but have not been used for repeatable features, with few exceptions like EBR [19].

On the other hand, the *distance transform* has been a very successful boundary operator, used in different contexts like shape filtering with watershed segmentation [21], or object detection [6]. But again, it has not been used for repeatable features, except [17]. The *medial axis transform* is even more unexploited in this direction. It is still considered unstable and typically used for shape representation and matching on single objects, *e.g.* [3].

In this work, we extend our rationale of [17] in several directions. Fig. 1 gives a preview of our result. No edges are needed—the *distance map* is computed directly on image gradient. It is *weighted* by the gray level of the gradient, but propagation obeys the Euclidean distance, evident *e.g.* at the top-right of Fig. 1(b). A *weighted medial axis* is computed on the entire image representing its structure as a *graph*. With a dual operation, labels are *backpropagated* to give a *partition* as shown in green lines, guided by a *decomposition* of the medial axis at saddle points of the distance map. The latter are exactly the points where the medial axis meets partition boundaries in Fig. 1(c).

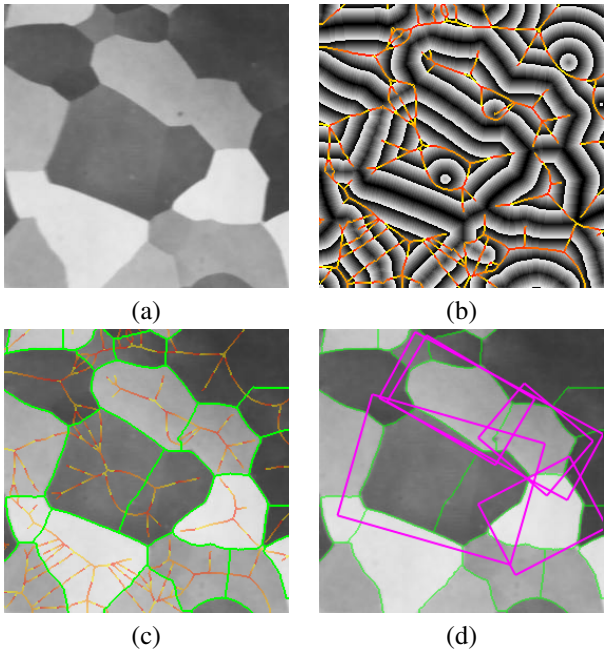


Figure 1. Medial feature detection. (a) Input image. (b) Weighted distance map and medial axis. The color map indicates ascending direction by (`black` → `white`) and (`yellow` → `red`) transitions, respectively. (c) Image partition. (d) Regions detected.

Graph vertices are constructed at distance peaks in region interior, and edges at saddle points on region boundaries. The partition is sensitive both to *contrast*, represented by gradient strength, and to *shape*—regions are separated even where intensity is uniform. Both elements are captured in edge weights, guiding a region grouping process

whereby candidate features are generated. The features, as shown in Fig. 1(d), are selected according to our *shape fragmentation factor*. The latter measures how well a region is enclosed by boundaries, in agreement with the Gestalt principle of closure. Incomplete boundaries are allowed, as well as independent selection of whole regions and their parts.

2. Related work and contribution

Weighted distance transform has been studied primarily as a solution to the *Eikonal equation*, in problems like shading from shape [20]. A given function specifies the *refractive index* on the plane, while a set of *source points* specifies boundary conditions. We do *not* require a given source map. Our weighting mechanism is more similar to [5] where the distance map is obtained by an *infimal convolution* operation, equivalent to *weighted erosion* [10].

Our implementation is a variant of *group marching* (GMM) [7], a linear-time fast marching method that selects a number of points on the propagating front to move as a group, thus avoiding the cost of sorting. We move *all* points of the front as a group using a constant-time *priority queue* on *quantized* distance. This is more similar to [22], and in the binary case it would reduce to the *two-queue* scheme of [10]. However, due to the Euclidean assumption and a bidirectional update, the entire computation is *exact*.

To our knowledge, this is the first work to study the medial axis on a distance map weighted by infimal convolution. Rather than working on PDE's like [18], we use a *residue* criterion based on proximity of source points along boundaries. It is naturally connected to the definition of the medial axis and guarantees connectedness. It has been used for *Voronoi skeletons* [15] on binary shapes. We extend it to arbitrary functions in the plane and show that it can be computed with a similar *constant-time* operation.

Medial decomposition methods are most often found in problems of computational geometry like domain decomposition [8] in binary images. We allow decomposition in *gray-level* images, similarly to the *upper/lower complete* function of [12]. Our approach is closest to *watershed* segmentation applied to the distance map of *binary* regions [21], but we use the *weighted distance map* of the gray-level input instead. Our *partitioning* is fundamentally different from gray-level watershed, in that the latter is guided by image gradient. To our knowledge, no work has studied the medial axis for image structure representation, region grouping and repeatable feature detection.

3. Weighted distance map

Representation. We represent 2D images by functions $f : \mathbb{X} \to \mathbb{V}$. As *range* $\mathbb{V}$ we use the extended real line $\overline{\mathbb{R}} = \mathbb{R} \cup \{-\infty, \infty\}$ and as *domain* $\mathbb{X}$ the continuous (discrete) space $\mathbb{R}^2$ ($\mathbb{Z}^2$). We denote by $\mathbb{F}$ the space of all such functions. In practice, we work on a bounded subset $X \subseteq \mathbb{X}$. In the *discrete* domain, we identify X with the set of vertices V of a *grid* (graph) $G = \{V, E\}$ and define its edges $E \subseteq V \times V$ as the set of vertex pairs $e = (u, v)$ such that $u, v \in V$ are *connected*. We use 4- or 8-connectivity, and write $u \diamond v$ ($u * v$) iff u, v are 4- (8-) connected.

