Cluster-based point set saliency

Flora Ponjou Tasse Jiri Kosinka Neil Dodgson
The Computer Laboratory, University of Cambridge, UK

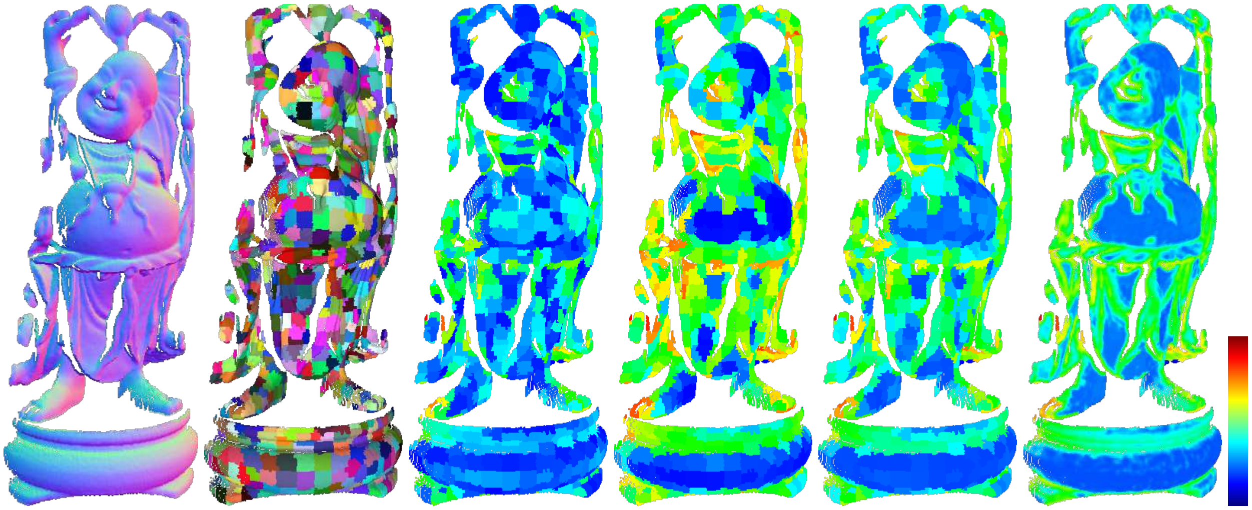


Figure 1: Cluster-based point set saliency on a range scan. From left to right: normal map, adaptive fuzzy clustering, cluster uniqueness, cluster spatial distribution, cluster saliency, and point saliency.

Abstract

We propose a cluster-based approach to point set saliency detection, a challenge since point sets lack topological information. A point set is first decomposed into small clusters, using fuzzy clustering. We evaluate cluster uniqueness and spatial distribution of each cluster and combine these values into a cluster saliency function. Finally, the probabilities of points belonging to each cluster are used to assign a saliency to each point. Our approach detects fine-scale salient features and uninteresting regions consistently have lower saliency values. We evaluate the proposed saliency model by testing our saliency-based keypoint detection against a 3D interest point detection benchmark. The evaluation shows that our method achieves a good balance between false positive and false negative error rates, without using any topological information.

1. Introduction

Mesh saliency, a measure of importance of points or regions on a 3D surface, has become a useful tool for several shape analysis tasks such as keypoint detection [2], shape similarity [9], mesh simplification [19], and viewpoint selection [20]. The human perceptual system is able to consistently and quickly estimate saliency for known and new shapes. But designing computational models that simulate this process remains a problem [15]. Perceptual research shows that contrast is an important factor in low-level visual saliency [7]. Thus research in both image and surface saliency is focused on finding better definitions of contrast. So far, surface-based methods have focused on contrast between vertices [19, 20, 27], whereas a new trend in image saliency detection uses region-based contrast [5, 25, 4] to produce significantly better results than local contrast. Inspired by these cluster-based approaches on images, we propose a new approach to point set saliency.

Due to the fact that most shape processing tasks have been aimed at polygonal meshes in the past, surface saliency has been mostly studied on such surface representation. These mesh-based methods depend on Gaussian smoothing, spectral properties [29], or shape descriptors [20], which all rely on topological information and cannot be trivially extended to point sets. Point sets are noisy, contain occlusions, and are often sampled at different densities depending on the position of the depth camera. This makes mesh-based methods more difficult to extend. Furthermore, 3D reconstruction without prior knowledge is still an active area of research. Few papers handle saliency of point sets [24, 21, 27]. Considering the recent increase in cheap depth camera availability and thus the proliferation of 3D point clouds, it is now becoming ever more important to process such point clouds. Our proposed saliency algorithm does not use topological information, and thus supports a wider range of 3D surface representations, including polygon soups and 3D range scans. We show that even though we do not use topological data, our saliency detection still performs better compared to mesh-based methods.

Our work is inspired by recent methods in image saliency detection [25, 4]. These methods cluster pixels into perceptually homogeneous regions, compute a per-region contrast from cluster uniqueness and spatial distribution, and use this to assign a saliency value to each cluster and then each pixel.

Our main contributions are:

1. An adaptive fuzzy clustering algorithm that not only assigns points to clusters but also computes the probability that a point belongs to a specific cluster (Section 4).

2. A novel method for computing cluster and point saliency from a set of fuzzy clusters (Section 5).

3. Application of our proposed saliency detection to keypoint extraction (Section 7).

2. Related work

Saliency attempts to emulate the bottom-up attentional mechanism in human vision [8]. Howlett et al. [14] show that mesh saliency exists and can be useful in mesh simplification. Their study uses eye fixations to simplify a set of meshes. Their results demonstrate that mesh simplification using both quadric error metrics [10] and eye fixation data produces meshes that are perceived as better representatives of the original, compared to only using quadric error metrics.

