Frank Rosenthal, Ulrike Fischer, Peter B. Volk, Wolfgang Lehner

# Global Slope Change Synopses for Measurement Maps

**Erstveröffentlichung in / First published in:**

*Ninth IEEE International Conference on Data Mining*. Miami Beach, 06.12.-09.12.2009. IEEE, S. 956-961. ISBN 978-1-4244-5242-2

DOI: https://doi.org/10.1109/ICDM.2009.117

Final edited form was published in "Ninth IEEE International Conference on Data Mining. Miami Beach 2009", S. 956-961, ISBN 978-1-4244-5242-2
https://doi.org/10.1109/ICDM.2009.117

# Global Slope Change Synopses for Measurement Maps

Frank Rosenthal, Ulrike Fischer, Peter B. Volk and Wolfgang Lehner
*Database Technology Group, Faculty of Computer Science*
*Dresden University of Technology*
*01062 Dresden, Germany*
*Email: dbinfo@mail.inf.tu-dresden.de*

***Abstract*—Quality control using scalar quality measures is standard practice in manufacturing. However, there are also quality measures that are determined at a large number of positions on a product, since the spatial distribution is important. We denote such a mapping of local coordinates on the product to values of a measure as a measurement map. In this paper, we examine how measurement maps can be clustered according to a novel notion of similarity—mapscape similarity—that considers the overall course of the measure on the map. We present a class of synopses called global slope change that uses the profile of the measure along several lines from a reference point to different points on the borders to represent a measurement map. We conduct an evaluation of global slope change using a real-world data set from manufacturing and demonstrate its superiority over other synopses.**

***Keywords*-measurement map; mapscape similarity; synopsis;**

## I. Introduction

Product quality is one of the most important concerns in manufacturing, since low or fluctuating product quality may result in dissatisfied customers. Hence, quality control needs to be employed. There are different strategies, but the core concept is to constantly monitor quality, even at intermediate processing steps, and to check for any quality degradation. When a critical deviation is detected, the manufacturing process has to be adjusted.

Quality control is standard practice and there is a rich methodology for using *scalar* quality measures, such as *dimensions of a product*. However, some quality measures, e.g. surface temperature, depend on the position of measurement. For example, heavily machined areas are hotter than other areas. To use such quality measures in standard quality control frameworks, the spatially distributed measure needs to be transformed into a scalar measure. For example, the temperature at a certain position or an aggregation like mean temperature can be used as representation. This approach may be admissible, but in any case, valuable information is lost.

Figure 1 shows example distributions of surface temperature that occur when a workpiece is formed or machined. Such distributions are mappings $(X, Y) \rightarrow Z$ of local coordinates $(X, Y)$ to a measure $Z(X, Y)$, and we denote the mapping as *measurement map*. The distribution mainly depends on the performed processing steps, but also on the internal composition of the raw material.

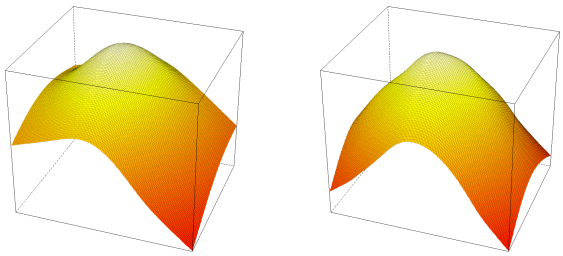


Figure 1. Example Measurement Maps

A key observation is that when the same processing steps are performed on several workpieces, the differences in the temperature distribution mostly characterize their internal structure. Therefore, we need to classify measurement maps into a set of classes of workpieces for which appropriate corrective actions are known. However, since this set does not exist currently, we need to employ clustering first to establish it. In the rest of this paper, we consider only clustering.

Standard clustering algorithms cannot be applied straightforward, which is due to the specific notion of similarity of measurement maps. Intuitively, it is the overall shape of a measurement map or, figuratively, *the landscape* that determines similarity. The examples in Figure 1 illustrate this. The measurement maps in Figure 1 are similar, since both exhibit a high elevation near the center and an approximately concentric decrease in all directions from the center to the borders. We denote this notion of similarity as *mapscape similarity*. We require *scale* and *rotation* invariance. For example, a map could be rotated by $90°$ around the vertical axis and would still be similar to the original in terms of *mapscape similarity*. These invariances are not ensured by applying standard distance metrics like Euclidean distance directly to two maps.