Definitions and properties. Given a metric d in $\mathbb{X}$, we define the *weighted distance transform* or *distance function* or *distance map* $\mathcal{D}_d(f)$ of image f w.r.t. d as

$$\mathcal{D}_d(f)(x) = \bigwedge_{y \in X} d(x, y) + f(y), \quad x \in X, \tag{1}$$

Most often we use a metric induced by a norm $\|\cdot\|$, that is, $d(x, y) = \|x - y\|$ for $x, y \in \mathbb{X}$. We then omit d and write $\mathcal{D}(f)$ instead. Although (1) applies to arbitrary functions f, if the problem at hand is region detection in images, we use a function that is related to *boundaries*, like *gradient*. This is discussed in section 5. It is known that (1) is equivalent to *infimal convolution* [10].

We define for each point $x \in X$ its *minimal set* $S^*_f(x)$ w.r.t. f as the (possibly empty) set of points $y \in X$ for which quantity $d(x, y) + f(y)$ is minimized:

$$S^*_f(x) = \{y \in X \,:\, d(x, y) + f(y) = \mathcal{D}(f)(x)\} \tag{2}$$

for $x \in X$. We often omit f and write $S^*(x)$ instead. We also write $y \succcurlyeq x$ iff $y \in S^*(x)$. We further define the *source set* $S_f(x)$ of x as the subset of its minimal set such that no two points $y, z \in S_f(x)$ are related by $y \succcurlyeq z$:

$$S_f(x) = \{y \in S^*_f(x) \,:\, \nexists z (y \succcurlyeq z \succcurlyeq x)\}. \tag{3}$$

We also omit f and write $S(x)$. We say that y is a *source of* x and write $y \succ x$ iff $y \in S(x)$. More generally, we say that $y \in X$ is a *source* iff $y \succ x$ for some $x \in X$, even itself. We assume in this work that each $x \in X$ has at least one source: $y \succ x$ for some $y \in X$. This is always true in the discrete domain.

Lemma 3.1 *Given $y \in X$, the following are equivalent: (a) y is a source, (b) $\mathcal{D}(f)(y) = f(y)$, (c) $S(y) = \{y\}$, (d) $y \succ y$.*

Define the *source set* $S(f)$ of f as the set of all sources $y \in X$. It follows that $S(f) = \{x \in X \,:\, x \succ x\}$. This makes it easy to detect sources. By $s(x), x \in X$ we denote the source of x if it is unique, otherwise any representative of $S(x)$. We call function $s : X \to X$ a *source map*.

Lemma 3.2 *The distance map $\mathcal{D}(f)$ is uniquely determined by the restriction $f|_{S(f)}$ of f on its source set.*

This is a generalization of an analogous observation on the binary distance map, which, for a binary input $B \subseteq \mathbb{X}$,

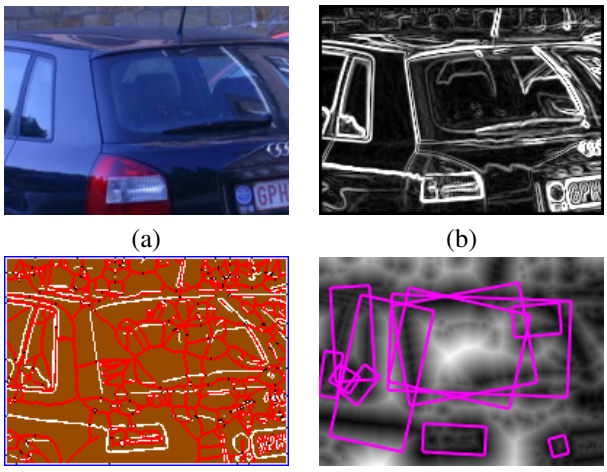


Figure 2. (a) Input image, detail of `leuven` scene. (b) Gradient map. (c) `White`: source; `red`: medial axis; `black`: saddle point; `blue`: image boundary. (d) Weighted distance map, with detected features in overlay.

is uniquely determined by its boundary ∂B. Source sets are then closely related to region boundaries. Accordingly, we define the *interior set* of f as $I(f) = X \setminus S(f)$.

Computation. Given an image f, we develop our *exact group marching* (EGM) algorithm to compute the distance map $h = \mathcal{D}(f)$ according to (1) and the source map s in the discrete domain, using the Euclidean metric. EGM is outlined in algorithm 1. We initialize propagation at the *source seed set* $S_+(f)$, defined as

$$S_+(f) = \{x \in X \,:\, f(x) < \min_{y \diamond x} f(y) + 1\}. \qquad (4)$$

Because $d(x, y) = 1$ for $y \diamond x$, it can be shown that $S_+(f)$ is a superset of the source set $S(f)$.

At the heart of propagation lies a *priority queue* with discrete priority levels, implemented as an array of internal FIFO queues. Points are labeled as `far`, `near`, or `done`. The queue holds points that are `near`, that is, points on the propagation front. Points are processed in *groups*: each point x is processed according to its level $\lfloor h(x) \rfloor$ and points with the same level at random order. Neighbors y that are `far` PROPagate the front; `near` ones participate in an UPDATE process *twice*, first in an inward (line 7) and then in an outward (line 8) direction w.r.t. x. The computation is *exact*, despite the random processing order:

Proposition 3.3 *(a) EGM correctly computes distance $\mathcal{D}(f)(x)$ as defined in (1) and source point $s(x)$ for each $x \in X$. (b) Its time complexity is $O(n)$, where $n = |X|$.*

Fig. 2 shows an example of weighted distance along with source points, medial axis and detected features. The image gradient is used as an input, as detailed in sections 5, 6.