Mesh saliency methods are typically based on contrast-based saliency techniques for images [19, 20, 34, 29]. For example, multi-scale image saliency [16] has inspired several 3D saliency methods based on the Difference of Gaussian scale space [24, 19, 17, 21, 2].

To achieve stable saliency models, robust to noise, several techniques use local shape descriptors to compute contrast. Gelfand et al. [11] compute the Integral Volume Descriptor for each point on a mesh, and saliency of a point is based on the uniqueness of its associated descriptor. Gal and Cohen-Or [9] approximate a mesh by a sparse set of local patches represented by descriptors. Saliency regions are grown around the local patches by incrementally adding neighbouring patches that maximize a saliency grade. The mesh saliency model of Leifman et al. [20] is inspired by context-aware saliency detection in images [13]. Salient regions are those that are distinct both locally and globally, and close to foci of attention. Shtrom et al. [27] extend this to point clouds. Song et al. [29] propose a spectral-based saliency model that produces results comparable to saliency probabilistic models [3]. Saliency is computed based on properties of the log-Laplacian spectrum of a mesh.

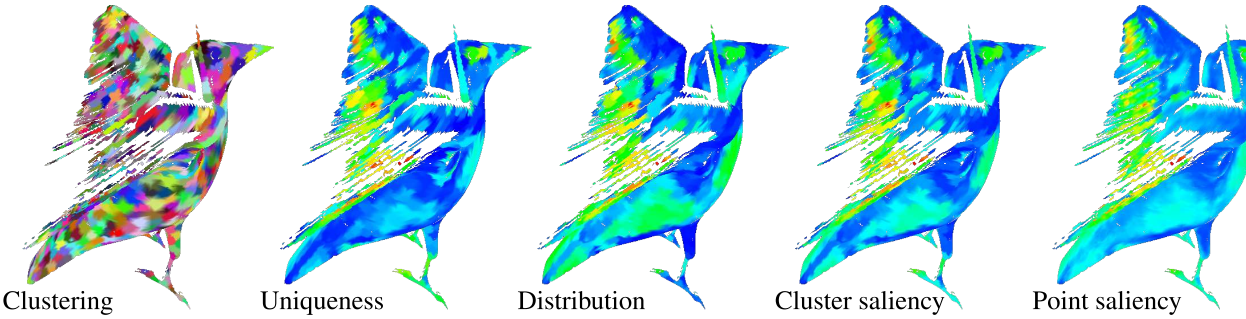
Relatively few saliency models support point clouds [24, 21, 32, 27]. The approach of Shtrom et al. [27] is the most robust point set saliency detection. However, it computes each point's distinctiveness by comparing the point's local descriptor to all other descriptors close enough in the descriptor space. This is an expensive process. Region-based approaches in images have become increasingly popular due to their efficiency and abstraction of unnecessary details [5, 25, 4]. They typically perform 3 steps: an over-segmentation of the image, a per-cluster saliency detection, and a propagation of the cluster saliency to obtain per-pixel saliency. Inspired by global contrast-based saliency in images [5], Wu et al. [34] use a multi-scale descriptor based on Zernike coefficients to compute patch-based local contrast and global point rarity on a mesh. Tao et al. [31] use the same descriptor to generate a set of uninteresting patches. Using these patches as queries, patch saliency is obtained by ranking all patches based on their relevance to the queries while respecting the manifold structure of the descriptor space of patches. Both of these methods [34, 31] produce stable saliency maps but they only support meshes and the descriptor computation is inefficient. Our work is inspired by such cluster-based methods and we investigate whether similar ideas can be applied to point sets.

3. Overview of our method

We propose a method for computing point set saliency that does not require any topological information. Our method is inspired by cluster-based saliency detection in images [25, 4]. We first cluster a given point set into similar regions, compute a saliency value per cluster, and then propagate these saliency values to points (Figure 1).

Adaptive fuzzy clustering (Section 4) Our fuzzy clustering consists of two steps: hard adaptive clustering (Section 4.1) and computation of point-cluster probabilities (Section 4.2). The second step ensures that in the saliency propagation stage, the saliency of a point p is most influenced by those clusters p most likely belongs to.

In the hard adaptive clustering step, we assign each point to a cluster based on the point's closeness to other points both in the Euclidean space and a descriptor space. Two points or cluster centres are close spatially if their Euclidean distance is small, relative to that of other pairs. Similarly, two points are geometrically close if their distance in the descriptor space is small. This is formalized in Section 4.



Clustering Uniqueness Distribution Cluster saliency Point saliency

Figure 2: An overview of our saliency model. The Bird partial scan contains holes and has variable point density. The saliency results show that our method is robust enough to detect salient regions such as the eyes, mouth, feet and wings.

Both Euclidean and descriptor spaces are used in the second step for computing probabilities that a point p belongs a cluster, by assuming each cluster is a combination of two Gaussian kernels defined over the Euclidean and descriptor spaces. In practice, computing these probabilities for all points and clusters will be inefficient. Instead, we construct an adjacency graph of clusters, where spatially closed regions are connected by an edge. This helps optimize our solution, since for any given point p allocated to a cluster C_p, we only need to consider clusters within a certain range of C_p for its probability distribution.

Our adaptive fuzzy clustering approach to saliency detection is in contrast to other saliency models [34, 31] that oversegment a shape into mutually exclusive clusters based on spatial proximity.

Cluster-based saliency (Section 5) From the set of clusters described above, we compute cluster uniqueness and distribution. Cluster uniqueness is a measure of how distinctive a cluster is, computed by comparing a cluster descriptor to every other cluster. Uniqueness is heavily influenced by nearby clusters, so that a region is more distinctive if it is different from its surroundings. On the other hand, cluster spatial distribution computes the spatial variance of geometrically similar clusters. This means that a cluster is widely distributed if similar clusters are spread across the point set. Salient regions are those that are most unique, with a small spatial distribution. Based on this premise, uniqueness and spatial distribution are integrated into a single saliency value per cluster.