Therefore, we propose to create a synopsis of a measurement map that can be compared using standard distance metrics like Euclidean distance. The synopsis shall represent the overall shape of the map. Clustering of measurement maps can then be performed by clustering the associated synopses. In this paper, we present a class of such synopses,

Final edited form was published in "Ninth IEEE International Conference on Data Mining. Miami Beach 2009", S. 956-961, ISBN 978-1-4244-5242-2
https://doi.org/10.1109/ICDM.2009.117

called *global slope change* ($GSC$). The basic idea is to use the slope profile along several cuts through the surface of the measurement map. These cuts are made from a reference point to different points on the borders of the measurement map. The synopses differ in the aggregation of the slope profiles, but they all ensure scale and rotation invariance. Note that we restrict our discussion to rectangular measurement maps. However, our approach can be extended easily to general measurement maps with convex borders.

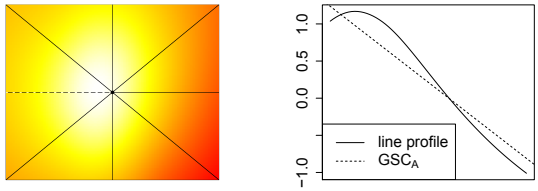
In the next section, we describe related work and discuss its limited ability to capture mapscape similarity. In Section III, we present the structure of global slope change synopses and how they are calculated. We evaluate our approach and compare it with other synopses in Section IV and conclude the paper in Section V.

## II. Related Work

Since the notion of mapscape similarity is novel, $GSC$ is the first synopsis to capture it. However, in *content-based image retrieval* (CBIR), similar synopses with similar goals were proposed, since CBIR requires to determine the similarity between two images based on the content. Since a measurement map can be viewed as a gray-scale image, with the measure providing the luminance information, these synopses can be applied. However, they are not suitable to capture mapscape similarity, as we will outline in the following. A large body of work has been created over the last 20 years and a number of surveys such as [1]–[3] summarize the progress in the field. A commonality of most proposed CBIR systems is that they define similarity in terms of features extracted from the images at hand [1]. A multitude of features have been proposed, but they can generally be classified in three main categories [2].

**Shape**: This class of features is based on prominent geometric details. For example, edges can be detected and then transformed into features, e.g. the distance of the points of the edge with respect to some reference point [4]. This approach is not applicable in our scenario since there are no clearly defined meaningful shapes.

**Texture**: The second class of features is defined by the fine-granular structure that emerges from virtually all surfaces or objects, e.g. clouds, concrete or shrubs. With regard to processing, features based on texture are often created by analyzing the neighborhood (e.g. the 8 surrounding pixels) of all pixels in an image. This approach is by definition *local*, while in our scenario, the global form of the map defines similarity. But since this class of features is applicable whatsoever, we selected a representative for our evaluation. The method described in [5] uses the texture spectrum of an image, which is gathered by moving a window (neighborhood of a pixel) over the image and creating a histogram of the encountered *local binary patterns* $LBP$. Patterns are calculated from neighborhoods, by comparing the luminance of each pixel in the neighborhood $p_n$ to the luminance of the central pixel $p_c$. The windows can have different forms, e.g. square or circle.




(a) Map and cut lines of $GSC$. (b) Shape of map along line.

Figure 2. Structure of $GSC$ synopses.

**Color**: The last class of features is derived from color. A huge amount of work has considered problems of equality of color, e.g. under different lighting or according to human perception. This problem does not exist in our setting. The most common approach is to represent the image–or parts of it–with a color/luminance histogram [6]. This approach does not capture spatial information, which is required in our scenario. However, since this approach is used often, we do use it in the evaluation.

## III. Global Slope Change

In this section, we will detail the notion of mapscape similarity and introduce the class of *global slope change* ($GSC$) synopses that are designed to capture it.

Mapscape similarity is intuitively defined as a similarity of the overall shape of a measurement map. For example, two maps with a high elevation near the center and a decrease in all directions (Figure 1) are considered similar although they differ locally. In other words, mapscape similarity results from a similar sequence of changes of the measure when viewing the map from an equal reference point in both maps. Consider Figure 1, where the profile in the vertical center—viewed from left to right—consists of a gradual rise followed by a gradual decrease in the measurement values in both maps. We will use this view as the basis for our proposed *Global Slope Change* ($GSC$) synopses.