Algorithm 1 Exact Group Marching

```
1:  procedure EGM(image f)
2:      initialize q, h, s; construct seed S+ as in (4)
3:      for x ∈ S+ do s(x) ← x; PROP(x, x)
4:      for x ∈ X \ S+ do label x as far
5:      while ¬ q.EMPTY( ) do
6:          x ← q.POP( ); label x as done
7:          for y◇x, y near do UPDATE(y, x)        ▷ inward
8:          for y◇x, y near do UPDATE(x, y)        ▷ outward
9:          for y◇x, y far do PROP(x, y)
10:     end while
11:     return distance map h, source map s
12: end procedure
13:
14: procedure PROP(point x, point y)
15:     z ← s(x); h(y) ← d(y, z) + f(z); s(y) ← z
16:     q.PUSH(y, ⌊h(y)⌋); label y as near
17: end procedure
18:
19: procedure UPDATE(point x, point y)
20:     z ← s(x); h* ← d(y, z) + f(z); if h* ≥ h(y) return
21:     h(y) ← h*; s(y) ← z
22: end procedure
```

4. Weighted medial axis

Definitions and properties. Given the previous definitions of sources in a weighted distance map, we say that $x \in X$ is a *medial point* of f if it has at least two distinct sources. The *weighted medial axis* or simply *medial axis* $A(f)$ is the set of all such points: $A(f) = \{x \in X \,:\, |S_f(x)| > 1\}$.

Lemma 4.1 *The source set and the medial axis of an image f are mutually exclusive: $S(f) \cap A(f) = \emptyset$. Hence the medial axis is contained in the interior set, $A(f) \subseteq I(f)$.*

The *medial axis function* $\mathcal{A}(f)$ is defined as the restriction of the distance map $\mathcal{D}(f)$ on the medial axis: $\mathcal{A}(f) = \mathcal{D}(f)|_{A(f)}$. It is a subset of the (3D) *product space* $\mathbb{E} = \mathbb{X} \times \mathbb{V}$. The definitions above make sense only in the continuous domain. In the discrete domain, we make use of the following properties.

Lemma 4.2 *Let A be the medial axis of f in a Euclidean space, and let $x \in A$ and $y \in S(x)$. (a) Construct a parametrized, open* line segment *from x to y. Then each point z on the segment has a unique source $s(z) = y$. (b) A has zero thickness,* i.e. *$A \subseteq \partial A$.*

Given two neighboring points $x \diamond y$ with $s(x) \neq s(y)$, lemma 4.2(b) suggests there is a medial point m with $S(m) = \{s(x), s(y)\}$ on the line segment between x, y. We therefore label *pair* (x, y) as medial, according to an extension of the *chord residue* criterion [15]. We first discuss our main algorithm.

Computation. Our *Weighted Medial Axis* (WMA) algorithm computes the medial axis $A(f)$ of image f given its weighted distance map $h = \mathcal{D}(f)$ and its source map s. Lemma 4.2(a) expresses the known fact that the medial axis function is associated to *peaks* (local maxima) and *ridges* of the distance map. We thus start with the *medial seed set*

$$A_+(f) = \{x \in X \,:\, h(x) \geq \max_{y \diamond x} h(y)\}, \qquad (5)$$

and continue propagating along $A(f)$ using a FIFO queue q. For each point x being processed, we SCAN 4-connected neighbors $y \diamond x$ to decide if (x, y) is a medial pair. We only PROPagate to x's 8-connected neighbors if x is found medial after SCANning. We record "medialness" by means of *residue* $r(x) = \max_{y \diamond x} \mathrm{res}(x, y)$ for $x \in X$ and the define the medial axis as $A(f) = \{x \in X \,:\, r(x) > 0\}$. Residue function res is discussed below.

Chord residue. To deal with singularities of the distance map in the discrete domain, Ogniewicz and Kübler use the chord residue [15]. They define it for binary shapes, as the difference between the length of a boundary curve segment and the corresponding chord length in a circle that is contained in the shape and bitangent to the boundary curve at the two endpoints. We use the weighted distance map (1) and see the distance value as a third dimension, or *height*. Recalling lemma 3.2, we define *source function* $\mathcal{S}(f)$ of f as the restriction of $\mathcal{D}(f)$ on the source set: $\mathcal{S}(f) = \mathcal{D}(f)|_{S(f)} = f|_{S(f)}$. Dually to the medial axis function, $\mathcal{S}(f) \subseteq \mathbb{E}$ is associated to *local minima* and *valleys* of the distance map. We generalize circles to *cones* lying below and bitangent to $\mathcal{S}(f)$, and 2D curve segments in $\mathbb{X}$ to *3D paths* along $\mathcal{S}(f)$ in $\mathbb{E}$. We measure distances with the *product metric* δ of the Euclidean metric d of 2D space $\mathbb{X}$ and the absolute difference of 1D space $\mathbb{V}$:

$$\delta(u, v) = d(u, v) + |h(u) - h(v)|, \quad u, v \in \mathbb{X}. \qquad (6)$$

Now, given two points $x, y \in X$ with sources $u = s(x), v = s(y)$, we generalize the *chord residue* as $\mathrm{res}(x, y) = \ell(u, v) - \delta(u, v)$ for $u \neq v$, or 0 otherwise.