Point-level saliency Each point p is finally assigned a saliency value which is a linear combination of saliency from all clusters, weighted by the probability that p belongs to a cluster. Figures 1 and 2 illustrate the results of clustering, uniqueness, spatial distribution, cluster saliency, and point saliency on point sets.

4. Adaptive fuzzy clustering

We discuss the local feature descriptor used to characterize points and present our adaptive clustering algorithm.

Having considered several alternatives, we chose to use the modified version [27] of Fast Point Feature Histogram (FPFH) as our local shape descriptor. It is a fast and robust local shape descriptor introduced by Rusu et al. [26]. The FPFH of a point p describes the angles between normals in the k-neighbourhood of p. This means that we need to have captured or calculated normals for each point. FPFHs computation complexity is $O(kN)$, where N is the size of the point cloud, which makes FPFH one of the fastest local shape descriptors. Shtrom et al. [27] propose an adaptation of FPFH that is invariant to reflections. To compute FPFH distances, we use the Chi-square distance χ^2.

4.1. Adaptive clustering

To decompose a point cloud into mutually exclusive clusters, we propose a 3D adaptation of Adaptive-SLIC (Simple Linear Iterative Clustering) superpixels [1]. Adaptive-SLIC is an efficient K-means clustering algorithm that clusters pixels both in the Euclidean and colour space, generating compact, nearly uniform superpixels. It takes as parameters the number of clusters and adaptively updates a compactness parameter in each iteration of the K-means clustering. Adaptive-SLIC has linear complexity. Our 3D adaptation runs K-means in both the Euclidean and the FPFH descriptor space. The combined space $\mathcal{F}$ is a 36 dimensional space, whose points are represented by

$$f = [x, y, z, h_1, ..., h_{33}], \tag{1}$$

where (x, y, z) is the position and $(h_1, ..., h_{33})$ the FPFH descriptor of f. In the rest of the paper, $X(f)$ refers to the spatial position of f and $H(f)$ refers to its descriptor.

Given a set of points in the combined space, the adaptive clustering is computed as follows:

Cluster mean initialization Following the method of Papon et al. [23], we create a 3D voxel grid over the input points using an octree. For each non-empty voxel, we add the mean of all the points within it to a list of initial cluster means. The voxel grid generation is based on seed resolution parameter R that we derive from a given number of desired clusters $\hat{K}$. R is the average diameter of spheres containing $N/\hat{K}$ points, where N is the number of points. This definition of R produces voxels that have $N/\hat{K}$ points on average. Note that due to variable sampling in point sets, the number of initial clusters K generated from the voxel grid often differs from the parameter $\hat{K}$.

K-means clustering Given a list of initial cluster means, each iteration of K-means assigns every point to its nearest cluster and then updates cluster means based on the new assignments. The key idea behind the linear time complexity of SLIC is that for any point, we only check its distance to cluster means that are within Euclidean distance $2R$. To find the nearest cluster to a point, we use the following distance function:

$$D(f,g) = \frac{\|X(f)-X(g)\|_2}{M_X} + \frac{D_{\chi^2}\big(H(f),H(g)\big)}{M_H}, \quad (2)$$

where $f, g \in \mathcal{F}$, M_X is the maximum spatial distance within a cluster, and M_H is the maximum descriptor distance within a cluster in the previous iteration.

To speed up the adaptive clustering, we simplify the point set prior to the K-means initialization and clustering described above. We follow a similar approach to Papon et al. [23] by first partitioning the 3D space into small voxels. The voxel resolution used here is equal to the average Euclidean distance between a point and its k neighbours. This creates a new point set consisting of means of non-empty voxel cells. This simplified point set is the input to the K-means clustering described above. The cluster assigned to a point p is equal to the cluster of its voxel centre.

Our adaptation of Adaptive-SLIC to 3D produces clusters of points that are spatially and geometrically similar. It produces less variation in the point descriptors within each cluster and thus the average descriptor of a cluster, used throughout our algorithm, is a better representative of the points in it. Our saliency detection is therefore more stable and better captures interesting regions, compared to a simple voxel-based spatial clustering, especially with small $\hat{K}$. This is illustrated in Figure 3.

4.2. Point-cluster probability

First, we introduce the notion of cluster adjacency. This is used to speed up the computation of point-cluster probability. We construct a cluster adjacency graph where nodes are clusters and edges connect any two clusters with neighbouring points. More precisely, two clusters are neighbours if there exists one point from each such that the Euclidean distance between the two points is within 1% of the point set bounding sphere radius. Each cluster is represented as a point in our combined space $\mathcal{F}$ by the mean of all the points within that cluster.

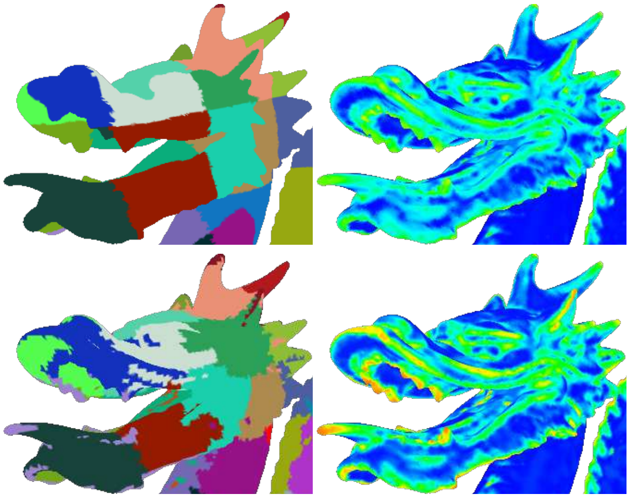


Figure 3: Compare saliency results based on a voxel-based spatial clustering (top) and our adaptive clustering (bottom). The number of desired clusters $\hat{K} = 100$.

