# W$\alpha$SH: Weighted $\alpha$-Shapes for Local Feature Detection

Christos Varytimidis, Konstantinos Rapantzikos, and Yannis Avrithis

National Technical University of Athens
{chrisvar,rap,iavr}@image.ntua.gr

**Abstract.** Depending on the application, local feature detectors should comply with properties that are often contradictory, *e.g.* distinctiveness vs. robustness. Providing a good balance is a standing problem in the field. In this direction, we propose a novel approach for local feature detection starting from sampled edges. The detector is based on shape stability measures across the *weighted $\alpha$-filtration*, a computational geometry construction that captures the shape of a non-uniform set of points. The extracted features are blob-like and include non-extremal regions as well as regions determined by cavities of boundary shape. Overall, the approach provides distinctive regions, while achieving high robustness in terms of *repeatability* and *matching score*, as well as competitive performance in a large scale image retrieval application.

## 1 Introduction

Local features are popular in computer vision tasks due to their invariance to illumination and viewpoint changes. Ideally, local features should comply to a set of properties, namely, *repeatability*, *distinctiveness*, *locality*, *quantity*, *accuracy* and *computational efficiency* [25]. These properties are often contradictory and cannot be satisfied simultaneously, like distinctiveness vs. locality or robustness and quantity vs. computational efficiency. Hence, although several feature detectors have been proposed, there is still a growing need for methods offering better balance between the desired properties.

State-of-the-art methods are based on measures of first- and second-order gray value derivatives (*e.g.* second-moment matrix or Hessian matrix [17]) or on intensity variations [23][14] and—depending on the application—can be considered quite successful. Image edges on the other hand, though bound to naturally stable structures, have not been so successful yet. In this paper, we conjecture that, within a multi-scale representation, a distinctive region approximately maintains its shape across scale, as determined by surrounding edges.

We therefore introduce a detector that groups edge points based on location, gradient strength and local shape. To capture local shape, we employ *$\alpha$-shapes*, a well-known method in computational geometry introduced by Edelsbrunner *et al.* [8]. In doing so, we devise an efficient way to overcome the main weakness of the method, namely automatic selection of the value of $\alpha$ that best represents

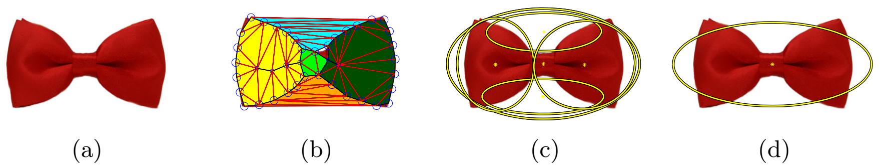

**Fig. 1.** Feature detection based on $\alpha$-shapes. (a) Input image. (b) Edge samples with assigned weights as squared radii (blue) and triangulation. Triangle colors denote distinct connected components of the $\alpha$-complex. Detected features by: (c) W$\alpha$SH detector and (d) MSER.

the underlying point set. We also show how noisy points or groupings are automatically filtered out by a shape-based stability measure and how the entire method is controlled by a simple and intuitive parameter. A detection example is given in Fig. 1.

We treat a local feature as a region delineated by a set of points sampled from its contour. Although not critical, we use a roughly uniform sampling scheme along binary edges found by an edge detector and employ $\alpha$-shapes to capture the evolution of local shape in a multi-scale setting. $\alpha$-shapes can be thought of as a generalization of the *convex hull*, being parametrized by scalar $\alpha \geq 0$. Starting from the convex hull of the point set for $\alpha = \infty$, they reduce to the set itself at the other extreme, $\alpha = 0$. An $\alpha$-shape is a *filtration*, a partial ordering of Delaunay simplices (edges and triangles in two dimensions) [27].

*Weighted* $\alpha$*-shapes* [7] provide a richer description of the input. For a single value of $\alpha$, *i.e.* for a single scale, weighted $\alpha$-shapes capture different levels of spatial details. The weighted $\alpha$-filtration is the multi-scale representation we adopt. To capture the evolving topology of local regions in the filtration, we employ a *component tree*, similar to [19]. By applying stability measures on the resulting representation we select distinctive and repeatable local features.

The remaining of the paper is organized as follows: In section 2 we discuss related work, and then describe our detector in detail in section 3. Results are provided in section 4 followed by conclusions and discussion in section 5.

## 2 Related Work

The literature on local feature detection is rich and since the early work of Beaudet [3] and Harris and Stevens [9], based on the Hessian matrix and the second moment matrix respectively, many detectors have been proposed grounding on similar or novel ideas. In his inspiring work, Lindeberg *et al.* extended detectors by making them scale-invariant [11] and establishing the theoretical foundations for making them affine-invariant [12]. Based on these foundations, Lowe proposed the *scale invariant feature transform* (SIFT) in [13] and Mikolajczyk *et al.* the affine-adapted version of the Harris and Hessian detectors [16], [17].

The *maximally stable extremal regions* (MSER) of Matas *et al.* [14], one of the best performing detectors in [17], detects regions of stable intensity and therefore avoids common problems of gradient-based methods like localization accuracy and noise. The recent trend of achieving a good balance between efficiency and performance has led to a group of computationally efficient detectors like SURF [2], an approximate version of SIFT, and FAST [23], introducing fast corner detection based on an intensity comparison test in a small neighborhood. Recently, BRISK [10] build on FAST detector and provide real-time detection and description of local features with matching performance being comparable to SIFT and SURF.