Length function. *Length function* ℓ generalizes the *potential function* of [15] as the length of the shortest path (geodesic) connecting points $(u, f(u))$ and $(v, f(v))$ along the surface of the source function $\mathcal{S}(f)$ in space $\mathbb{E}$. Its computation is facilitated by the following.

Lemma 4.3 *The medial axis $A(f)$ is uniquely determined by the restriction $f|_{\partial S(f)}$ of function f on the* boundary *of its source set.*

In the discrete domain, we start by computing the source set $S(f) = \{x \in X \,:\, x \succ x\}$ and its discrete boundary w.r.t. 4-connectivity as

$$\partial S(f) = \{x \in S(f) \,:\, \exists y (y \diamond x \wedge y \in I(f))\}. \qquad (7)$$

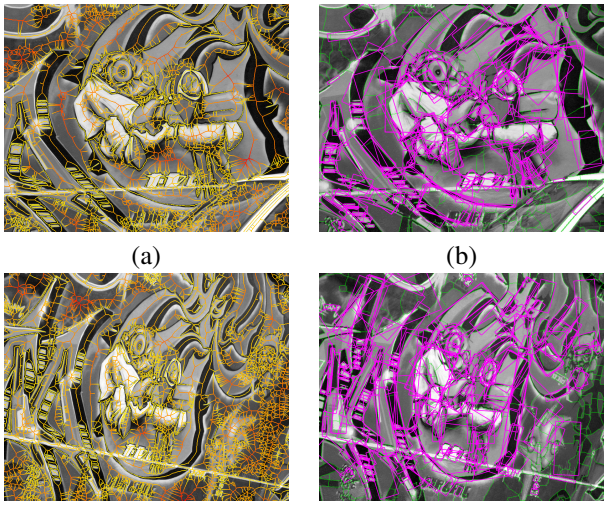


Figure 3. (a) Medial axis for `graffiti` scene image 1. Minimum (maximum) height on the medial axis is mapped to `yellow` (`red`), as in Fig. 1. (b) Detected features and underlying partition. (c), (d) The same for image 3 of the same scene. The medial axis appears to have changed form, but the features are quite stable.

We then construct a weighted graph H as a subgraph of grid G with vertex set $V(H) = \partial S(f)$, and weight function $w(e) = \delta(u, v)$ for edge $e = (u, v) \in E(H)$. We compute its *components* and the *faces* of each component. Then, seeing each face c as a cycle with start vertex v_0, we compute for each vertex v of c the *weight* $w_c(v)$ of path $(v_0, \ldots, v)$. Each vertex $v \in V(H)$ may belong to up to four faces. If $C(v)$ denotes the set of faces containing v, intersection $C(u, v) = C(u) \cap C(v)$ is either empty (if u, v belong to distinct components, in which case we define $\ell(u, v) = +\infty$) or contains exactly one common face c, associated to the component of $I(f)$ containing x, y. In the latter case,

$$\ell(u, v) = \min(\ell_c(u, v), w(c) - \ell_c(u, v)) \qquad (8)$$

where $\ell_c(u, v) = |w_c(u) - w_c(v)|$ and $w(c)$ is the total weight of face c. This is a *constant-time* operation.

Proposition 4.4 *(a) Given point pairs (x, y), (x', y') in the same component of interior set $I(f)$ with source pairs (u, v), (u', v'), respectively, define paths $\pi = (u, \ldots, v)$, $\pi' = (u', \ldots, v')$. If $\pi \subset \pi'$, then $\mathrm{res}(x, y) < \mathrm{res}(x', y')$. (b) WMA generates exactly one component of $A(f)$ for each component of $I(f)$. (c) Its complexity excluding initialization is $O(k)$, where $k = |A(f)|$.*

Hence, as in [15], the residue function is increasing w.r.t. *inward* moves along the medial axis, and pruning is as simple as thresholding. Figures 3(a),(c) show examples of what our weighted medial axis looks like on a gray-level image.

5. Feature detection

Medial axis decomposition. While the medial axis typically represents the shape of single object, we represent the *structure* of an entire image. We decompose it into components and construct a weighted graph $\mathcal{G}$ such that: (a) its vertices $\mathcal{V}$ correspond to local maxima (*peaks*) of the distance map; (b) its edges $\mathcal{E}$ correspond to local minima *along the medial axis* (*i.e.*, along ridges), therefore to *saddle points* of the distance map; (c) the weight of each edge is defined as the *height* at the associated saddle point. Peaks correspond to the interior of image regions, and saddle points to *mountain passes* between adjacent regions. Red components in Fig. 2(c) or 6(b) correspond to regions, each contains a peak, and each is represented in $\mathcal{G}$ as a vertex. Similarly, black points correspond to saddle points, and each is represented as an edge.

Our *medial axis decomposition* (MAD) algorithm constructs graph $\mathcal{G}$ given a distance map $h = \mathcal{D}(f)$ and the associated medial axis $A(f)$. We start with the distance peaks on the medial axis, $A_+(f) \cap A(f)$, and continue propagating downwards. We use again a *priority queue* q and propagate to 8-connected neighbors according to height, as in EGM. However, the priority level is now *negated* in PROP, because of the downward direction. We assign a component label $\kappa(x)$ to each $x \in X$, represented by a vertex of graph $\mathcal{G}$. We build $\mathcal{G}$ by gradually inserting a VERTEX whenever we first visit a peak with unlabeled neighbors, and an EDGE whenever two fronts with distinct labels meet.

