Finding Trees in Mountains – Outlier Detection on Polygonal Chains

Michael Singhof, Daniel Braun, and Stefan Conrad

Heinrich-Heine-Universität Düsseldorf, Institut für Informatik,
Universitätsstr. 1, 40225 Düsseldorf, Germany
{singhof,braun,conrad}@cs.uni-duesseldorf.de

Abstract. In this work, we present an approach to the detection of outliers in certain polygonal chains. These originate from images of mountains which are first segmented in order to extract the mountains silhouette. In general, the aim of our framework is to recognise the mountain in the image in order to overcome the problem of large amounts of images on the internet that are not tagged and thus cannot be searched in a sensible fashion.

The appearance of outliers in our case is specific by them either being obstacles in the image that are in front of the mountain or they are due to problems during the silhouette extraction step. In this work, we show, how outliers are defined in our context, namely as sub-sequences that are found by a double threshold technique. Therefore, we describe how the anomaly scores for single vertices in the polygonal chains are computed via a histogram distance based approach. We also introduce an improved way to compute reference data and outlier scores and show that this changes allow for significant better outlier detection results.

1 Introduction

Social networks and image sharing platforms enable users to easily share their photos, with millions of pictures uploaded every day. However, most of these photos are not properly tagged. Because of this, they are not easily accessible since they cannot be simply searched for. The aim of our framework presented in [3] therefore is, to be able to recognise mountains in a given image. We hope to overcome this problem for the example of mountain recognition by being able to automatically tag them.

Since our idea is to be able to annotate every image of a mountain, even without GPS information, we have to extract much more precise silhouettes than approaches that will work on GPS tagged images, only. Therefore, it is of imminent importance to correct errors that occur during the initial segmentation step. In this work, we present an enhancement to our outlier detection step from [3], that uses clustering on the reference silhouettes in order to enable a more precise outlier detection.

2 Related Work

There are approaches for automatic tagging, which provide solutions for other motifs than mountains [13]. However, this work is based on image features, which are very similar for different mountains and not feasible for mountain recognition.

The area of mountain recognition is rather small. Baatz et al. [1] use human interaction to correct silhouettes that are first computed by an algorithm. They also have released their data set, which is used in this work. However, our approach is intended to be able to recognise a mountain without human intervention. Other approaches to mountain annotation, such as [2, 6, 12], rely on GPS tags in order to estimate the position of the mountain in the picture. Thus, the number of mountains to compare with is relatively small and because of this, the extraction of the silhouette from the mountain does not have to be as exact as in our case. To our knowledge, our approach is the only approach that uses adaptive correction techniques in order to enhance the quality of the extracted silhouette.

A silhouette is essentially a two dimensional polygonal chain. However, there is not much work on outlier detection on polygonal chains. A closely related problem is outlier detection in time series. There are many approaches to this, such as Hot Sax [8] and its variants [11, 4, 9]. These approaches search for discords rather than outliers in our sense. A discord is defined as the sub-sequence of a given time series that has the greatest distance to all other sub-sequences of the same length. These methods have advantages in cases, where periodic events are measured with a time series, since the length of the discord is given as a parameter. To our problem, this is not directly applicable because neither do we know the length of outliers in the silhouette nor do we want to find the most unusual outlier, only.

Another, but less closely related, approach to outlier detection in time series is change point detection [5, 7]. A change point is a point in a time series, that is unusual for the part of the time series up until that point but introduces a new behaviour to the time series that might be repeated later on. While this method is useful on time series it cannot be adapted easily to polygonal chains, since the latter do not have a strong time dimension where one can assume that earlier events affect later ones.

3 AdaMS Framework and Mountain Identification

In this section we shortly introduce our adaptive mountain silhouette (AdaMS) framework that initially has been presented in [3] to put the outlier detection task presented in this work into perspective. The task of mountain recognition can be divided in two large parts: first, the silhouette extraction from an input image and second, the identification of that silhouette by mapping it to a known silhouette of a mountain.