There are two challenges when trying to capture mapscape similarity: scale invariance and rotation invariance. Scale invariance is required for two reasons. First, measurement maps can have different sizes, i.e. the domains of $X$ and $Y$ may differ, yet they must still be comparable with respect to the global shape. Second, since only the overall shape is important, a homogeneous scaling in the domain of measure $Z$ must be tolerable, too. Rotation invariance is important, since rotated measurement maps exhibit the same distribution in just another orientation.

Final edited form was published in "Ninth IEEE International Conference on Data Mining. Miami Beach 2009", S. 956-961, ISBN 978-1-4244-5242-2
https://doi.org/10.1109/ICDM.2009.117

### A. The Global Slope Change Synopses

We now give an overview of the common steps of creating Global Slope Change synopses and then describe each step in detail. The first step applies a normalization to the measurement map. Second, we define a reference point based on which we observe the slope change and extract profiles along these lines. Then, we apply a discretization and represent the profiles by a discrete number of values. The fourth step aggregates these values to a constant number of feature values. Finally, we combine all these feature values to one synopsis.

**Normalization**: In order to assure comparability in $Z$, we first apply a normalization to the measurement map. For this, we use the z-score normalization [7], which normalizes the values of measure $Z$ to mean 0 and variance 1.

**Profile Extraction**: We analyze the change of the global slope profile of a map. Since the slope varies in dependence of the two spatial dimensions $(X, Y)$, we need a comparable reference point that defines the origin for profiles of the map. Hence, profiles with different orientations in the $(X, Y)$-plane can be extracted. In our application setting, we noticed that the different classes of maps were well differentiable when comparing the slope of the map from the center of the map to several points on the border. Figure 2(b) shows this slope profile along a line marked in Figure 2(a). Hence, we choose the center of the map as reference point. Additionally, choosing the center as reference point is beneficial to realize rotation invariance. Therefore, we perform $n$ cuts through the surface of the measurement map from the center to the border of the map. One cut realizes a straight *line* in the $(X, Y)$-plane, and the associated $Z$ values form the slope profile along this line. The target points of the lines at the borders are chosen in equally spaced intervals in order to observe each part of the map evenly, resulting in a star-shaped pattern (Figure 2(a)).

**Discretization**: Next, we represent each line by $p$ values of the measure $Z$. For the horizontal and vertical lines, we choose these values from the discrete raster of the map. For the diagonal lines, we interpolate to acquire $p$ values that have the same spatial distance as the values on the horizontal and vertical lines.

**Aggregation**: We have divided our map into $n$ different lines and represented each line by $p$ values. This could be used as a form of representation, but it is not very compact and is not scale-invariant since maps can have different sizes and thus different numbers of values on a line. We need to aggregate these values to a constant number of $k$ feature values. Therefore, we require an aggregation function ($aggFeatures$) that transforms the $p$ values on a line into $k$ feature values. In this paper, we present three different approaches to aggregate the values along a line to feature values. All three approaches encode the global shape of the map and ensure scale invariance. According to different parameters, they all result in the same number of features when applied to different maps.

Finally, we need to combine the features of all lines to one synopsis or *representation vector*. This synopsis represents the overall slope profile of the whole map. Since each map outputs the same number of features and because the features are numeric, we can compare our synopses using standard distance metrics like Euclidean distance. A clustering algorithm can then be applied directly on the synopses. For now, our synopsis requires one parameter: the number of lines $n$, because we will introduce a rule to set $p$, while $k$ is determined by the chosen synopsis.