Image partition. Next, we *partition* the entire image via a reconstruction operation. We exploit a *duality* property whereby this operation reduces to EGM algorithm. Recall that the distance map $\mathcal{D}(f)$ applies to functions f defined on domain X whereas the medial axis function $\mathcal{A}(f)$ is restricted to subset $A(f) \subset X$. Given any function $f : U \to \mathbb{V}$, we define the *extension* operator $f|^X = f \cup ((X \setminus U) \times \{-\infty\})$, which extends its domain to X with value $-\infty$ wherever f is not defined. We now define the *extended medial operator* $\mathcal{M}$ by $\mathcal{M}(f) = \mathcal{A}(f)|^X$ for $f \in \mathbb{F}$. Since $\mathcal{M}(f)$ is defined on domain X, we can apply distance or medial axis operators sequentially:

Proposition 5.1 *Given function f, let $g = \mathcal{M}(f)$ in a Euclidean space, and define $f' = -\mathcal{M}(-g)$, $g' = \mathcal{M}(f')$. Then source function $\mathcal{S}$ and medial axis function $\mathcal{A}$ are* dual*: (a) $-\mathcal{S}(-g) = \mathcal{A}(f)$, and (b) $\mathcal{S}(f') = -\mathcal{A}(-g)$. (c) This process is* stable*: $g' = g$.*

This result is quite condensed, but an one-dimensional example in Fig. 4 illustrates the idea. Proposition 5.1 suggests that we can define the *extended boundary operator* $\mathcal{B}$ by $\mathcal{B}(f) = -\mathcal{M}(-f)$ for $f \in \mathbb{F}$. Then, similarly to morphological *erosion* and *dilation*, the two operators are *dual*. Also, similarly to *opening* (*closing*), composition $\mathcal{B} \circ \mathcal{M}$ ($\mathcal{M} \circ \mathcal{B}$) is *idempotent* and has *fixed point* f iff $f = \mathcal{B}(g)$ ($f = \mathcal{M}(g)$) for some $g \in \mathbb{F}$.

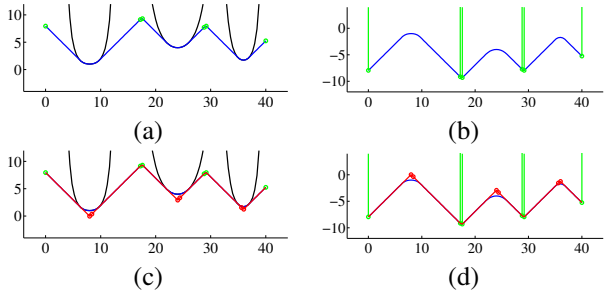


Figure 4. Illustrating duality of proposition 5.1 in one dimension. Functions in (b), (d) are negated versions of (a), (c); horizontal axis is X. (a) Black: f, blue: $\mathcal{D}(f)$, green dots: $\mathcal{A}(f)$. f is low at image boundaries, high inside regions. (b) Blue: $-\mathcal{D}(f)$, green line: $-g$ for $g = \mathcal{M}(f)$, green dots: $-\mathcal{A}(f) = \mathcal{S}(-g)$. (d) Red line: $\mathcal{D}(-g)$, red dots: $\mathcal{A}(-g)$. (c) Red dots: $\mathcal{S}(f') = -\mathcal{A}(-g)$. This is where fronts meet during partitioning.

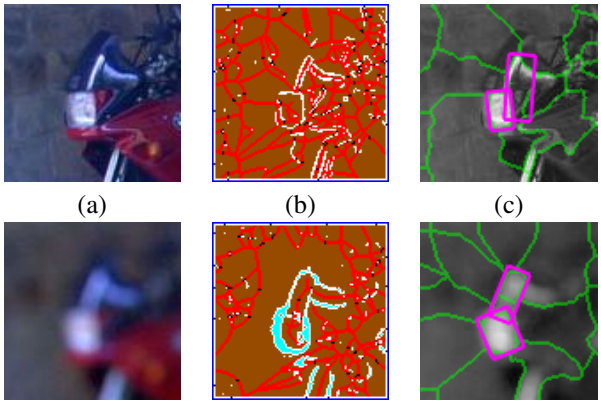
What we do in practice is that, given distance map $h = \mathcal{D}(f)$ and medial axis $A(f)$, we invoke EGM giving as input function g with $g(x) = -h(x)$ if $x \in A(f)$, or $+\infty$ otherwise. Further, we use label map κ from MAD and construct component labels $\kappa(s(x))$ for $x \in X$. This label map provides a partition of domain X.

Discussion. Propagation in MAD is equivalent to the *watershed segmentation* of the negated distance map restricted to the medial axis (i.e. on $\mathcal{A}(f)$) with peaks as markers. It is topological but based on a distance function that is also contrast-weighted. MAD also performs additional tasks like building the graph and finding markers (components) in parallel to propagation.

Similarly, image partitioning is equivalent to watershed segmentation of the negated distance map $-h$ on X with the components of MAD as markers. This differs from the typical *topological watershed* of binary region masks resulting from gray-level watershed; we are working on the weighted distance map resulting directly from the gray-level input. Most importantly, casting it as distance propagation enables the use of EGM, hence a very fast implementation.