AdaMS is our approach to solve the first of these problems. It uses a grid based segmentation algorithm for the initial silhouette extraction. This silhouette is then, as a second step, searched for outliers which then get classified. Regions around correctable outliers are then resegmented with a different parameter set. The resulting silhouette is again searched for outliers, until no more correctable outliers are found or a maximum number of iterations has been reached.

The identification of the silhouette itself can be divided into two problems, namely the mapping to a known silhouette and the creation of the reference data, i.e. the known silhouettes. Due to the great number of mountains, the mapping of an extracted silhouette to a labelled silhouette has to be precise and with as little computing cost as possible. Therefore it seems advisable to use a step wise process that uses fast to compute distance measures in a first step in order to exclude reference data with low resemblance. As the number of relevant silhouettes decreases, more precise similarity measures can be applied.

The creation of the reference data is a task, that could theoretically be carried out by humans. However, that would be an immense workload because one labelled silhouette per mountain would not suffice to support a reliable identification. This is because mountains look different from different angles. It seems much more feasible to automatically extract the reference silhouettes from digital elevation maps.

4 Enhancing Outlier Detection in AdaMS

For finding outliers, we first have to define outliers for our application case. Based on this definition, we show the basic AdaMS outlier detection and then introduce recent improvements.

4.1 Outlier Definition

A silhouette is a two dimensional polygonal chain that can be converted to a relative silhouette:

Definition 1 *A* relative silhouette $RS = (v_1, \ldots, v_n)$*,* $n > 0$*, is a polygonal chain with* $v_i = (l_i, a_i)$ *for all* $1 \le i \le n$*, where* $l_i > 0$ *is the length of a line segment and* $a_i \in (-180°, 180°]$ *is the angle relative to the* x*-axis. [3]*

On such a relative silhouette we want to find unusual parts that are caused by obstacles such as trees or segmentation faults due to low contrast. So an outlier $o = (v_i, \ldots, v_j)$, $1 \le i < j \le n$, is a part of a relative silhouette, where the combination of vertices $v_i, \ldots, v_j$ does not fit the usual patterns in relative silhouettes extracted from mountain images. This is introduced formally in the following definition.

Definition 2 *Let* $l > 0$*, and* $RS = (v_1, \ldots, v_n)$ *be a relative silhouette.*
We call $o = (v_i, \ldots, v_j)$ *an* l-outlier *if the following is true:*

1. *For all* v_k*,* $i \le k \le j$*, it holds that* v_k *is a weak anomaly.*
2. *There exist* $m_1, m_2 \in \{i, \ldots, j\}$ *such that* $m_2 - m_1 \ge l$ *and for all* v_k*,* $m_1 \le k \le m_2$*, it holds that* v_k *is a strong anomaly. [3]*

An outlier $o = (v_i, \ldots, v_j)$ *is called a* maximum l-outlier *if and only if neither* $(v_{i-1}, \ldots, v_j)$ *nor* $(v_i, \ldots, v_{j+1})$ *are* l*-outliers.*

Definition 2 mentions strong and weak anomalies. These, in contrast to outliers, are single vertices that have unusual properties, that is, high anomaly scores. We therefore show how these concepts can be defined. Anomaly scores are computed via histograms of parts of relative silhouettes.

Definition 3 *Given a relative silhouette RS, then* $H_{RS}(s, l)$ *denotes the histogram consisting of the points* $v_s, \ldots, v_{s+l-1}$ *of RS and* H_{RS} *denotes the histogram over all vertices of RS. [3]*

Based on these histograms and the distance to the reference histogram H_{ref}, we are now able to introduce the vertices' anomaly scores:

Definition 4 *Given a relative silhouette* $RS = (v_1, \ldots, v_n)$ *and a reference histogram* H_{ref}*.*

The anomaly score

$$an(v_i) := \frac{1}{l} \sum_{j=i-l+1}^{i} d_j$$

of vertex v_i *is the average of the distances* $d_j = \mathrm{dist}(H_{RS}(j, l), H_R)$*.*

With the anomaly scores we can now finally introduce the different kinds of anomalies used in definition 2:

Definition 5 *Let* $RS = (v_1, \ldots, v_n)$ *be the silhouette of an image with corresponding anomaly scores* $an(v_i)$ *for vertex* v_i*, reference anomaly score distribution mean* μ *and standard deviation* σ *and two thresholds* $0 < \tau_{out} < \tau_{in}$*.*

Then we call v_i *a* weak anomaly *if*

$$an(v_i) \geq \mu + \tau_{out} \cdot \sigma$$

and a strong anomaly *if*

$$an(v_i) \geq \mu + \tau_{in} \cdot \sigma. \text{ [3]}$$

This shows, that an outlier in our case is an application of a double threshold technique. The idea here is that often, within an obstacle or a segmentation fault, only small parts of the outlier consist of vertices with unusually high anomaly scores. The rest of the outlier consists of vertices whose anomaly scores are still high, but on their own would not suffice to identify an outlier. Figure 1 shows such an example.

Fig. 1. An outlier – strong anomalies are marked red, weak pink and the silhouette yellow.

4.2 SingleRef Outlier Detection and Reference Data Computation

As in [3], our first version outlier detection algorithm computes the vertices anomaly via a sliding window approach, following the idea outlined in the previous section. More than one window length can be used and the same window length may be used multiple times to induce a weighting to the distances computed by the different window lengths. This is necessary, because a longer window length gives a single vertex more distance scores than a shorter one and thus is represented stronger in the anomaly score.

Given the anomaly values $an(v_i)$ for the vertices, finding the maximum l-outliers according to definition 2 is straightforward. In the first step, we search for sequences of vertices with a length of at least l that all are strong anomalies. Once we have found such a sequence, we let it grow by adding vertices that are neighbours of the currently detected outlier and weak anomalies on both sides.

In regard to reference data computation, the algorithm gets a selection of outlier free silhouettes that where chosen by hand and computes a single reference histogram H_{ref} of all those silhouettes. We therefore refer to this version of the algorithm as SingleRef outlier detection. The statistical properties μ and σ are then computed by using the same window lengths as in the actual outlier detection to compute the anomaly score of every vertex in the reference silhouettes.

4.3 MultiRef – Improving SingleRef

Computing one histogram, in a certain manner, aggregates the data represented by the histogram. Due to the differences in details, a stronger aggregation results in higher standard deviation of the single data points to this aggregates than a weaker aggregation. This, in turn, leads to obscured real anomalies, since distances to the reference data are in general rather high. As the evaluation chapter shows, this leads to either high rates of false positives or relatively low detection rates for SingleRef.

A solution to this problem would be the usage of multiple reference histograms. Intuitively, one would choose one histogram per silhouette, resulting in Histograms $H_{ref}^1, \ldots, H_{ref}^n$ if we assume n reference silhouettes. In order to use more than one reference histogram, however, we have to adjust the distance computation. The base distance used in AdaMS is the following:

Definition 6 *Let* $G = (g_1, \ldots, g_n)$*,* $H = (h_1, \ldots, h_n)$ *be histograms with* n *buckets.*

The above average distance *of* G *to* H *is defined by*

$$\mathrm{dist}(G, H) := \max(|aab(G)|, |aab(H)|) - |aab(G) \cap aab(H)|,$$

where

$$aab(F) := \left\{ i \in \{1, \ldots, n\} \,\middle|\, f_i \geq \frac{1}{n} \sum_{i=1}^{n} f_i \right\}$$

with $F = (f_1, \ldots, f_n)$ *being a histogram with* n *buckets.*

Essentially, by the above average distance, the number of above average buckets that are the same in both histograms is subtracted from the higher number of above average filled buckets.

Theorem 7 *The above average distance from definition 6 is a pseudometric.*

For the proof of this theorem see appendix A.

Now, when comparing a histogram with not just one reference histogram but several, we compute the above average distance to all reference histograms and then choose the minimum of that distances.