### B. Average Global Slope Change

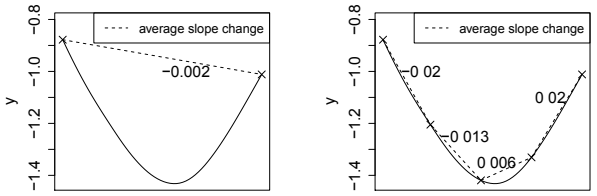
In this section, we want to introduce a first naive approach to calculate the aggregation function $aggFeatures$. This first idea consists of calculating the difference of two consecutive points on a line. By calculating the difference, we generalize from different scales and base levels of measure $Z$. Then we can aggregate the differences. For this basic approach, called *average global slope change* ($GSC_A$), we calculate the average over the differences. This average represents the slope profile along a line, the sign represents the direction of the slope, and the value is the rate of change. The $GSC_A$ approach results in only one feature value per line, i.e., $k = 1$. The dotted line in Figure 2(b) illustrates this feature value, which is negative in this example and thus represents an important global characteristic of the map: the decrease of the measure from the center to the border.

This approach has the advantage that it is simple to calculate, requires no parameters beside the number of lines $n$, and only stores $n$ values in the final representation vector (since $k = 1$). However, $GSC_A$ may reflect the course of the map incorrectly. Since we only have one feature value, we not only generalize local fluctuations along the lines, but we might also lose important global characteristics of the map. As an example, Figure 3(a) shows the profile along a line chosen from a different map and a representation of the associated average global slope change. Since the line shows a deep minimum in the middle, the average global slope change is close to zero. Therefore, it does not reflect the line's real course and represents the U-shaped profile equal to a near flat profile.

### C. Multi-Segment Global Slope Change

Since $GSC_A$ may overgeneralize in some cases, it may yield distorted feature values. In this section, we extend our first naive approach in order to compensate for this problem. We need to approximate the lines more precisely by analyzing several sections of the lines. We refine $GSC_A$ by dividing each line into $s$ equally sized segments and describe the slope profile along the segments separately. These segments have constant size in order to be comparable between different maps. The associated feature value is now

Final edited form was published in "Ninth IEEE International Conference on Data Mining. Miami Beach 2009", S. 956-961, ISBN 978-1-4244-5242-2
https://doi.org/10.1109/ICDM.2009.117




(a) GSC$_A$ (b) GSC$_{AMS}$

Figure 3. Average Slope Global Change

calculated according to $GSC_A$ for each of these segments separately. We call this strategy *Multi-Segment Global Slope Change* ($GSC_{AMS}$), resulting in a final representation vector of size $n*s$ ($k = s$). For this synopsis, we have to define a second parameter besides the number of lines: the number of segments $s$.

Looking back at the example from the $GSC_A$ approach, we now divide the line into four segments (Figure 3(b)) and calculate the $GSC_A$ for each of these segments. As a result, the corresponding feature vector reproduces the minimum in the middle of the line. A benefit of this approach is that we can adopt the accuracy of our description by changing the number of segments. With increasing number of segments, the lines are approximated more precisely and with more detail. However, if we set the number of segments too high, we might incorporate too many details of the line into the final synopsis.

### D. Wavelet-based Global Slope Change

In the last subsection, we proposed an approach that can be used to capture the slope profile along a line at different levels of detail, i.e., resolution. Wavelet analysis [8] is a well-known approach and widely used in the areas of image and signal processing, and it can similarly be used to approximate a sequence of values at multiple levels of resolution. Therefore, in this section, we want to calculate the feature values using wavelet analysis.

The wavelet transform divides an input function into low- and high-frequency components at progressively coarser scales of resolution. For this, the discrete wavelet transform uses discrete-time filterbanks and produces wavelet coefficients at different resolutions $r$ as output. The simplest and most used wavelet transform is the Haar wavelet transform, which simply pairs up input values, providing the differences as wavelet coefficients and passing the sums. The wavelet coefficients at resolution $r$ are calculated from resolution level $r-1$. Additionally, a normalization is applied to each resolution level. When using the Haar wavelet transform to implement $aggFeatures$, we gain a new parameter—the resolution level $r$—for which we calculate the according wavelet coefficients. We get $k = 2^r$ feature values per line, because wavelet analysis yields $2^r$ coefficients. An important characteristic of the wavelet transform is that it requires the number of input values $p$ to be a power of two. Depending on the choice of $p$, the maximum resolution level $R$ of the wavelet transform is $ld(p) - 1$. For the choice of $p$, our goal is to do as less interpolation as necessary in order to avoid adding or losing information in the map. Therefore, we choose $p$ close to half the average size of the maps and calculate $p$ values on the lines in equally spaced distances. As a result, we do not need to perform interpolation on the straight lines in most cases; only the diagonal lines require some interpolation.