Feature detection. We now have both a partition of domain X and a graph $\mathcal{G}$ representing weighted adjacency relations between components. We define a *feature* to be the *union of any number of adjacent components*. We *group* components in non-increasing order of edge weights, using the union-find algorithm. Each newly acquired component is considered as a potential feature, as follows.

The source set is frequently disconnected or *fragmented*. *Gaps* appear either due to variation of f along edges, or to region shape. Examples are shown in Figs. 1, 2, 5. MAD helps overcome fragmentation because every gap is associated to a *saddle point* of the distance map. We still get a component associated to an image region even if its



(a) (b) (c)

(d) (e) (f)

Figure 5. (a) Input image, detail of `bikes` scene, image 1. (b) Point labels. (c) Image partition and detected features. (d),(e),(f) Same for image 6 of the same scene. Color legend for point labels: `white`: source boundary; `cyan`: source interior; `red`: medial axis; `black`: saddle point; `blue`: image boundary.

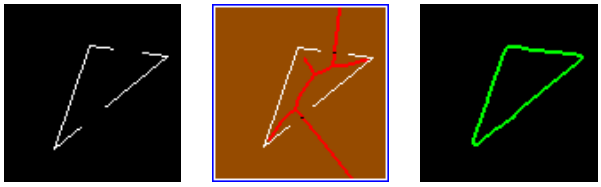
boundary is fragmented, because the distance map is maximized somewhere in the interior. The surrounding saddle points give rise to edges of graph $\mathcal{G}$. For each edge $e \in \mathcal{E}$, let $x(e)$ be the saddle point where e is generated. Then the width of the associated gap is $w(x(e))$, where $w(x) = d(x, s(x)) = h(x) - h(s(x))$ for $x \in X$. Given component (vertex) κ with area $a(\kappa)$ and incident edge set $E(\kappa) \neq \emptyset$, we define its *shape fragmentation factor* as

$$\phi(\kappa) = \frac{1}{a(\kappa)} \sum_{e \in E(\kappa)} w^2(x(e)), \tag{9}$$

whereas $\phi(\kappa) = 0$ if $E(\kappa) = \emptyset$ (κ is isolated). This factor is a dimensionless, scale invariant quantity. It is higher for a single large gap, lower for several smaller ones, and identically zero for closed shapes. As shown in Fig. 6, it measures how well a region is enclosed by boundaries (which may be incomplete), capturing the *Gestalt principle of closure*.

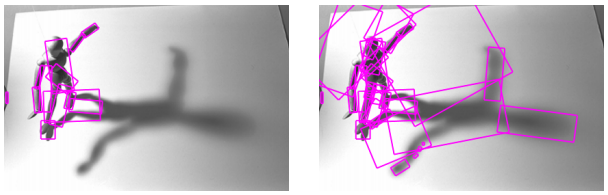
Iterating over edges $e \in \mathcal{E}$, we compute the sum of squared gap widths appearing in (9) for each component, prior to component grouping. Similarly, iterating over points $x \in X$, we collect the statistics of each component up to second order. We recursively update all quantities during grouping. We *select* a component κ as an image *feature* if $\phi(\kappa) < \tau$ and measure its position (centroid) and local shape (covariance matrix) via its statistics. Threshold $\tau > 0$ controls the selectivity of the detector.

Height function. Given an input image f_0, we now define function f used in distance computation. Starting with gradient magnitude $g = |\nabla f_0|$, define $f(x) = \sigma / g(x)$ for $x \in X$, where σ is a *scale* parameter. This generalizes the $0/\infty$ *indicator function* used in binary distance transform.



(a) (b) (c)

Figure 6. (a) Binary input image (no gradient used here). (b) Point labels; color legend is the same as in Fig. 5. (c) Image partition. There are two saddle points on the medial axis of (b), and two gaps with widths w_1, w_2. The fragmentation factor of triangle κ in (c) is $\phi(\kappa) = (w_1^2 + w_2^2)/a(\kappa)$ where $a(\kappa)$ is its area.



(a) (b)

Figure 7. Medial features on `mannequin` image with $\sigma = 4$ and (a) $\tau = 0.3$, (b) $\tau = 0.7$.

6. Experiments

Datasets, evaluation protocol, and tuning. We first carry out parameter tuning of our *Medial Feature Detector* (MFD). *Shape fragmentation* threshold τ controls the selectivity of the detector. *Scale* parameter σ controls the level of detail retained by the weighted distance map, seen as a nonlinear filtering operator. Without constructing a scale space, MFD is still able to detect regions at different scales in the same image, as in Fig. 7. We obtain gradient magnitude via convolution with Gaussian derivative $g = |\nabla G_t * f_0|$ using derivation scale $t = 0.5$, and normalize g in $[0, 1]$. We choose $\sigma = 4$ and $\tau = 0.6$, giving balanced performance across difference images. An example with these parameters is shown in Fig. 3(b),(d).