Definition 8 *Given a histogram* H *and reference histograms* $H_{ref}^1, \ldots, H_{ref}^n$*, then*

$$\mathrm{dist}_{mr}(H) := \min_{1 \leq i \leq n} \left\{ \mathrm{dist}(H, H_{ref}^i) \right\}$$

is called the min-ref distance*.*

As the number of reference silhouettes is potentially large and it is beneficial to add further silhouettes free of outliers to the reference data, it is clear that the usage of a reference histogram per reference silhouette is not feasible and the number of histograms has to be reduced. On the other hand, we want to minimise the loss of detail due to aggregation. We therefore utilise a k-means clustering [10] on the reference silhouettes' histograms.

By this, we are able to reduce the number of reference silhouettes to any given k while ensuring, that we loose as little detail as possible, because the subadditivity holds for the above average distance. This means, the distance to the cluster representative is an upper bound on the distance to the closest silhouette's histogram. We ensure small distances between representatives and members of the cluster by clustering with random start representatives 1000 times and choosing the clustering with the smallest quadratic distance. It is also noteworthy, that by setting $k = 1$ we only get one reference silhouette that is identical to the reference silhouette of the SingleRef method.

By this, as the evaluation shows, we are able to achieve better outlier detection without changing our outlier detection algorithm as such nor do we need more reference data.

4.4 Further Steps

The next steps after identifying the outliers are to classify them into obstacles in the picture and errors in the segmentation step, for example due to low contrast between parts of the mountain and the sky. As described in [3], we use four classes of outliers, namely obstacles, segmentation errors where the silhouette is too high, segmentation errors where the silhouette is too low and false positives. For the classification we utilise a k-nearest neighbour approach on the outliers' histograms.

5 Evaluation

In order to evaluate the approaches introduced in the previous section, we manually annotated a test set of 111 outliers from 14 silhouettes, that, in total, consist of 3580 vertices. The silhouettes have been automatically extracted from the images by a variant of the the segmentation algorithm presented in [3], but without the outlier detection steps and therefore without the adaptive correction.

The outlier detection algorithm has been trained with 48 silhouettes that are mostly free of outliers and have been extracted with the same mechanism as described above. These have been clustered 1000 times per number of clusters and the clustering with the lowest overall quadratic distance of histograms in respect to their cluster representative has been chosen.

We first evaluated the precision and recall based on detected outlier vertices for different values for k, the number of reference silhouette clusters and thus reference histograms. As parameter set we chose the minimum length for an inner outlier $l = 3$. The inner and outer thresholds have been set to $\tau_{in} = 2$ and $\tau_{out} = 1$. The results of this are shown in table 1. Note here, that for $k = 48$ no clustering is used, but here one reference histogram per reference silhouette is used.

Precision and recall are computed by counting the vertices that have been declared as parts of outliers correctly and dividing that number by the total number of detected outlier vertices respectively the total number of annotated outlier vertices. The last number is shown in the last row of the table. It can be seen here, that MultiRef, for every tested value of k shows better results than SingleRef. Especially recall is much better than with SingleRef and gets higher

Method	Value	Silhouette														Total
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	
SingleRef	Precision	86	0	18	0	0	0	82	0	58	0	27	98	48	90	72
	Recall	54	0	15	0	0	0	65	0	25	0	5	43	47	46	33
$k = 3$	Precision	87	0	69	42	0	42	79	22	51	88	52	95	72	86	73
	Recall	66	0	41	29	0	66	72	20	64	8	17	58	69	47	46
$k = 5$	Precision	82	0	68	37	0	70	82	43	44	80	56	89	73	89	74
	Recall	68	0	39	37	0	66	71	43	52	19	28	65	75	62	54
$k = 7$	Precision	89	0	57	40	0	100	82	42	45	79	61	97	71	89	**77**
	Recall	68	0	17	28	0	60	71	40	50	19	27	53	68	58	49
$k = 10$	Precision	85	0	59	41	0	100	78	42	45	79	73	95	72	89	**77**
	Recall	65	0	18	29	0	60	71	40	51	19	28	60	63	56	49
$k = 20$	Precision	88	0	57	40	0	65	79	39	45	82	74	94	72	89	74
	Recall	68	0	18	37	0	64	88	41	52	18	28	49	70	56	49
$k = 30$	Precision	84	0	78	41	0	65	89	41	46	76	61	89	76	90	76
	Recall	70	0	59	48	0	64	91	39	59	24	38	68	79	71	**61**
$k = 48$	Precision	84	0	78	41	0	65	71	45	46	67	66	89	78	91	75
	Recall	70	0	59	48	0	64	91	39	59	24	38	69	79	71	**61**
#Outlier vertices		305	30	71	234	46	47	144	121	147	344	192	682	315	902	3580