Point-cluster probability is represented by a sparse matrix P such that the probability that a point p_i belongs to cluster c_j is $P_{i,j}$. P is computed with the help of another sparse matrix W of the same size $N \times K$ with elements

$$W_{i,j} = \begin{cases} \frac{\exp\left(-w^X_{i,j} - \lambda_H w^H_{i,j}\right)}{\sigma_X \sigma_H} & p_i \in \mathcal{N}_j; \\ 0 & \text{otherwise,} \end{cases} \quad (3)$$

with

$$w^X_{i,j} = \|X(p_i) - X(c_j)\|_2,$$
$$w^H_{i,j} = D_{\chi^2}\big(H(p_i), H(c_j)\big),$$

where $\mathcal{N}_j$ is the set of points in clusters within a 3-ring neighbourhood of c_j in the adjacency graph, σ_X is the average Euclidean distance to the spatial mean of the point set, σ_H is the average χ^2 distance to the global descriptor mean, and λ_H is a parameter that determines the importance of geometric distance over spatial distance. For all results shown in this paper, we used $\lambda_H = 5$.

Next, columns of W, each representing a cluster, are normalized:

$$W'_{i,j} = \frac{W_{i,j}}{\sum_{k=1}^{N} W_{k,j}}.$$

Finally, each row is normalized to produce the point-cluster probability matrix given by

$$P_{i,j} = \frac{W'_{i,j}}{\sum_{k=1}^{K} W'_{i,k}}. \quad (4)$$

5. Saliency detection

This section describes the computation of cluster uniqueness and spatial distribution. Both cues can be seen as weighted sums of descriptor distances and Euclidean distances, respectively. The weight of a cluster is

$$w_j = \frac{\sum_{i=1}^{N} W_{i,j}}{\sum_{k=1}^{K} \sum_{i=1}^{N} W_{i,k}}. \quad (5)$$

Uniqueness This measures the local distinctiveness of a point. It is based on the same premise used in almost all saliency models, i.e., regions that differ from their surroundings are more interesting. Shtrom et al. [27] compute the low-level distinctiveness D_i of a point p_i as

$$D_i = 1 - \exp\left(-\frac{1}{N}\sum_{k=1}^{N} \frac{D_{\chi^2}\big(H(p_k), H(p_i)\big)}{1 + \|X(p_k) - X(p_i)\|_2}\right). \quad (6)$$

In our case, we are computing cluster-level uniqueness instead, and thus the variability in clusters should be taken into account. This is achieved by adding cluster weight and a normalization term to Equation 6. Thus, the uniqueness U_j of a cluster c_j is a weighted sum of descriptor distances to every cluster defined as

$$U_j = 1 - \exp\left(-\frac{1}{Z_j}\sum_{k=1}^{K} w_k d_{k,j} D_{\chi^2}\big(H(c_k), H(c_j)\big)\right)$$

with the same weight based on spatial distance

$$d_{k,j} = 1/\big(1 + \|X(c_k) - X(c_j)\|_2\big)$$

and the normalizer $Z_j = \sum_{k=1}^{K} d_{k,j}$.

Spatial distribution The local uniqueness of a cluster is often not sufficient to emulate the human low-level visual attention. Regions that are similar but spread over a large area are less salient than similar regions that are more compact. Uninteresting points tend to be widely distributed, with a high spatial variance, whereas more interesting regions are more compact. For a given cluster with index j, we first compute the spatial mean

$$\mu_j = \frac{1}{Z'_j}\sum_{k=1}^{K} d'_{k,j} X(c_k)$$

of geometrically similar regions with a weight based on geometric distance

$$d'_{k,j} = \exp\left(-\frac{D_{\chi^2}\big(H(c_k), H(c_j)\big)}{\sigma_H}\right)$$

and the normalizer $Z'_j = \sum_{k=1}^{K} d'_{k,j}$ with σ_H as above.

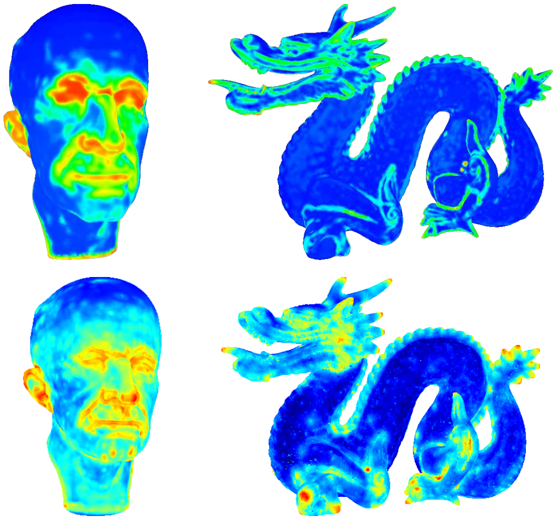


Figure 4: Comparison of point set saliency. Top: Our method. Bottom: Shtrom et al.'s method [27]. On the Max Planck model, our saliency is concentrated around facial features, with values close to zero elsewhere, in contrast with Shtrom et al. result where only a small area at the top of the head has low saliency. We also note on the Dragon model that our saliency result is less noisy and emphasizes fine-scale features such as the outline of the dragon mouth.

The cluster spatial distribution is then the weighted sum of Euclidean distances to the mean μ, defined as

$$V_j = \exp\left(-\frac{1}{Z'_j}\sum_{k=1}^{K} w_k d'_{k,j} \|X(c_k) - \mu_j\|_2\right).$$

Cluster saliency We combine the saliency map obtained from the two cues above to construct a cluster saliency map $S_j = U_j + \lambda V_j$, where λ is a parameter between 0 and 1 that determines the relative importance of spatial distribution. In this paper, we used $\lambda = 1.0$ for all our examples.