Although naturally meaningful, image edges have attracted less attention with the reasons being mainly related to the lack of stable edges and the computational inefficiency. One of the earliest attempts, *edge-based region detector* (EBR), starts from corner points and exploits nearby edges by measuring photometric quantities across them. It is suitable for scenes containing intersection of edges (*e.g.* man-made structures like buildings), but not for generic matching, as shown in [17]. Mikolajczyk *et al.* in [18] propose an edge-based detector that combines Canny edge detection with automatic scale selection and use it for object recognition. For efficiency, they start from densely sampled edge points, an approach that we also adopt. Rapantzikos *et al.* compute the binary distance transform of Canny edges and detect regions by grouping its local maxima, guided by the gradient strength of nearby edges [22].

There are also methods that exploit gradient strength directly, without edge detection. Zitnick *et al.* [26] apply an oriented filter bank to the input image and detect *edge foci* (EF), i.e. points that are roughly equidistant from edgels with orientations perpendicular to the points. The idea is quite interesting, but computationally expensive. Avrithis *et al.* compute the weighted medial axis, decompose it into meaningful parts and group regions by taking both contrast and shape into account [1]. *Medial features* (MFD) are then detected based on shape. Although more information is used, gradient strength is more sensitive to lighting and scale variations than binary edges.

Computational geometry is rich in applications based on $\alpha$-shapes, as opposed to computer vision. One of the earliest applications has been to surface reconstruction from an unorganized set of points [7]. Teichmann *et al.* [24] have used them to reconstruct the shape of point sets and achieve a more accurate separation of surfaces, but they rely heavily on user input. *Conformal $\alpha$-shapes* and the corresponding filtration, introduced by Cazals *et al.* in [5], are based on a global scale parameter $\hat{\alpha}$ and two local ones, adjusted to some neighborhood of each point. They show interesting results on surface reconstruction of non-uniformly sampled surfaces. In the same direction, Zomorodian *et al.* [27] predict protein structure by employing $\alpha$-shapes to detect interacting atoms in a protein molecule.

Furthermore, $\alpha$-shapes have been used for studying *pockets*, defined as regions with limited accessibility from the outside, measuring the surface area and volume of macromolecules, and the packing density of proteins [7]. $\alpha$-shapes have been also used to reconstruct boundaries of noisy edge maps. Starting from binary edge samples, Meine *et al.* [15] construct the Delaunay triangulation and select a subset of its edges to complete the object boundaries. Their main criteria are triangle size and average color.

Building on edge-based methods, we start from sampled edges like [18]. We use shape as the main selection criterion, a choice that bears similarities to [1], although our geometric representation is entirely different. There are also similarities to MSER [14], in the sense that we employ a hierarchical representation of nested sets. However, we apply this representation on the $\alpha$-filtration rather than the level sets of image intensity. As a by-product, our method is able to detect gray regions that are adjacent to both brighter and darker ones, as opposed to MSER that can only detect bright or dark extremal regions, as in Fig. 3bc. Furthermore, the proposed detector can handle regions determined by cavities of the boundary shape, as in Fig. 1, bearing similarities to the *pockets* [7], and regions that are not enclosed by complete boundaries like in Fig. 3a.

## 3 W$\alpha$SH Detector

### 3.1 Image Representation

We assume an input (grayscale) image is given as a function $f$ on the plane and that $g$ is its gradient magnitude $\|\nabla f\|$ normalized in interval $[0, 1]$. We apply a binary edge detector on $g$ and sample the resulting edge map $E$ to obtain a discrete set of edge points $P \subseteq \mathbb{R}^2$. Sampling is roughly uniform along edges with a fixed *sampling interval* $s$, so samples will typically not correspond to keypoints like maxima of curvature. Although sparsely sampled along the edges, the points in $P$ capture the shape of the detected boundaries. For each point $p \in P$, we define its *weight* $w(p) \geq 0$ to be multiple of its gradient strength,

$$w(p) = g(p) \left(\frac{s}{2}\right)^2, \tag{1}$$

with $g(p) \in [0, 1]$. The reason for this choice of weight $w$ will be made clear at the end of section 3.2. In practice, the binary edge map is obtained by the *Canny* edge detector [4]. We do not rely on a clear edge map and therefore the high and low hysteresis thresholds of the detector are kept fixed.

### 3.2 Weighted $\alpha$-shapes

The definition of $\alpha$-shapes is based on the Delaunay triangulation. Weighted $\alpha$-shapes are based on its generalization, the *regular triangulation* formed by replacing the Euclidean distance by the power to weighted points. The discussion given here mostly follows [7], but simplifies for the case of 2 dimensions.