In summary, equivalent to the $GSC_{AMS}$ approach, we have two parameters left for this synopsis: the number of lines $n$ and the resolution level $r < R$.

### E. Achieving Rotation Invariance

When the map is rotated, the lines will correspondingly move around the reference point. Since the final representation vector is created from a fixed starting line, rotating the map will result in a different vector. To remove the effect of rotation, our idea consists of finding the best arrangement of two maps and determining the distance for this arrangement. Therefore, we adjust the distance function, so that we determine the minimal distance between all possible rotations of two maps. With this approach, we find the best arrangement of two similar maps.

To adjust the distance function, we fix one representation vector $v_1$ and rotate the other $v_2$. Since we used the center of the map as reference point, rotating the vector is equivalent to rotating the map around the center point. Now, to rotate the vector, we start from a given vector $v_2^1$ and perform a circular right shift $n$ times. Apparently, when doing one shift operation, all feature values for one line have to be moved together, since we only want to rotate the lines. As a result, we get $n$ vectors $\{v_2^1, v_2^2, ..., v_2^n\}$, which represent the slopes along the lines at different rotations of the map. Finally, we calculate the distances between vector $v_1$ and all possible rotations of vector $v_2^r$ and pick the minimum distance as representative:

$$rotInvDist(v_1, v_2) = \min_{1 \le r \le n} dist(v_1^1, v_2^r).$$

## IV. Evaluation

We now present the results of the experimental evaluation of the GSC synopses and a comparison with selected related work. We implemented all variants of $GSC$ in `Java` using the open-source data mining framework `RapidMiner` as well as the direct comparison of two maps using the Euclidean distance ($ED$), a color histogram ($CH$) [6] and the local binary pattern ($LBP()$) operator from [5].

The real-world test set consists of 90 measurement maps that have been acquired from a manufacturing process. The

Final edited form was published in "Ninth IEEE International Conference on Data Mining. Miami Beach 2009", S. 956-961, ISBN 978-1-4244-5242-2
https://doi.org/10.1109/ICDM.2009.117

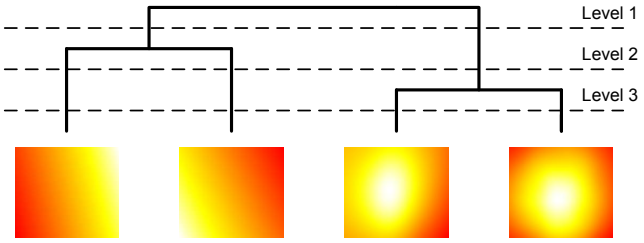
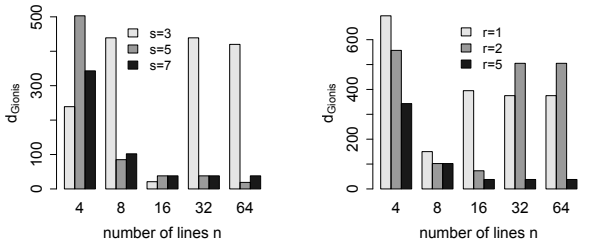



Figure 4. Example dendrogram of measurement maps.




(a) $GSC_{AMS}$: performance for selected number of segments $s$. (b) $GSC_W$: performance for selected resolution levels.

Figure 5. Parameter influence on the distance of clusterings to $C_{ref}$.

maps have different sizes, varying from 100 to 300 values in $X$ and $Y$ direction. Domain experts performed a manual inspection of the test set and identified five groups according to mapscape similarity. Each map was assigned a label and this labeled set serves as reference clustering $C_{ref}$.

### A. Experimental Setup

To measure how well $GSC$ captures mapscape similarity, we apply it to maps in the test set and cluster the created synopses. The resulting clustering should resemble a predefined meaningful clustering $C_{ref}$ as closely as possible. However, a clustering does not yield consistent and meaningful labels. There is no given correspondence between the groups in the result and those in $C_{ref}$, especially since the number of groups might differ. Therefore, we use the method proposed in [9], which presents a distance measure between clusterings $d_G$. The basic idea is to check whether clusterings agree on the assignment of all possible pairs of objects and to increment a counter whenever clusterings disagree. For example, two clusterings disagree if a pair of objects is in the same group in clustering $\mathcal{C}_1$ but is in different groups in clustering $\mathcal{C}_2$.