Point-level saliency Finally, the saliency of a point with index i is given by $s_i = \sum_{j=1}^{K} P_{i,j} S_j$.

6. Results

We now compare our method against previous point-based and mesh-based saliency methods, followed by a discussion on the influence of the parameter $\hat{K}$.

Comparison with point-based saliency method In Figure 4, we compare results generated by our algorithm against state-of-the-art in point set saliency [27]. Note how our saliency is concentrated in interesting regions such as the eyes on the Max Planck model. We are also able to detect fine-scale salient features on the head of the Dragon.

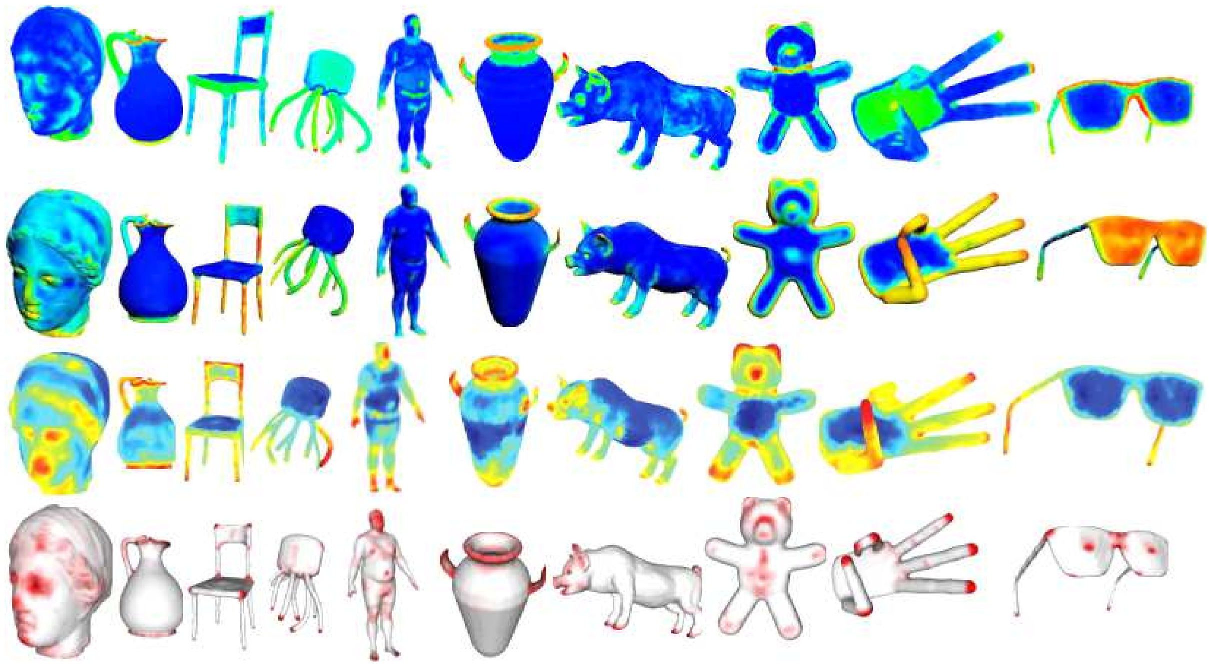


Figure 5: From the top row down: Comparison of our method with Tao et al. [31], Song et al. [29], and ground-truth [4]. Tao et al. [31] fail to capture salient regions on the human and glasses models, and in general, do not match high saliency regions in the ground-truth. Song et al. [29] saliency model tends to assign different saliency values to similar regions such as the bodies of the vases and arms of the teddy shape. Only our method succeeds in capturing the salient ends of the tentacles on the octopus model. Overall, our saliency values closely match ground-truth, especially on the glasses model.

Comparison with mesh-based saliency method In Figure 5, we compare against the mesh-based methods by Song et al. [29], Tao et al. [31], as well as the ground truth data gathered by Chen et al. [3]. The only mesh-based method that consistently produces correct saliency maps is spectral mesh saliency [29]; it detects salient regions correctly but similar unsalient regions, like the main bodies of the vases, often have different saliency values. Note that our method may assign high saliency to a flat region (e.g. the hand palm) when another region is close to it as these regions may be clustered together. This is because we use Euclidean space for clustering, which may in such cases substantially change the cluster descriptor from that of a flat surface.

Influence of the number of clusters Figure 6 shows how the number of desired clusters $\hat{K}$ affects saliency detection. Our method is robust enough to detect salient regions with very small $\hat{K}$ (as small as $\hat{K} = 10$). A larger $\hat{K}$ increases the quality of the saliency. After a certain point, increasing the number of desired clusters, and thus the number of generated clusters K, does not noticeably change the saliency. The smallest value of K beyond which more clusters do not improve the saliency depends on the input point cloud.

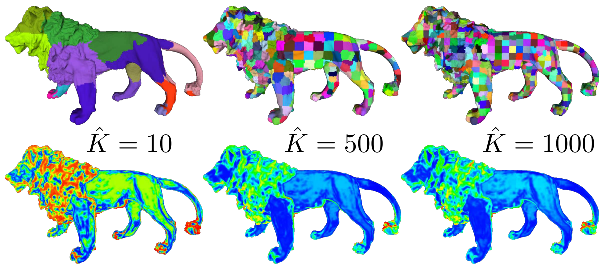

$\hat{K} = 10$ $\hat{K} = 500$ $\hat{K} = 1000$

Figure 6: Influence of the number of desired clusters.

Computational cost With N the size of the point cloud, FPFH's computation complexity is $O(mN)$, where m is the number of neighbours per point. Adaptive fuzzy clustering has complexity $O((N + K)\log N)$, where K is the number of clusters. Saliency computation has complexity $O(N + K^2)$. All the results in this paper, unless specified otherwise, were generated with $K \approx 500$, by setting the number of desired clusters $\hat{K} = 500$. Table 1 presents the computation times for some results presented in this paper.