A point $p \in P$ along with its weight $w(p) \geq 0$ makes up a pair $(p, w(p))$ that is called a *weighted point* and can be seen as a *circle* centered at $p$, with squared radius $w(p)$. We will use the same symbol $p$ for both a circle or weighted point, and will disambiguate as necessary. Given two points $p, q \in P$, define

$$\pi(p, q) = \|p - q\|^2 - w(p) - w(q). \tag{2}$$

Circles $p, q$ intersect at right angles iff $\pi(p, q) = 0$; we then say that $p, q$ are *orthogonal*. Given a subset $T \subseteq P$ of three points, there is a unique circle that is orthogonal to all circles of $T$. We denote the corresponding weighted point by $c_T$. The triangle $\sigma_T$ with vertices given by $T$ is called *regular* if

$$\pi(p, c_T) = 0 \text{ for all } p \in T, \tag{3}$$

$$\pi(p, c_T) > 0 \text{ for all } p \in P \setminus T, \tag{4}$$

where (3) is equivalent to $c_T$ being orthogonal to all points of $T$. The collection $\mathcal{R}$ of all regular triangles over $P$ is called the *regular triangulation* of $P$. Observe that, if $w(p) = 0$ for all $p \in T$, the weighted point $c_T$ is the *circumcircle* of triangle $\sigma_T$. In this unweighted case, the triangulation reduces to *Delaunay.*

The collection $\mathcal{K}$ of all triangles in $\mathcal{R}$ and their faces (line segments and points) is a *simplicial complex*: each triangle, line segment or point is a 2-, 1-, or 0-simplex respectively, and we will refer to it as *simplex* in general. If we define a *size* $\rho_T \geq 0$ for each simplex $\sigma_T \in \mathcal{K}$, then the *weighted $\alpha$-complex* of $P$ is the subset of $\mathcal{K}$ containing all simplices up to a given size $\alpha \geq 0$,

$$\mathcal{K}_\alpha = \{\sigma_T \in \mathcal{K} : \rho_T < \alpha\}. \tag{5}$$

Finally, the *weighted $\alpha$-shape* of $P$ [7] is the union of all such simplices,

$$\mathcal{W}_\alpha = \bigcup_{\sigma \in \mathcal{K}_\alpha} \sigma. \tag{6}$$

Observe that $\mathcal{W}_{+\infty}$ is the *convex hull* of $P$. It remains to define the *size* $\rho_T$ of a point, line segment or triangle $\sigma_T$, when $T$ contains $1, 2$ or $3$ points of $P$, respectively. The size of a point $p \in P$ is equal to its weight, $\rho_{\{p\}} = w(p)$. Given a set of two points $T = \{p, q\} \subseteq P$, the size $\rho_T$ of the associated line segment is the squared radius of the unique circle that is orthogonal to circles $p, q$, whose center is collinear with $p, q$, as in Fig. 2a. Similarly, given a set of three points $T = \{p, q, r\} \subseteq P$, the size $\rho_T$ of the associated triangle is the squared radius of the unique circle that is orthogonal to circles $p, q, r$, as in Fig. 2b. From the definition of the weight function in (1) follows that for two points $p, q$ along an image edge, sizes $\rho_{\{p\}}$ and $\rho_{\{q\}}$ take values in $[0, (s/2)^2]$. This ensures that their corresponding circles, as shown in Fig. 2, do not overlap and the line segment $T = \{p, q\}$ has $\rho_T \geq 0$.

## 3.3 Component Tree

All simplices of $\mathcal{K}$ are typically ordered by ascending size to obtain what is called a *weighted $\alpha$-filtration* [7]. In this work, we deviate from this standard setting in two ways. First, we only consider *triangles* and their *edges* (line segments) discarding points $p \in P$ themselves. We thus construct complex

$$\mathcal{K}' = \{\sigma_T \in \mathcal{K} : |T| \geq 2\}. \tag{7}$$

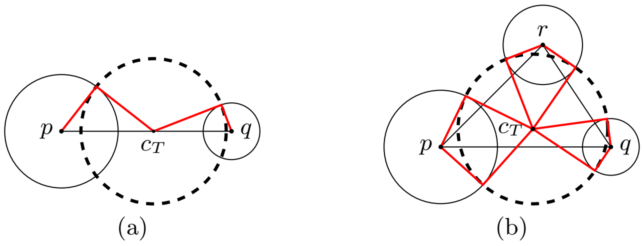

**Fig. 2.** (a) Size of a 2-point set $\{p, q\}$, as the squared radius of a circle that is orthogonal to $p, q$. (b) Size of a 3-point set $\{p, q, r\}$, as the squared radius of the circle that is orthogonal to $p, q, r$.

This is justified because edge size helps control connectivity of triangles, but points do not. Second, contrary to (5), we consider the *upper α-complex*

$$\overline{\mathcal{K}}_\alpha = \mathcal{K}' \setminus \mathcal{K}_\alpha = \{\sigma_T \in \mathcal{K}' : \rho_T \geq \alpha\} \tag{8}$$

ordered by *descending* $\alpha$. As in [7], we need only consider a finite set of values for $\alpha$. In particular, we sort all $\sigma_T \in \mathcal{K}'$ by descending size $\rho_T$ to obtain sequence $(\sigma_1, \ldots, \sigma_n)$ where $n = |\mathcal{K}'|$. If $\rho_i$ is the size of $\sigma_i$ for $i = 1, \ldots, n$, then $\rho_1 \geq \ldots \geq \rho_n$. Now, if $K_i = \{\sigma_1, \ldots, \sigma_i\}$, we obtain the *upper α-filtration*

$$\emptyset = K_0 \subseteq K_1 \subseteq \cdots \subseteq K_n = \mathcal{K}'. \tag{9}$$

Starting from the largest element of size $\rho_1$ and decreasing the value of $\alpha$ towards $\rho_n$, the upper $\alpha$-complex models the growing *cavities* of the original shape. To capture its evolving topology, we construct a *component tree*, similar to [19].