We then conduct two sets of experiments. One involves *matching* across viewpoint, zoom, rotation, light, and blur changes according to the dataset and evaluation protocol of [14]. We measure performance in terms of *repeatability* and *matching score*, using the 1-NN strategy and SIFT descriptors for all detectors. We give here results for the four scenes `graffiti`, `boat`, `wall`, and `bikes`, of six images each. The other experiment involves larger scale *retrieval* using a BoW model and fast spatial matching (FastSM), following the experimental setup of [16] and measuring performance in terms of mean Average Precision (mAP) score. Here we use the *Oxford* 5K dataset, comprising $5,062$ images with 55 queries. We extract features and construct 50K and 200K vocabularies. We construct an inverted index and rank images with TF-IDF, without stop list. We re-rank with FastSM and verify images having at

	Features ($\times 10^6$)		Memory (MB)	
Detector	Total	Used	50K	200K
MFD	9.32	9.32	62	68
Hessian-affine	29.03	11.61	116	126
MSER	13.33	11.33	72	76
SURF	4.24	4.24	30	34
SIFT	11.13	11.13	76	82

Table 1. Total number of features in *Oxford* 5K dataset, and features used in vocabulary construction. Total memory required for the inverted index, for the 50K and 200K vocabularies.

Query phase	Inv. index		Re-ranking	
Detector	50K	200K	50K	200K
MFD	5.32	3.96	26.8	8.2
Hessian-affine	11.80	6.72	161.3	44.9
MSER	6.25	4.30	60.9	13.2
SURF	3.43	3.00	2.5	1.7
SIFT	7.01	4.61	35.8	8.4

Table 2. Average query time in ms. Inverted index refers to total TF-IDF ranking time; re-ranking to FastSM per image pair.

mAP	Inv. index		Re-ranking	
Detector	50K	200K	50K	200K
MFD	**0.515**	**0.580**	**0.568**	**0.617**
Hessian-affine	0.488	0.573	0.537	0.614
MSER	0.473	0.544	0.537	0.589
SURF	0.488	0.531	0.497	0.536
SIFT	0.395	0.457	0.434	0.495

Table 3. Mean average precision, with and without re-ranking.

least 4 inliers. Times are measured on a 3GHz Core i7-950 processor with 12GB memory, with our own C++ implementations.

Repeatability and matching score experiments. Fig. 8 presents repeatability and matching score measurements of MFD compared to the six methods studied in [14], that is, Hessian-Affine, Harris-affine, MSER, IBR, EBR and Salient Regions. MFD achieves excellent matching score, outperforming all other detectors in certain scenes, while repeatability is also high in most cases. Its performance is striking at the `wall` scene. This is justified because features are identified via boundaries rather than intensity differences. Good performance is accompanied by a reasonably small number of responses. Still, as shown in Fig. 8(d), these fewer features provide good image *coverage*. Thanks to shape fragmentation, highly textured areas do not give any response, *e.g.* the grass in the `boat` scene.

Retrieval experiments. Tables 1, 2, 3 present statistics on indexing space, average query time, and mean average precision (mAP), respectively. Indexing space and query time depend on the average number of features per image, and the objective is highest mAP with reasonable space/time requirements. All experiments are conducted for the two 50K and 200K vocabularies. We use up to approximately 11M features for vocabulary construction, selecting features at random when more are available. MFD outperforms all detectors in terms of retrieval performance, followed very closely by Hessian-affine, with the difference being even smaller after re-ranking. The latter is however quite impractical in terms of memory and query times.

7. Discussion

Without any explicit scale space construction, and without any inherent affine covariance properties, our medial feature detector achieves state of the art performance in image matching and retrieval experiments. An interesting future direction may be the exploitation of the exact region shape, or the extension to other types of features like corners. We also see a number of different directions like segmentation, edge detection and grouping, or shape-based object detection. More can be found at our project homepage[1] including the MFD executable code and documentation.

Acknowledgements. We thank Prof. Petros Maragos for helpful discussions. We are grateful to our colleague Giorgos Tolias for conducting the retrieval experiments.

References

[1] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. From contours to regions: an empirical evaluation. In *CVPR*, pages 2294–2301, 2009. 1

[2] H. Bay, T. Tuytelaars, and L. Van Gool. SURF: Speeded up robust features. In *ECCV*, 2006. 1

[3] M. Demirci, A. Shokoufandeh, Y. Keselman, L. Bretzner, and S. Dickinson. Object recognition as many-to-many feature matching. *IJCV*, 69(2):203–222, 2006. 1

[4] M. Donoser, H. Riemenschneider, and H. Bischof. Linked edges as stable region boundaries. In *CVPR*, pages 1665–1672, 2010. 1

[5] P. Felzenszwalb and D. Huttenlocher. Distance transforms of sampled functions. Technical report, 2004. 2

[6] D. Gavrila and V. Philomin. Real-time object detection for smart vehicles. In *ICCV*, volume 1, pages 87–93, 1999. 1