Using $d_G$, we can evaluate clustering results quickly. However, in general, the result does not solely depend on the selected synopsis and the used parameters but also on the selected clustering algorithm. We experimented with several algorithms to find a suitable one. Partitioning clustering algorithms like k-Means generally had a rather bad performance, which could be due to the form of the clusters. Density-based clustering can help in such situations. We got very good results using DBSCAN [7]. However, DBSCAN has two parameters and the optimal values for these parameters varied when we changed the parameters of the synopses. This means a full evaluation using DBSCAN requires an expensive parameter optimization for each tested parameter combination of our synopses.

To avoid this problem, we used Agglomerative Nesting (AGNES) [7], a hierarchical clustering algorithm. The key benefit is that no parameters have to be set, aside from the selection of the linkage strategy. We evaluated several linkage strategies and found that *complete linkage* provided the best overall results. The result of AGNES is a hierarchical clustering structure. It can be cut at a certain level $l$ to get a single clustering. For example, a cut at the topmost node yields a clustering that consists of one cluster containing all objects. This is illustrated in Figure 4.

In the evaluation, we cut a hierarchical cluster model at all levels and acquire an ensemble of clusterings $\mathbf{C} = \{\mathcal{C}_1, \mathcal{C}_2, ..., \mathcal{C}_k\}$, with $k$ being the number of levels (90 in our scenario, since we have 90 maps). For each $\mathcal{C}_i \in \mathbf{C}$, the distance $d_G(\mathcal{C}_1, \mathcal{C}_{reference})$ is calculated, which results in a set of distances $\mathbf{D} = \{d_G(\mathcal{C}_i, \mathcal{C}_{reference})\}, i = 1 \ldots k$. We argue that the minimal distance $d_G$ in $D$ is an appropriate measure for the generalization power of the tested synopsis, since it represents how well the best clustering result resembled $C_{ref}$. Hence, this is a result that can be achieved if the clustering's parameterization is good.

For the clusterings that were performed using DBSCAN, we observed that a well parameterized run of DBSCAN can yield a result equal to the best results of AGNES. Therefore, we use AGNES in our evaluation to alleviate the parameterization problem of DBSCAN.

### B. Parameter Influence

In this section, we evaluate how the parameters of the $GSC$ synopses influence the achievable clustering quality. The results are depicted in Figure 5 as a bar plot of the minimal $d_G$ with respect to different parameter values.

$GSC_{AMS}$: This synopsis has two parameters: the number of lines $n$ and the number of segments $s$. For $s = 1$, $GSC_{AMS}$ is equal to $GSC_A$. We evaluated all combinations of $n = 4, 8, 16, 32, 64$ and $s = 1 \ldots 7$. As can be seen from Figure 5(a), the distance $d_G$ becomes small for larger numbers of segments $s$ and larger number of lines $n$. There are two exceptions for $s = 3$ and $s = 4$, where larger $n$ lead to an increase in $d_G$. We conclude that large values for both $n$ and $s$ generally lead to better results, although the optimal choice of $s$ is difficult because of the possible occurrence of outliers.

$GSC_W$: This synopsis has two parameters: the number of lines $n$ and the resolution level $r$. We did not vary the number of input values $p$ but set it to 64 i.e., close to half

Final edited form was published in "Ninth IEEE International Conference on Data Mining. Miami Beach 2009", S. 956-961, ISBN 978-1-4244-5242-2
https://doi.org/10.1109/ICDM.2009.117

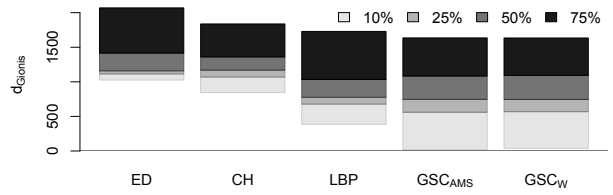



Figure 6. Comparison of minimum and average distance of clusterings to $C_{ref}$.

the average size as stated in Section III-D. We evaluated all combinations of $n = 4, 8, 16, 32, 64$ and $r = 0 \ldots 5$. As can be seen from Figure 5(b), higher resolution levels lead to a better performance in terms of $d_G$. The optimal value for $n$ depends on $r$. A generally good value is $n = 8$. However, for larger $r$, the number of lines $l$ should be set large, too. We conclude that $GSC_W$ can more easily be parameterized.