To define *connectedness* on the complex, we specify neighboring relations as follows: the neighbors of each triangle $\sigma_T \in \mathcal{K}'$ (with $|T| = 3$) are its three edges, while, the neighbors of each edge $\sigma_E \in \mathcal{K}'$ (with $|E| = 2$) are the two adjacent triangles in the triangulation. We denote the *neighborhood* of simplex $\sigma \in \mathcal{K}'$ by $N(\sigma)$. Since an edge is not larger than its two adjacent triangles, the intuition is that this edge can keep the two triangles disconnected until it is processed itself. Eventually, this timing depends on image gradient.

Given this neighborhood system, we start off with all simplices in $\mathcal{K}'$ being individual components, and process them in descending order of size, joining them with their neighbors that have already been processed. This process is specified in Algorithm 1.

## 3.4 Feature Selection

While tracking the evolution of the connected components of the upper $\alpha$-complex, we measure the significance of changes in its topology so as to decide on stability or distinctiveness of relevant image regions. Whenever a new component arises by joining two adjacent components, these two components are

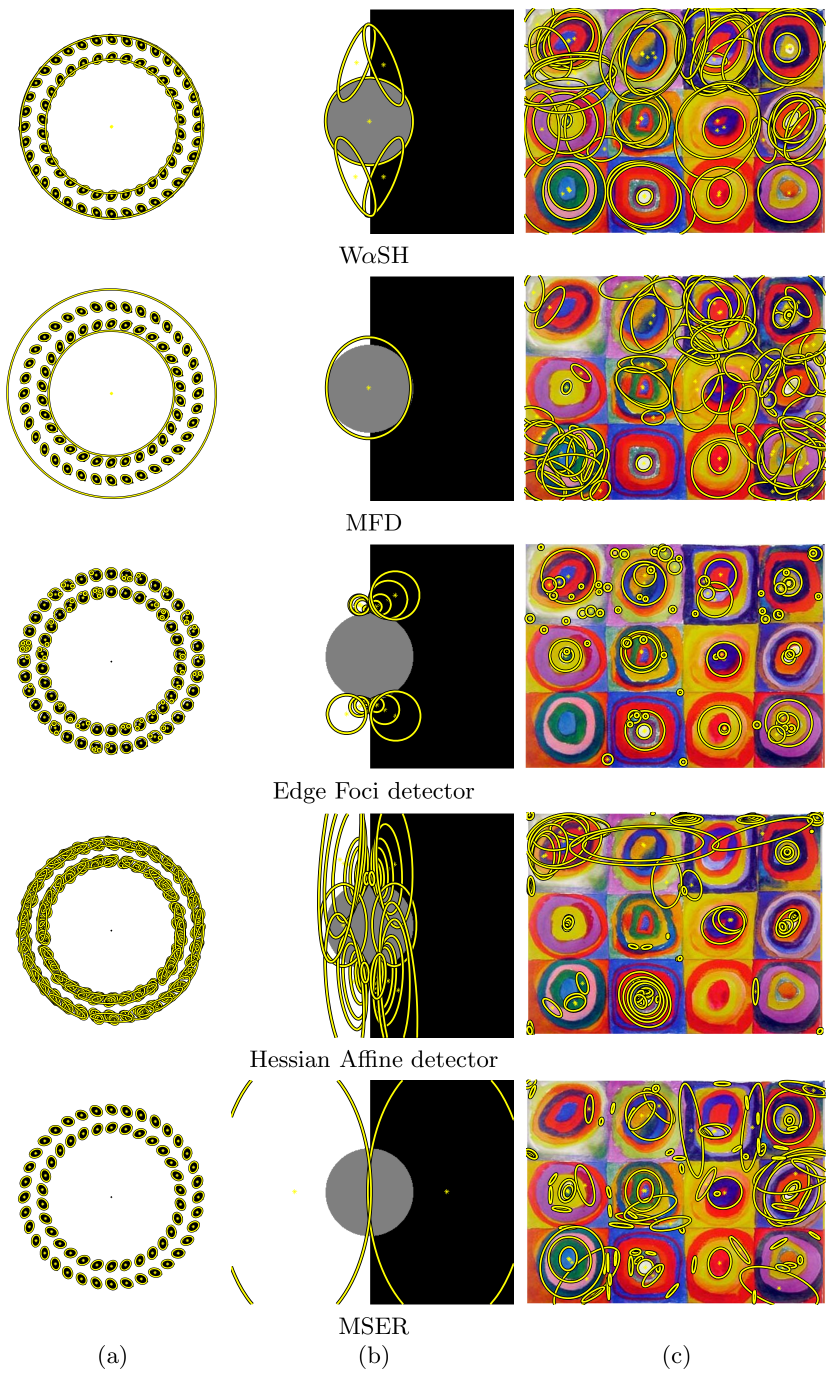

**Fig. 3.** Detection of (a) incomplete boundaries and (b)(c) non-extremal regions

individually considered as potential *features*. We choose to base the decision on simple measurements requiring minor computational overhead.

In particular, consider component $\kappa_U$, which is a set of simplices (edges and triangles) that have been processed. When $\kappa_U$ is joined through a boundary edge $\sigma_T$ of size $\rho_T$ to another component, we compute the *strength* of $\kappa_U$ as the ratio

$$s(\kappa_U) = \frac{a(\kappa_U)}{\rho_T}, \tag{10}$$

where $a(\kappa_U)$ is its total area—precisely, the area of the union of all simplices in $\kappa_U$. What remains to do is *select* component $\kappa_U$ as a feature, if its strength is larger than a fixed threshold $\tau$, and fit a patch over its region support for descriptor extraction. Observe that, since we are processing simplices in descending order of size, $\rho_T$ stands in fact for the largest opening or the weakest gradient strength over the boundary of component $\kappa_U$. It follows that strength favors large components with small (or no) openings in their boundary. Since size $\rho$ measures squared length, component strength is a scale invariant quantity.