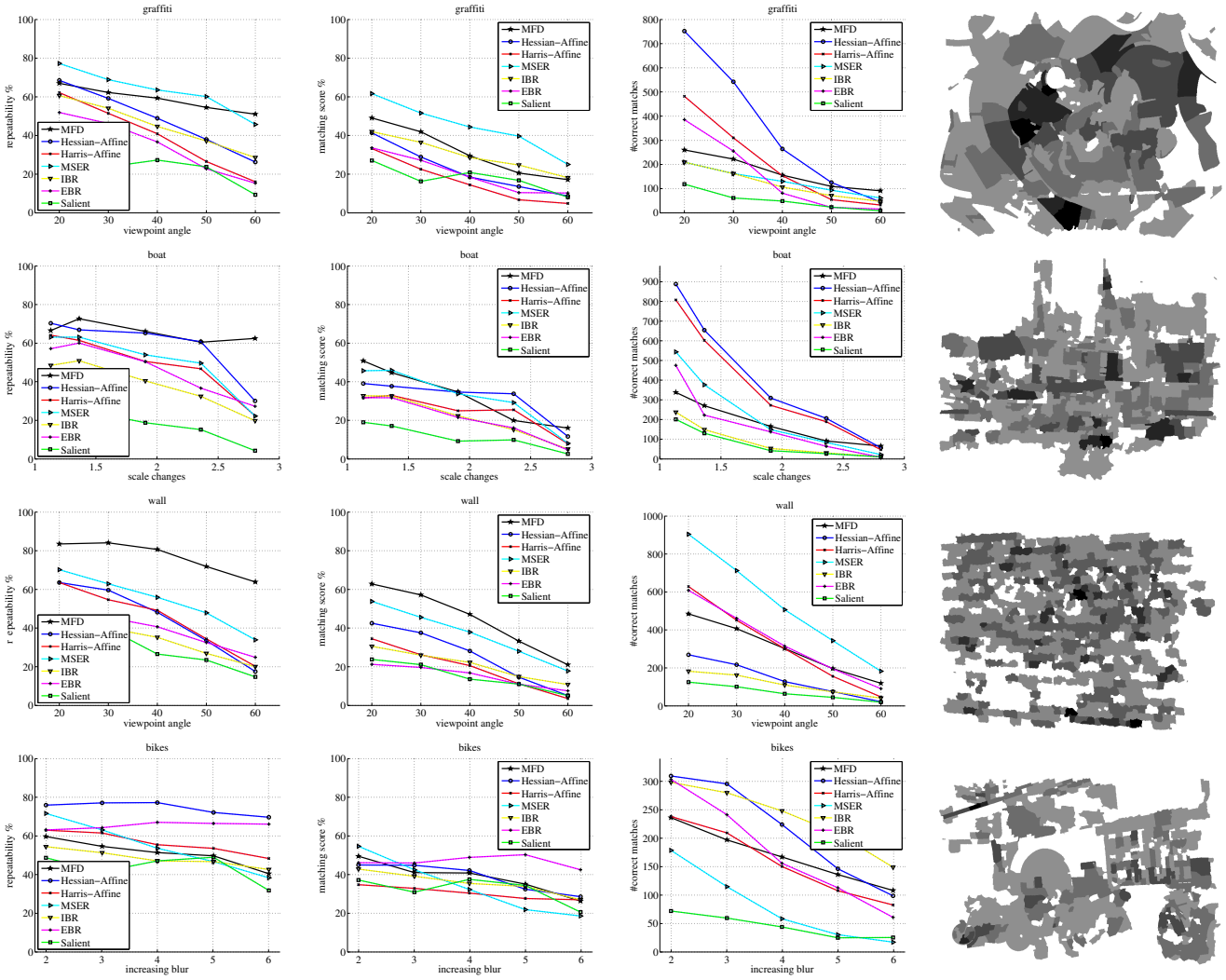
[7] S. Kim. An $O(N)$ Level Set Method for Eikonal Equations. *SIAM journal on scientific computing*, 22(6):2178–2193, 2001. 2

[8] L. Linardakis and N. Chrisochoides. A static medial axis domain decomposition for 2d geometries. *ACM Transactions on Mathematical Software*, 34(1):1–19, 2005. 2

[9] D. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004. 1

[10] P. Maragos and M. Butt. Curve evolution, differential morphology, and distance. *Fundamenta Informaticae*, 41:91–129, 2000. 2

[1] http://image.ntua.gr/iva/research/medial_features



(a) (b) (c) (d)

Figure 8. (a) Repeatability, (b) matching score, (c) correct matches, and (d) coverage for image 1. From top to bottom, (scene / #features in image 1 / detection time): (`graffiti` / 530 / 0.55s), (`boat` / 665 / 0.72s), (`wall` / 876 / 1.08s), and (`bikes` / 545 / 0.84s).

[11] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide baseline stereo from maximally stable extremal regions. In *BMVC*, 2002. 1

[12] F. Meyer and S. Beucher. Morphological segmentation. *JVCIR*, 1(1):21–46, 1990. 2

[13] K. Mikolajczyk and C. Schmid. Scale & affine invariant interest point detectors. *IJCV*, 60(1):63–86, 2004. 1

[14] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. van Gool. A comparison of affine region detectors. *IJCV*, 65(1):43–72, 2005. 6, 7

[15] R. Ogniewicz and O. Kübler. Hierarchic voronoi skeletons. *Pattern Recognition*, 28(3):343–359, 1995. 2, 3, 4

[16] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *CVPR*, 2007. 6

[17] K. Rapantzikos, Y. Avrithis, and S. Kollias. Detecting regions from single scale edges. In *ECCV SGA Workshop*. Springer, 2010. 1

[18] K. Siddiqi, S. Bouix, A. Tannenbaum, and S. Zucker. Hamilton-jacobi skeletons. *IJCV*, 48(3):215–231, 2002. 2

[19] T. Tuytelaars and L. Van Gool. Content-based image retrieval based on local affinely invariant regions. In *Visual Information Systems*, 1999. 1

[20] P. Verbeek and B. Verwer. Shading from shape, the eikonal equation solved by grey-weighted distance transform. *Pattern Recognition Letters*, 11(10):681–690, 1990. 2

[21] L. Vincent and P. Soille. Watersheds in digital spaces: an efficient algorithm based on immersion simulations. *PAMI*, 13(6):583–598, 2002. 1, 2

[22] L. Yatziv, A. Bartesaghi, and G. Sapiro. $O(N)$ Implementation of the fast marching algorithm. *Journal of Computational Physics*, 212(2):393–399, 2006. 2