Table 1. Evaluation on detected outlier points. Precision and Recall are given in percent. Thresholds are $\tau_{in} = 2$ and $\tau_{out} = 1$, $l = 3$.

overall with increasing k. Interestingly, while there is a huge improvement in recall from $k = 20$ to $k = 30$, recall does not get any better when using one histogram per reference silhouette. Precision, too, is higher for the MultiRef variants in respect to SingleRef, but it is noteworthy here, that, instead of rising with the number of clusters, a maximum is reached for $k = 7$ respectively $k = 10$ and for bigger values, a slight decrease can be noticed. Note here, that the results in table 1 are given without carrying out the classifying of outliers. By that step, some outliers will be classified as false positives and thus precision after classification should increase.

Table 1 also shows, that for silhouettes 2 and 5, no correct outliers have been found at all. The outliers in both silhouettes are rather short. The three outliers in silhouette 2 have a length of 8, 9 and 13 vertices and the four outliers in silhouette 5 have a length of 9, 10, 12 and 15 vertices.

Based on this observation, table 2 shows the number of outliers that have been hit by the detection algorithm. The same parameter set as above has been used for this. An outlier is counted as being hit by the detection algorithm, if at least one vertex of it has been detected as being part of an outlier. In context with AdaMS, if this happens, the silhouette will be recomputed in this region and can thus be corrected. It is clear, that all algorithms have problems with smaller outliers. For the extraction of good silhouettes, it is more important to find huge outliers, and our detection rates for those are promising, in general. The detection rate increases with the length of the outlier for every variant. On

Method	$\lvert out \rvert \leq 5$	$5 < \lvert out \rvert \leq 10$	$10 < \lvert out \rvert \leq 20$	$20 < \lvert out \rvert \leq 50$	$50 < \lvert out \rvert$	Total
SingleRef	0	1	2	5	10	18
$k = 3$	2	5	7	14	12	40
$k = 5$	5	7	9	15	13	49
$k = 7$	4	5	7	14	13	43
$k = 10$	4	5	8	14	13	44
$k = 20$	4	8	8	15	12	47
$k = 30$	6	10	11	16	15	58
$k = 48$	6	10	11	16	15	58
#Outliers	18	38	19	21	15	111

Table 2. Number of detected outliers. The length of outliers is denoted by $|out|$. Thresholds are $\tau_{in} = 2$ and $\tau_{out} = 1$, $l = 3$.

the other hand, with increasing k, the detection rate in general increases, too. Interestingly, for $k = 5$ more outliers have been detected than for surrounding values of k. However, due to the relatively small numbers of outliers used in our evaluation, this might be coincidence.

τ_{in}	τ_{out}	l	$\lvert out \rvert \leq 20$	$20 < \lvert out \rvert \leq 50$	$50 < \lvert out \rvert$	Total	Precision	Recall
2	1	3	21	15	13	49	74	54
1.5	1	3	25	17	14	56	69	58
2.5	1	3	12	13	12	37	77	49
2	**0.75**	3	24	15	13	52	71	59
2	**1.25**	3	20	15	13	48	76	50
2	1	**1**	21	15	13	49	72	55
2	1	**5**	17	14	13	44	76	53

Table 3. Effect of parameter changes based on MultiRef with $k = 5$.

The results in table 3 show the results of our investigation of the effects of parameter changes. Essentially, lowering τ_{in} or l results in a greater number of detected outliers, while raising that values reduces the number of outliers. Changes to τ_{out} affect the size of detected outliers. The lower τ_{out} becomes, the bigger are the resulting outliers. Due to overlaps with the real outliers, lowering of one of the values leads to higher recall and decreased precision. Increasing them raises precision but induces losses to recall.