For small K, the main bottleneck of our algorithm is the adaptive clustering. However, our saliency detection is robust enough that our adaptation of SLIC can be replaced with a faster clustering algorithm such as voxelization, with an expected small quality loss.

Model	Size	FPFHs	Clustering	Total
Bird scan	12K	0.20s	0.13s	0.84s
Max Planck	49K	1s	2s	4s
Buddha scan	78K	1s	2s	4s
Lion	183K	3s	7s	11s
Dragon	437K	2s	11s	13s

Table 1: Computation times of a single-threaded implementation of our saliency detection, on an Intel Core i7 CPU with 2.5 Ghz and 16GB RAM. Computing cluster and point saliency took less than 1 second (included in the total time) for all the models listed.

Compared to the state-of-the-art in point set saliency, our method is faster. Shtrom et al. [27] saliency time complexity is $O(kN \log N)$, where k is the neighbourhood size used. In their results, k is proportional to 10% of the maximum descriptor distance, and thus can grow very quickly for large point clouds. They detect saliency on the Igea model ($134K$ points) in 2 minutes. Our method took less than 15 seconds for each model illustrated in this paper. An example of our saliency detection on a scan of the Bremen city centre (12M points) is available in the supplementary material.

7. Evaluation of our keypoint detection

We evaluate our method by testing our saliency-based keypoint detection against a 3D interest point detection benchmark [6]. The benchmark consists of human-marked interest points on a total of 43 triangular meshes. Dutagaci et al. [6] evaluate six keypoint detection algorithms against this data: local maxima computed from Mesh Saliency [19], salient points based on a combination of mesh saliency with statistical descriptors [2], scale-dependent corners [22], 3D-SIFT [12], 3D-Harris [28], and a multi-scale detector based on the Heat Kernel Signature (HKS) [30]. The results of their evaluation show that overall, the Mesh Saliency method has lower false negative error rates than all other methods, but has high false positive rates (these rates are defined below). On the other hand, HKS-based keypoint detection finds very few keypoints, most of which are correct, and thus has a very low false positive error rate compared to others, with high false negative error rates. Shtrom et al. [27] also tested their point set saliency detection model against this benchmark by selecting local maxima above a certain saliency threshold as keypoints. Their results show that they achieve slightly lower false negative error rates than the competing methods, with false positive rates comparable to other methods except HKS.

Following Dutagaci et al. [6] and Shtrom et al. [27], our keypoints are local maxima over a saliency threshold given by the average saliency over all local maxima. Figure 7 illustrates a comparison of keypoint detection.

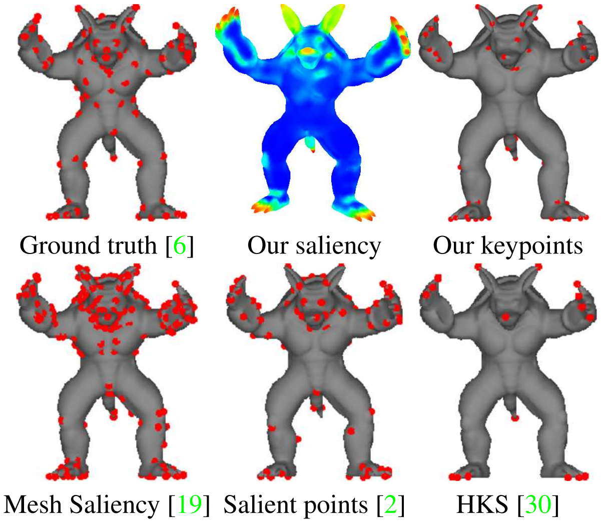

Ground truth [6]
Our saliency
Our keypoints
Mesh Saliency [19]
Salient points [2]
HKS [30]

Figure 7: Interest point detection: keypoints detected by our algorithm compared to other methods. The ground truth displayed is based on $\sigma = 0.03$, $n = 2$.

Error rate computation Given a parameter σ representing the radius of an interest region and d_M the diameter of a model M, interest points whose geodesic distance to each other is less than $2\sigma d_M$ are grouped together. Ground truth is determined by σ and an additional parameter n representing the minimum number of human participants. The set of ground truth points $G_M(n, \sigma)$ of a mesh M consists of all interest points agreed upon by at least n distinct users. Given a point $p \in G$, its geodesic neighbourhood C_r is

$$C_r(p) = \{q \in M \mid d_g(p, q) \leq r\},$$

where $d_g(p, q)$ is the geodesic distance between p and q, and r represents the radius tolerance factor. Let D be the set of keypoints detected by an algorithm. A point $p \in G_M(n, \sigma)$ is correctly detected if there exists $a \in D \cap C_r(p)$ which is not closer to any other ground truth point. The False Negative Error rate is computed as $E_{FN}(r) = 1 - N_C/N_G$, where N_G is the number of ground truth points and N_C is the number of correctly detected ground truth points.

All points in D that do not have a corresponding ground truth point are false positives. The False Positive Error rate is computed as $E_{FP}(r) = 1 - N_C/N_D$, where N_D is the number of keypoints detected by the evaluated algorithm.