### C. Performance Analysis

Figure 6 shows the result of the comparison of our synopses with three other approaches. We depict the minimum and selected percentiles of the distribution of $d_G$ between $C_{ref}$ and the clusterings that resulted from cuts on all levels of the hierarchical clustering structure. Figure 6 contains $d_G$ for a range of parameter combinations for each tested synopsis.

This statistic has the following important characteristic. The maximum $d_G$ is equal for all synopses (and was excluded from the figure to make the other percentiles more readable). The minimum $d_G$ provides us with an indicator of how good a clustering can be when optimally parameterized. The selected percentiles (10%, 25%, 50%, 75%) show how bad the given percentage of the results became at maximum. They are an indicator of how a not optimally parameterized clustering performs.

From Figure 6, we draw the following conclusions: The direct application of the Euclidean distance ($ED$) is clearly the worst approach. Not even the best achievable clustering came to resemble $C_{ref}$. This confirms our expectations, since this approach is not rotation-invariant. The color histogram ($CH$) performs better than $ED$ but is still not usable at all. $LBP$ shows a distinctly better performance than $ED$ and $CH$, but even the best results still exhibit a large distance to the $C_{ref}$. $GSC_{AMS}$ is the best approach in terms of minimum $d_G$, only beating $GSC_W$ by a margin. The 10th and 25th percentile are better than any other approach except $GSC_W$. This shows that non-optimal parameterizations can still lead to good results. $GSC_W$ is similar to $GSC_{AMS}$ in performance, reaching nearly identical results for minimum $d_G$ and the 10th and 25th percentile.

In summary, $GSC_{AMS}$ and $GSC_W$ are able to create synopses that capture mapscape similarity, since under a good parameterization, these clusterings resemble the reference set very closely. The other approaches are of limited utility.

## V. Conclusion

Clustering of measurement maps can help improve the control of production processes. In this paper, we presented the global slope change $GSC$ synopses. The basic $GSC_A$ approximates a map using the average slope along a number of lines from the rotational center to the borders of the map. We refined $GSC_A$ in $GSC_{AMS}$ and $GSC_W$, where the lines are either approximated by the slope of several segments or by wavelet coefficients. Both approaches are able to yield excellent results. However, $GSC_W$ was easier to parameterize than $GSC_{AMS}$ and should therefore be preferred.

In the future, we will examine whether the problem of choosing suitable parameters can be tackled by a multi-level synopsis. The idea is to store $GSC_W$ or $GSC_{AMS}$ synopses for a range of parameter combinations. Since selecting the optimal parameterization is not possible without a reference set, the challenge lies in designing a tailored distance function that compares the multi-level synopsis and achieves good results without a-priori knowledge.

Another topic of investigation is the extension of the current approach to higher-dimensional measurement maps, e.g. where the measure depends on three spatial dimensions.

## References

[1] R. Datta, J. Li, and J. Z. Wang, "Content-based image retrieval: approaches and trends of the new age."

[2] A. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain, "Content-based image retrieval at the end of the early years," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 22, no. 12, pp. 1349–1380, Dec 2000.

[3] R. C. Veltkamp and M. Tanase, "Content-based image retrieval systems: A survey," 2000.

[4] S. Loncaric, "A survey of shape analysis techniques," *Pattern Recognition*, vol. 31, pp. 983–1001, 1998.

[5] T. Ojala, M. Pietikäinen, and T. Mäenpää, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 7, pp. 971–987, 2002.

[6] M. J. Swain and D. H. Ballard, "Color indexing," *International Journal of Computer Vision*, vol. 7, no. 1, pp. 11–32, Nov. 1991.

[7] J. Han and M. Kamber, *Data Mining: Concepts and Techniques*, 2nd ed. Berlin: Morgan Kaufmann, 2006.

[8] G. Nason, *Wavelet Methods in Statistics with R*. Berlin: Springer, 2008.

[9] A. Gionis, H. Mannila, and P. Tsaparas, "Clustering aggregation."