In summary, the results show that even for the worst choice of k, the number of hit outliers is more than two times higher than that detected by SingleRef, while at the same time precision and recall are increased.

6 Conclusion and Future Work

In this work, we have presented our definition of outliers and our approach to make the detection of outliers in polygonal chains that represent silhouettes extracted from pictures of mountains more effective. Our results show, that the MultiRef variant introduced in this work greatly improves the outlier detection results. The number of detected outliers is increased by a factor of two to three, depending on the number of clusters, while precision and recall are also increased.

However, there are some points that we want to address in the future. One of the main questions is, whether our distance function as given in definition 6 is ideal or if a more elaborate histogram distance function such as the earthmover's distance [14] will yield better results. Also, we plan to use additional data for the outlier detection. The contrast strength seems to be a good measure, since segmentation faults usually occur in regions of low contrast.

A Appendix

The fact of the above average distance being a pseudometric is of importance since it ensures that the triangle equation is satisfied by that construct. From this it can be derived that the greater the distance between two histograms is, the greater the difference between those is and there are no short cuts by using intermediate histograms.

Proof. In order to show that the above average distance is a pseudometric, four properties have to be shown. Let g, h, k be histograms with the same number of buckets.

Non-negativity $\text{dist}(g, h) \geq 0$. This is trivial since $|aab(g)| \geq |aab(g) \cap aab(h)|$ and $|aab(h)| \geq |aab(g) \cap aab(h)|$, thus

$$\max(|aab(g)|, |aab(h)|) \geq |aab(g) \cap aab(h)|.$$

Identity of indiscernibles

$$\begin{aligned}\text{dist}(g, g) &= \max(|aab(g)|, |aab(g)|) - |aab(g) \cap aab(g)| \\ &= |aab(g)| - |aab(g)| = 0.\end{aligned}$$

Symmetry

$$\begin{aligned}\text{dist}(g, h) &= \max(|aab(g)|, |aab(h)|) - |aab(g) \cap aab(h)| \\ &= \max(|aab(h)|, |aab(g)|) - |aab(h) \cap aab(g)| \\ &= \text{dist}(h, g)\end{aligned}$$

since both $\max(\cdot, \cdot)$ and the intersection of sets are symmetric functions.

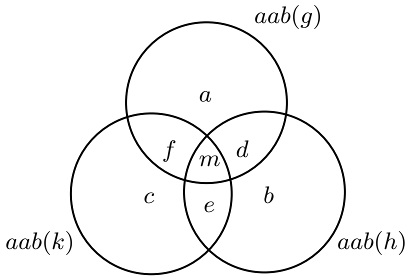

Fig. 2. Sets for proof of subadditivity.

Subadditivity In order to proof the subadditivity an auxiliary construction is necessary. As shown in figure 2, the sets $aab(g), aab(h)$ and $aab(k)$ are split in four disjoint sets each, such that $aab(g) = a \cup d \cup f \cup m$, $aab(h) = b \cup d \cup e \cup m$ and $aab(k) = c \cup e \cup f \cup m$.

Now, without loss of generality, let $|aab(g)| \geq |aab(h)|$. Then $\text{dist}(g, h) = |aab(g)| - |aab(g) \cap aab(h)| = |a| + |f|$.

Case 1 Now, let $|aab(h)| \geq |aab(k)|$. Then it holds that $\text{dist}(k, h) = |b| + |d|$ and $\text{dist}(g, k) = |a| + |d|$, since $|aab(g)| \geq |aab(h)| \geq |aab(k)|$. Thus, in this case

$$\begin{aligned}&\text{dist}(g, k) + \text{dist}(k, h) - \text{dist}(g, h)\\=&|a| + |d| + |b| + |d| - |a| - |f|\\=&|b| + 2|d| - |f| \geq 0,\end{aligned}$$

because

$$\begin{aligned}|aab(h)| &\geq |aab(k)|\\\Rightarrow |b| + |d| + |e| + |m| &\geq |c| + |e| + |f| + |m|\\\Rightarrow |b| + |d| &\geq |c| + |f| \geq |f|.\end{aligned}$$

Case 2 Assume now, that $|aab(h)| < |aab(k)|$, so $\text{dist}(k, h) = |c| + |f|$.