To take into account the popularity of individual ground truth points, the Weighted Miss Error is defined as

$$E_{WM}(r) = 1 - \frac{1}{\sum_{i=1}^{N_G} n_i} \sum_{i=1}^{N_G} n_i \delta_i \tag{7}$$

with

$$\delta_i = \begin{cases} 1 & \text{if } g_i \text{ is detected;} \\ 0 & \text{otherwise,} \end{cases}$$

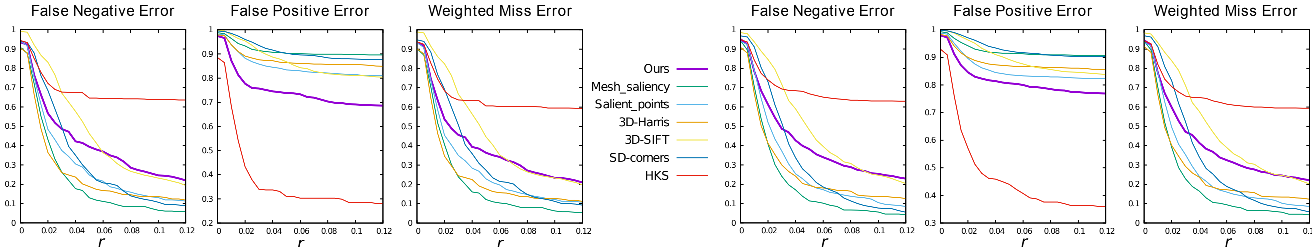


Figure 8: Performance on two datasets. Left triple: Dataset A (24 models, 23 Subjects), $\sigma = 0.03$, $n = 11$. Right triple: Dataset B (43 models, 16 Subjects), $\sigma = 0.03$, $n = 8$.

where g_i are ground truth points and n_i the number of subjects that selected an interest point in $C_r(g_i)$.

Benchmark results The benchmark collected data is split into two sets A and B [6]. Dataset A contains interest points on 24 meshes marked by 23 subjects. Dataset B consists of interest points on all 43 models that have been marked by at least 16 participants. Figure 8 shows the behaviour of the error rates E_{FN}, E_{FP}, and E_{WM} of our algorithm as the radius of tolerance r increases, compared to other methods.

The results show that our method has consistently lower E_{FP} rates compared against all other methods, except HKS. This is at the expense of E_{FN} and E_{WM} rates that are higher than most other methods but still comparable with 3D-SIFT and considerably better than HKS. If we compare our method against Mesh Saliency, ours has a higher E_{FP} rate, and against HKS that detects few points and thus has a high E_{FN} rate, our method seems to strike a reasonable balance between false positives and false negatives. A good balance is particularly relevant in applications such as feature-based shape retrieval [18]. In feature-based shape retrieval, shapes are typically compared based on keypoints detected on these shapes, thus it is important to extract interesting keypoints able to differentiate between classes.

To quantify this balance between false positive and false negative rates, we compute for each detection method

$$F_\beta = (1+\beta^2)\frac{PR}{\beta^2 P + R},$$

where $P = 1 - E_{FP}$ is precision and $R = 1 - E_{FN}$ is recall. F_β is interpreted as a measure of effectiveness to a user who attaches β times as much importance to recall as precision [33]. Table 2 shows that our method achieves the best F_β score when recall is weighted twice as much as precision, and is second best after HKS otherwise. This means that our algorithm performs best when emphasis is put on detecting as many ground-truth points as possible, at the expense of detecting uninteresting points. This emphasis is helpful in applications like shape retrieval, where uninteresting points occur more often than interesting features and thus are given less weight when computing similarities.

	Dataset A			Dataset B		
	$F_{0.5}$	F_1	F_2	$F_{0.5}$	F_1	F_2
Ours	0.27	0.32	**0.42**	0.20	0.26	**0.37**
Salient Points	0.16	0.23	0.37	0.16	0.22	0.36
3D-Harris	0.14	0.21	0.37	0.14	0.20	0.35
HKS	**0.54**	**0.43**	0.35	**0.45**	**0.38**	0.33
MeshSaliency	0.10	0.15	0.29	0.09	0.13	0.27
3D-SIFT	0.10	0.13	0.21	0.07	0.10	0.17
SD-corners	0.07	0.11	0.21	0.05	0.08	0.17

Table 2: F_β scores of keypoint detection for $\beta = 0.5, 1, 2$. We used $\sigma = 0.03$ and tolerance radius $r = 0.03$ to compute precision and recall.

Our algorithm is able to achieve the above competitive results without using any mesh connectivity information during saliency detection or key point extraction. This makes our algorithm more robust to topological inconsistencies and more flexible than mesh-based methods.

8. Conclusion

Our novel cluster-based method for detecting point set saliency does not use topology information and thus supports a wide range of 3D shape representations, including polygon soups and range scan data. Our qualitative results show that our saliency model detects fine-scale salient features better than other state-of-the-art point-based and mesh-based methods. We show that region-based contrast [5, 25, 4] can be successfully applied to point clouds, giving results (Figure 4) comparable with other point cloud saliency methods. We apply our saliency model to keypoint detection and show that it has significantly lower false positive error rates than previous work, with the exception of keypoints based on the Heat Kernel Signature, which, on the other hand, has the highest false negative error rates. We also show that our keypoint detection has the best effectiveness score F_β when β is in the high recall range.

Tasse was supported by a Google European Doctoral Fellowship and an IDB Cambridge International Scholarship. Kosinka was supported by the Engineering and Physical Sciences Research Council [EP/H030115/1].