Some examples are shown in Fig. 3. We can detect regions that are not completely bounded by edges, as the two perceived concentric circles in Fig. 3a, or regions that are not extremal in the intensity domain, in contrast to MSER, as shown in Fig. 3bc.

## 3.5 Algorithm

The pseudocode for the complete method is given in Algorithm 1. The first steps are straightforward. Function Complex computes complex $\mathcal{K}$ and size map $\rho$, given the regular triangulation $\mathcal{R}$. Function Neighbor computes the neighborhood map $N$ of a set of edges and triangles, while $\mathcal{K}'$ is as defined in (7). Function Area computes the area of a triangle given its vertices in point set $T$, and it returns 0 for an edge.

We use two different tree structures to keep track of connected components, as in [19]. The first is a forest where each simplex serves as the root of a subtree containing all larger simplices in the same component. We maintain a list of children for each simplex, using function AddChild, while all simplices are initially assumed to be leaves. The second is a standard disjoint set forest, with simplices pointing only to their parent, and with functions MakeSet, Find and Union having their ordinary meaning [6]. The two structures are interconnected via pointer *root*. The second is used for efficiency, while the first for information access. In particular, for each selected component, we collect its simplices via breadth-first search in the subtree of its root simplex and fit a patch to their convex hull—this is not shown in Algorithm 1.

The computational cost of the regular triangulation is $O(n \log n)$ and the required space is $O(n)$. Hence the cost of weighted $\alpha$-shape construction is also $O(n \log n)$. The triangulation only needs to be computed once; the remaining part of the algorithm that is based on the component tree is quasi-linear in $n$ [19], that is, linear for all practical purposes.

**Algorithm 1.** W$\alpha$SH Feature Detection

```
input  : grayscale image $f$
output: local feature set $F$

$g \leftarrow \|\nabla f\| / \max\{\|\nabla f\|\}$                          // normalized gradient
$E \leftarrow$ CANNY($g$)                                                     // edge detection
$P \leftarrow$ SAMPLE($E$)                                                    // edge sampling
$\mathcal{R} \leftarrow$ REGULAR($P$)                                         // regular triangulation
$(\mathcal{K}, \rho) \leftarrow$ COMPLEX($\mathcal{R}$)                        // simplicial complex + sizes
$N \leftarrow$ NEIGHBOR($\mathcal{K}'$)                                        // neighborhood system

$F \leftarrow \emptyset$
foreach $\sigma_T \in \mathcal{K}'$ do                                        // initialize each simplex
    MAKESET($\sigma_T$)                                                       // as an individual component
    $\sigma_T.area \leftarrow$ AREA($T$)                                      // with its own area
    $\sigma_T.root \leftarrow \sigma_T$

foreach $\sigma_T \in \mathcal{K}'$ in descending order of $\rho_T$ do        // current simplex
    $\kappa_T \leftarrow$ FIND($\sigma_T$)                                    // current component $\kappa_T$
    $r_T \leftarrow \kappa_T.root$
    foreach $\sigma_U \in N(\sigma_T)$ such that $\rho_U \geq \rho_T$ do      // adjacent, processed simplex
        $\kappa_U \leftarrow$ FIND($\sigma_U$)                                // adjacent component $\kappa_U$
        $r_U \leftarrow \kappa_U.root$
        if $\kappa_T \neq \kappa_U$ then                                      // if different components
            if $|U| = 3 \wedge r_U.area/\rho_T > \tau$ then                   // if $\kappa_U$ is triangle & strong
                $F \leftarrow F \cup r_U$                                     // select it

            $r_T$.ADDCHILD($r_U$)                                             // add it below $\kappa_T$
            $r_T.area \leftarrow r_T.area + r_U.area$                         // merge areas
            $\kappa_T \leftarrow$ UNION($\kappa_T, \kappa_U$)                 // and disjoint sets
            $\kappa_T.root \leftarrow r_T$
```

## 4 Experiments

We demonstrate the effectiveness of our W$\alpha$SH detector by measuring *repeatability* and *matching score* on the standard dataset and benchmark of [17], and by carrying out a retrieval experiment on *Oxford Buildings* [20], following the evaluation protocol that has been introduced in [21] and recently used to evaluate detectors in [26] and [1]. The performance of the W$\alpha$SH detector depends only on the *selection threshold* $\tau$. For all experiments we set $\tau = 100$, which has been set after subjective evaluation on the dataset of [17]. The W$\alpha$SH detector executable is available online[1].