Case 2.1 Let $|aab(g)| \geq |aab(k)|$. It follows that

$$\begin{aligned}&\text{dist}(g, k) + \text{dist}(k, h) - \text{dist}(g, h)\\=&|a| + |d| + |c| + |f| - |a| - |f|\\=&|d| + |c| \geq 0.\end{aligned}$$

Case 2.2 The last case to be considered occurs if $|aab(g)| < |aab(k)|$ which results in $\text{dist}(g, k) = |c| + |e|$. Then

$$\begin{aligned}&\text{dist}(g, k) + \text{dist}(k, h) - \text{dist}(g, h)\\=&|c| + |e| + |c| + |f| - |a| - |f|\\=&2|c| + |e| - |a| \geq 0.\end{aligned}$$

This is because of a similar argument to case 1, because

$$|aab(k)| \geq |aab(g)| \Rightarrow |c| + |e| \geq |a| + |d| \geq |a|.$$

□

References

1. Baatz, G., Saurer, O., Köser, K., Pollefeys, M.: Large Scale Visual Geo-Localization of Images in Mountainous Terrain. In: Computer Vision - ECCV 2012 (2012)
2. Baboud, L., Čadík, M., Eisemann, E., Seidel, H.P.: Automatic Photo-to-terrain Alignment for the Annotation of Mountain Pictures. In: Proc. of the 2011 IEEE Conference on Computer Vision and Pattern Recognition (2011)
3. Braun, D., Singhof, M., Conrad, S.: AdaMS: Adaptive Mountain Silhouette Extraction from Images. In: Proc. of MLDM 2016 (2016)
4. Buu, H.T.Q., Anh, D.T.: Time Series Discord Discovery Based on iSAX Symbolic Representation. In: Third International Conference on Knowledge and Systems Engineering (2011)
5. Fawcett, T., Provost, F.: Activity Monitoring: Noticing Interesting Changes in Behavior. In: Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining (1999)
6. Fedorov, R., Fraternali, P., Tagliasacchi, M.: Mountain Peak Identification in Visual Content Based on Coarse Digital Elevation Models. In: Proc. of the 3rd ACM International Workshop on Multimedia Analysis for Ecological Data (2014)
7. Kawahara, Y., Sugiyama, M.: Change-Point Detection in Time-Series Data by Direct Density-Ratio Estimation. In: Proc. of 2009 SIAM International Conference on Data Mining (SDM2009), (2009)
8. Keogh, E., Lin, J., Fu, A.: Hot Sax: Efficiently Finding the Most Unusual Time Series Subsequence. In: Fifth IEEE International Conference on Data Mining (ICDM'05) (2005)
9. Khanh, N.D.K., Anh, D.T.: Time Series Discord Discovery Using WAT Algorithm and iSAX Representation. In: Proceedings of the Third Symposium on Information and Communication Technology (2012)
10. MacQueen, J.: Some Methods for Classification and Analysis of Multivariate Observations. In: Proceedings of the fifth Berkeley Symposium on Mathematical Statistics and Probability (1967)
11. Pham, N.D., Le, Q.L., Dang, T.K.: HOT aSAX: A Novel Adaptive Symbolic Representation for Time Series Discords Discovery. In: Asian Conference on Intelligent Information and Database Systems (2010)
12. Porzi, L., Buló, S.R., Valigi, P., Lanz, O., Ricci, E.: Learning Contours for Automatic Annotations of Mountains Pictures on a Smartphone. In: Proc. of the International Conference on Distributed Smart Cameras (2014)
13. Rischka, M., Conrad, S.: Image Landmark Recognition with Hierarchical K-Means Tree. In: Database Systems for Business, Technology and Web (BTW 2015) (Mar 2015)
14. Rubner, Y., Tomasi, C., Guibas, L.J.: A Metric for Distributions with Applications to Image Databases. In: Computer Vision, 1998. Sixth International Conference on. IEEE (1998)