References

[1] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Susstrunk. SLIC superpixels compared to state-of-the-art superpixel methods. *IEEE Trans. Pattern Anal. Mach. Intell.*, 34(11):2274–2282, Nov. 2012. 3

[2] U. Castellani, M. Cristani, S. Fantoni, and V. Murino. Sparse points matching by combining 3D mesh saliency with statistical descriptors. *Computer Graphics Forum*, 27(2):643–652, 2008. 1, 2, 7

[3] X. Chen, A. Saparov, B. Pang, and T. Funkhouser. Schelling points on 3D surface meshes. *ACM Trans. Graph.*, 31(4):29:1–29:12, July 2012. 2, 6

[4] M.-M. Cheng, J. Warrell, W.-Y. Lin, S. Zheng, V. Vineet, and N. Crook. Efficient salient region detection with soft image abstraction. In *ICCV 2013*, pages 1529–1536, December 2013. 1, 2, 6, 8

[5] M.-M. Cheng, G.-X. Zhang, N. J. Mitra, X. Huang, and S.-M. Hu. Global contrast based salient region detection. In *CVPR 2011*, pages 409–416, 2011. 1, 2, 8

[6] H. Dutagaci, C. P. Cheung, and A. Godil. Evaluation of 3D interest point detection techniques via human-generated ground truth. *Vis. Comput.*, 28(9):901–917, Sept. 2012. 7, 8

[7] W. Einhauser and P. Konig. Does luminance-contrast contribute to a saliency map for overt visual attention? *European Journal of Neuroscience*, 17(5):1089–1097, March 2003. 1

[8] S. Frintrop, E. Rome, and H. I. Christensen. Computational visual attention systems and their cognitive foundations: A survey. *ACM Trans. Appl. Percept.*, 7(1):6:1–6:39, Jan. 2010. 2

[9] R. Gal and D. Cohen-Or. Salient geometric features for partial shape matching and similarity. *ACM Trans. Graph.*, 25(1):130–150, Jan. 2006. 1, 2

[10] M. Garland and P. S. Heckbert. Surface simplification using quadric error metrics. In *SIGGRAPH 1997*, pages 209–216, 1997. 2

[11] N. Gelfand, N. J. Mitra, L. J. Guibas, and H. Pottmann. Robust global registration. In *SGP 2005*, pages 197–206, 2005. 2

[12] A. Godil and A. I. Wagan. Salient local 3D features for 3D shape retrieval. *Proc. SPIE*, 7864:93–102, 2011. 7

[13] S. Goferman, L. Zelnik-Manor, and A. Tal. Context-aware saliency detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 34(10):1915–1926, Oct. 2012. 2

[14] S. Howlett, J. Hamill, and C. O'Sullivan. Predicting and evaluating saliency for simplified polygonal models. *ACM Trans. Appl. Percept.*, 2(3):286–308, July 2005. 2

[15] L. Itti and C. Koch. Computational modelling of visual attention. *Nature Reviews Neuroscience*, 2(3):194–203, Mar 2001. 1

[16] L. Itti, C. Koch, and E. Niebur. A model of saliency-based visual attention for rapid scene analysis. *IEEE Trans. Pattern Anal. Mach. Intell.*, 20(11):1254–1259, Nov. 1998. 2

[17] Y. Kim, A. Varshney, D. W. Jacobs, and F. Guimbretière. Mesh saliency and human eye fixations. *ACM Trans. Appl. Percept.*, 7(2):12:1–12:13, Feb. 2010. 2

[18] G. Lavoué. Combination of bag-of-words descriptors for robust partial shape retrieval. *The Visual Computer*, 28(9):931–942, 2012. 8

[19] C. H. Lee, A. Varshney, and D. W. Jacobs. Mesh saliency. *ACM Trans. Graph.*, 24(3):659–666, July 2005. 1, 2, 7

[20] G. Leifman, E. Shtrom, and A. Tal. Surface regions of interest for viewpoint selection. In *CVPR 2012*, pages 414–421, 2012. 1, 2

[21] X. Li and I. Guskov. Multi-scale features for approximate alignment of point-based surfaces. In *SGP 2005*, pages 217–226, 2005. 2

[22] J. Novatnack and K. Nishino. Scale-dependent 3D geometric features. In *ICCV 2007*, pages 1–8, Oct 2007. 7

[23] J. Papon, A. Abramov, M. Schoeler, and F. Wrgtter. Voxel cloud connectivity segmentation – supervoxels for point clouds. In *CVPR 2013*, pages 2027–2034, June 2013. 4

[24] M. Pauly, R. Keiser, and M. H. Gross. Multi-scale feature extraction on point-sampled surfaces. *Comput. Graph. Forum*, 22(3):281–290, 2003. 2

[25] F. Perazzi, P. Krähenbühl, Y. Pritch, and A. Hornung. Saliency filters: Contrast based filtering for salient region detection. In *CVPR 2012*, pages 733–740, 2012. 1, 2, 8

[26] R. B. Rusu. *Semantic 3D Object Maps for Everyday Manipulation in Human Living Environments*. PhD thesis, Computer Science department, Technische Universitaet Muenchen, Germany, October 2009. 3

[27] E. Shtrom, G. Leifman, and A. Tal. Saliency detection in large point sets. In *ICCV 2013*, pages 3591–3598, 2013. 1, 2, 3, 5, 7

[28] I. Sipiran and B. Bustos. A robust 3D interest points detector based on Harris operator. In *Eurographics Workshop on 3D Object Retrieval*, pages 7–14, 2010. 7

[29] R. Song, Y. Liu, R. R. Martin, and P. L. Rosin. Mesh saliency via spectral processing. *ACM Trans. Graph.*, 33(1):6:1–6:17, Feb. 2014. 1, 2, 6

[30] J. Sun, M. Ovsjanikov, and L. Guibas. A concise and provably informative multi-scale signature based on heat diffusion. In *SGP 2009*, pages 1383–1392, 2009. 7

[31] P. Tao, J. Cao, S. Li, L. Liu, and X. Liu. Mesh saliency via ranking unsalient patches in a descriptor space. *Computers and Graphics*, 2014. 2, 3, 6

[32] R. Unnikrishnan and M. Hebert. Multi-scale interest regions from unorganized point clouds. In *Workshop on Search in 3D (S3D)*, pages 1–8, June 2008. 2

[33] C. J. van Rijsbergen. *Information Retrieval*. Butterworth-Heinemann, Newton, MA, USA, 2nd edition, 1979. 8

[34] J. Wu, X. Shen, W. Zhu, and L. Liu. Mesh saliency with global rarity. *Graphical Models*, 75(5):255 – 264, 2013. 2, 3