### 4.1 Repeatability and Matching Score

We use the publicly available datasets of [17] to test our detector against the state-of-the-art. The sequences evaluate the effect of blur (*bikes*), scale/rotation

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

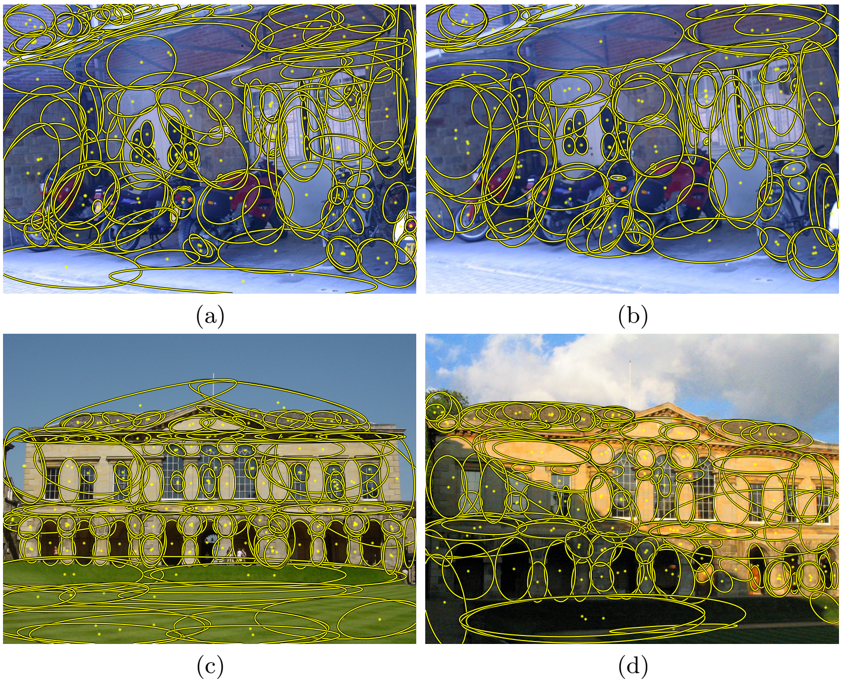

**Fig. 4.** Features detected by WαSH detector for a pair of images from (a)(b) the *bikes* sequence and (c)(d) *Oxford Buildings* (Worcester College)

(*boat*), as well as changes in lighting (*leuven*) and viewpoint (*wall*). Fig. 4ab show a detection example on the first two images of the *bike* sequence. There are not many overlapping detections, while most part of the image is covered.

Fig. 5 shows repeatability and matching score plots for all the detectors included in [17] with the addition of MFD [1] that has proved to have competitive performance to ours (see also Section 4.2). The WαSH detector performs among the best in all cases, while keeping the number of features considerably low; in particular, for the first image of each sequence, *bikes*: 409, *boat*: 365, *leuven*: 299 and *wall*: 782 features. It is quite invariant to scale and blur, which is attributed to the proposed selection criterion on the $\alpha$-filtration and—naturally—to the stability of image edges across scales. For the same reasons, the detector performs very well under illumination changes.

### 4.2 Large Scale Image Retrieval

In this section we test the performance of the WαSH detector on an image retrieval task using the *Oxford Buildings* dataset (an example is given in Fig. 4cd). Comparisons are performed against the Hessian-Affine, MSER, SIFT and SURF detectors, plus the recently introduced EF [26] and MFD [1] detectors, follow-

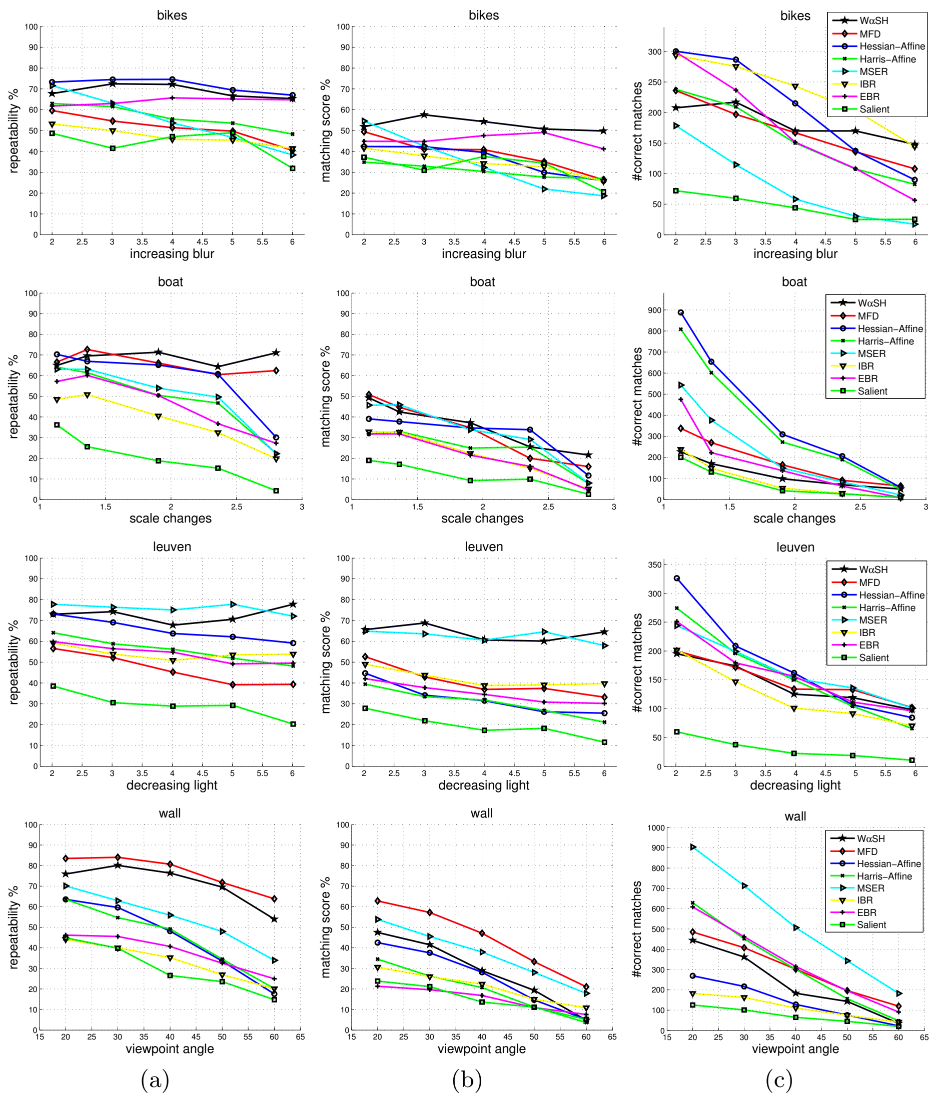

**Fig. 5.** (a) Repeatability, (b) matching score, and (c) correct matches. Scenes from top to bottom: bikes, boat, leuven, wall

ing their experimental framework. Features are extracted by the corresponding publicly available executables, using default parameters.

We extract SIFT descriptors for all detectors except SURF, for which we use the corresponding descriptor. We then build two visual vocabularies of different sizes for each detector, namely 50K and 200K, by clustering the descriptors using *approximate k-means* [20]. For each vocabulary, we carry out the retrieval experiment using the *bag-of-words* (BoW) model for representation, an *inverted*

**Table 1.** Results of the retrieval experiment *Oxford Buildings* for all detectors tested and for the 50K and 200K vocabularies. The number of features refers to the entire dataset, while detection time is average per image. Query times are average per query. mAP is measured on the same dataset used for vocabulary training.

| Detector | | **W$\alpha$SH** | MFD | EF | HessAff | MSER | SIFT | SURF |
|---|---|---|---|---|---|---|---|---|
| Features ($\times 10^6$) | | **7.19** | 7.64 | 19.72 | 29.02 | 13.33 | 11.13 | 6.84 |
| Detection time (s) | | **1.32** | 2.35 | 13.51 | 6.67 | 4.48 | 5.29 | 0.42 |
| Inverted file (MB) | 50K | **44.1** | 51.9 | 132.1 | 116.2 | 71.2 | 75.9 | 47.8 |
| | 200K | **49.1** | 58.4 | 146.2 | 128.8 | 78.8 | 84.0 | 53.5 |
| BoW query (ms) | 50K | **0.92** | 0.94 | 3.11 | 2.71 | 1.32 | 1.51 | 0.88 |
| | 200K | **0.75** | 0.68 | 1.81 | 1.61 | 0.88 | 0.95 | 0.64 |
| FastSM query (s) | 50K | **1.43** | 2.45 | 26.01 | 25.17 | 6.57 | 8.35 | 3.75 |
| | 200K | **1.35** | 0.93 | 4.69 | 6.10 | 2.20 | 5.29 | 3.45 |
| BoW (mAP) | 50K | **0.529** | 0.531 | 0.455 | 0.489 | 0.489 | 0.422 | 0.466 |
| | 200K | **0.592** | 0.600 | 0.528 | 0.578 | 0.568 | 0.494 | 0.575 |
| FastSM (mAP) | 50K | **0.541** | 0.540 | 0.500 | 0.516 | 0.524 | 0.446 | 0.497 |
| | 200K | **0.588** | 0.600 | 0.566 | 0.608 | 0.593 | 0.516 | 0.591 |

**Table 2.** mAP measurements for a similar retrieval experiment as in Table 1 for W $\alpha$SH detector against MFD, with a lower number of detected features, targeting $3 \times 10^6$ features for the entire dataset. This time 50K and 100K vocaburaries are built, to avoid overfitting.

| Detectors | | **W$\alpha$SH** | MFD |
|---|---|---|---|
| Features ($\times 10^6$) | | **3.03** | 2.59 |
| BoW (mAP) | 50K | **0.530** | 0.516 |
| | 100K | **0.543** | 0.534 |
| FastSM (mAP) | 50K | **0.524** | 0.517 |
| | 100K | **0.539** | 0.537 |

*file* for indexing, *tf-idf* weighting and *fast spatial matching* (FastSM) [20] for spatial verification. BoW histograms are matched using the histogram intersection following $\ell_1$ normalization, while for geometric re-ranking we set the minimum inliers to 7. The evaluation metric is *mean Average Precision* (mAP).

Table 1 summarizes the total number of features, the average detection time per image, average query time and mAP measurements for all detectors. The number of detected features used is critical, since it determines both the amount of memory used for the index and the query time, with FastSM being quadratic in the number of features. The performance of the W$\alpha$SH detector is at the state-of-the-art, despite using a low number of features (e.g. 1/4 of the Hessian-affine features), hence having a much lower memory footprint. The benefit in terms of query time is also considerable. Increasing the size of the vocabulary boosts the performance of all detectors up to a point.

The performance of MFD is similar to that of the W$\alpha$SH detector using approximately the same number of features, an observation that has led us to an additional experiment with the detectors tuned to produce a significantly smaller number of regions. Our aim is to test the ability of these detectors to go large scale. By reducing the number of features detected, we have to decrease the size of the vocabulary to prevent overfitting. In this setup, we create vocabularies of 50K and 100K visual words. The performance of the W$\alpha$SH detector is still high in all cases, especially with the 50K visual vocabulary, as shown in Table 2.

## 5 Discussion

We have introduced the W$\alpha$SH detector that relies on grouping edge samples using a shape stability measure on a weighted $\alpha$-filtration. Weighted $\alpha$-shapes appear particularly well-suited to model the topology of sampled points, with weights reflecting edge strength. Among our contributions are the use of the $\alpha$-filtration to represent cavities of boundary shape, the neighboring system of triangles and edges to associate connectedness to boundary strength, the use of the component tree to capture the topology of the $\alpha$-complex and the feature selection scheme resulting in stable features with minimal computational overhead.

The detected features include cases of salient regions that cannot always be handled by existing detectors, like non-extremal regions or regions that are not enclosed by complete boundaries, and regions determined by cavities of boundary shape. Yet, the detected feature set is typically small, resulting in significant gains in memory and speed in a large-scale image retrieval application without compromising performance. The detection itself is computationally efficient, exploiting regular triangulation and component trees. In the future, we will consider exploiting them for description and propose an integrated method for local features and descriptors.

## References

1. Avrithis, Y., Rapantzikos, K.: The medial feature detector: Stable regions from image boundaries. In: International Conference on Computer Vision, ICCV (2011)
2. Bay, H., Ess, A., Tuytelaars, T., Van Gool, L.: Speeded-up robust features (SURF). Computer Vision and Image Understanding (CVIU) 110, 346–359 (2008)
3. Beaudet, P.: Rotationally invariant image operators. In: International Joint Conference on Pattern Recognition (1978)
4. Canny, J.: A computational approach to edge detection. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) 8(6), 679–698 (1986)
5. Cazals, F., Giesen, J., Pauly, M., Zomorodian, A.: Conformal alpha shapes. In: Point-Based Graphics. Eurographics/IEEE VGTC Symposium Proc., (2005)
6. Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C.: Introduction to Algorithms, 3rd edn. MIT Press (2009)
7. Edelsbrunner, H.: Alpha shapes — a survey. In: Tessellations in the Sciences: Virtues, Techniques and Applications of Geometric Tilings. Springer (2010)

8. Edelsbrunner, H., Kirkpatrick, D., Seidel, R.: On the shape of a set of points in the plane. IEEE Transactions on Information Theory 29(4), 551–559 (1983)
9. Harris, C., Stephens, M.: A combined corner and edge detector. In: Alvey Vision Conference (1988)
10. Leutenegger, S., Chli, M., Siegwart, R.: BRISK: Binary Robust Invariant Scalable Keypoints. In: International Conference on Computer Vision (ICCV) (2011)
11. Lindeberg, T.: Feature detection with automatic scale selection. International Journal of Computer Vision (IJCV) 30(2), 79–116 (1998)
12. Lindeberg, T., Garding, J.: Shape-adapted smoothing in estimation of 3-D shape cues from affine deformations of local 2-D brightness structure. Image and Vision Computing 15(6), 415–434 (1997)
13. Lowe, D.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision (IJCV) 60(2), 91–110 (2004)
14. Matas, J., Chum, O., Urban, M., Pajdla, T.: Robust wide-baseline stereo from maximally stable extremal regions. Image and Vision Computing 22(10), 761–767 (2004)
15. Meine, H., Köthe, U., Stelldinger, P.: A topological sampling theorem for robust boundary reconstruction and image segmentation. Discrete Applied Mathematics 157(3), 524–541 (2009)
16. Mikolajczyk, K., Schmid, C.: An Affine Invariant Interest Point Detector. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) ECCV 2002, Part I. LNCS, vol. 2350, pp. 128–142. Springer, Heidelberg (2002)
17. Mikolajczyk, K., Tuytelaars, T., Schmid, C., Zisserman, A., Matas, J., Schaffalitzky, F., Kadir, T., Van Gool, L.: A comparison of affine region detectors. International Journal of Computer Vision (IJCV) 65(1), 43–72 (2005)
18. Mikolajczyk, K., Zisserman, A., Schmid, C.: Shape recognition with edge-based features. In: British Machine Vision Conference (BMVC) (2003)
19. Najman, L., Couprie, M.: Building the component tree in quasi-linear time. IEEE Transactions on Image Processing 15(11), 3531–3539 (2006)
20. Philbin, J., Chum, O., Isard, M., Sivic, J., Zisserman, A.: Object retrieval with large vocabularies and fast spatial matching. In: Computer Vision and Pattern Recognition, CVPR (2007)
21. Philbin, J., Chum, O., Sivic, J., Isard, M., Zisserman, A.: Lost in quantization: Improving particular object retrieval in large scale image databases. In: Computer Vision and Pattern Recognition, CVPR (2008)
22. Rapantzikos, K., Avrithis, Y., Kollias, S.: Detecting regions from single scale edges. In: Intern. Workshop on Sign, Gesture and Activity (SGA), European Conference on Computer Vision (ECCV) (September 2010)
23. Rosten, E., Drummond, T.W.: Machine Learning for High-Speed Corner Detection. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3951, pp. 430–443. Springer, Heidelberg (2006)
24. Teichmann, M., Capps, M.: Surface reconstruction with anisotropic density-scaled alpha shapes. In: IEEE Visualization (1998)
25. Tuytelaars, T., Mikolajczyk, K.: Local invariant feature detectors: A survey. Foundations and Trends in Computer Graphics and Vision 3(3), 177–280 (2008)
26. Zitnick, C., Ramnath, K.: Edge foci interest points. In: International Conference on Computer Vision (ICCV), pp. 359–366 (2011)
27. Zomorodian, A., Guibas, L., Koehl, P.: Geometric filtering of pairwise atomic interactions applied to the design of efficient statistical potentials. Computer-Aided Geometric Design 23(6), 531–544 (2006)