Robust pointwise correspondences for point cloud based deformation monitoring of natural scenes

Zan Gojcic[1], Caifa Zhou[1], Andreas Wieser[1]

[1] Institute of Geodesy and Photogrammetry, ETH Zürich, Stefano-Franscini-Platz 5, 8093 Zurich, Switzerland
{zan.gojcic, caifa.zhou, andreas.wieser}@geod.baug.ethz.ch

Key words: terrestrial laser scanning, deformation monitoring, deep learning, outlier detection, supervoxel

ABSTRACT

Areal-based deformation monitoring based on point clouds can be a very valuable alternative to the established point-based monitoring. However, due to naively establishing the pointwise correspondences, established deformation analysis approaches for point clouds do not expose the true 3D changes in parts, which actually did change. Herein we extend the recently proposed algorithms that establish pointwise correspondences in the feature space, with a neural network based outlier detection algorithm capable of classifying the putative pointwise correspondences into inliers and outliers based on information only extracted from the point clouds. We demonstrate the proposed approach on two data sets, including a real case data set of a landslide located in the Swiss Alps. We show that while the traditional approaches greatly underestimate the magnitude of the displacements, our approach can correctly estimate the true 3D displacement vectors.

I. Introduction

Despite the increasing use of point clouds to detect and quantify displacement and deformation of man-made and natural structures, several challenges regarding the point cloud based deformation analysis remain unsolved (Holst and Kuhlmann, 2016). In particular, this includes the estimation of 3D displacement vector fields, parameterization of deformations and quantification of error probabilities like false alarm rate or probability of missed detection. These challenges are particular demanding for point cloud data of natural environments, because in two epochs, identical points are either not acquired or cannot be readily identified. Due to the lack of regular structures and smooth objects, which could be represented with geometric primitives or free-form shapes, the deformation analysis of natural scenes is predominantly based on point cloud and surface based deformation models (Neuner *et al.* 2016, Wunderlich *et al.* 2016).

The point based deformation models, traditionally represented by the cloud-to-cloud (C2C) and multiscale model-to-model cloud (M3C2) (Lague *et al.* 2013) methods, can be used to compare the point clouds directly. On the other hand, surface based models such as cloud-to-mesh (C2M) and mesh-to-mesh (M2M), require that either one (C2M) or both (M2M) point clouds are first triangulated and then the resulting meshes are compared. The estimation of the displacement magnitudes or vectors in the above-mentioned methods mostly differs in how correspondences among the points are determined. In the C2C method, the correspondence is established by selecting the respective nearest point from the other epoch. Differently, other methods try to incorporate the local geometric information by constraining the search for corresponding points in the direction of the normal vector of either the underlying surface (C2M and M2M) or a plane fitted to the neighboring points (M3C2). These naïve ways of establishing the correspondences typically result in underestimation of the displacement magnitude in the parts of the point clouds that have changed (see Section III). Furthermore, as noted by Holst *et al.* (2017) the point cloud based and surface based models are incapable of correctly detecting and estimating the in-plane deformations and rigid body motion. A detailed explanation of point cloud based and surface based deformation models is available in Holst *et al.* (2017).

Deformation analysis based on point cloud data is not the only application that requires establishing the corresponding points in two or more epochs. In fact, the correspondences determined using local feature descriptors are regularly used, e.g. to estimate the coarse registration parameters. Additionally, Wagner *et al.* (2017) and Gojcic *et al.* (2018) already proposed the use of local feature descriptors for establishing correspondences with the goal of deformation analysis. However, Wagner et al. (2017) require RGB data together with the point clouds, and actually establish the correspondences in the RGB-D space. Furthermore, both have only tested their approaches on toy examples, which do not represent a real use case.

Herein, we make a step further and propose an algorithm that takes a set of putative correspondences, established using a local feature descriptor as input and performs a classification into inlier and outlier correspondences. This classification is then used to establish a robustly estimated displacement vector field, which denotes a 3D vector field of displacements for points, for which reliable correspondences were established. Finally, we show on two different data

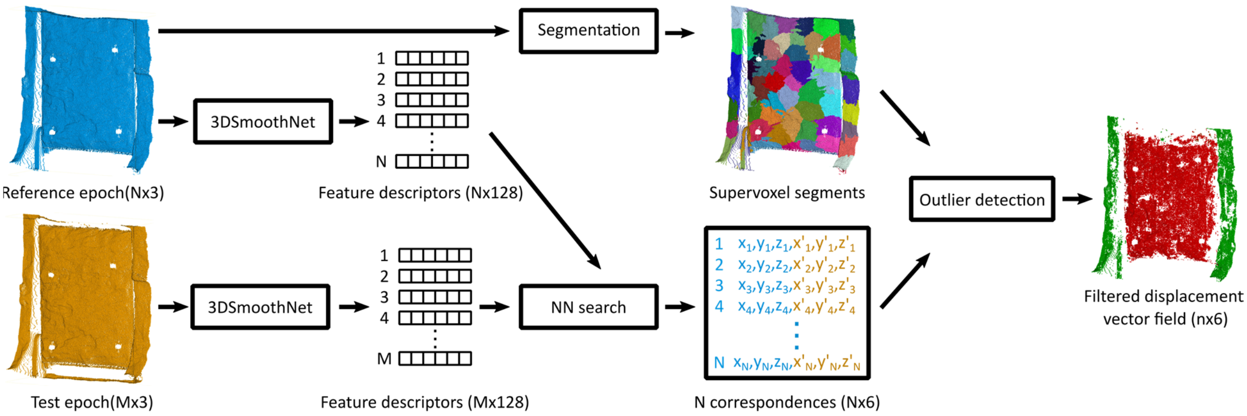

Segmentation
3DSmoothNet
1
2
3
4
N
Reference epoch(Nx3)
Feature descriptors (Nx128)
Supervoxel segments
Outlier detection
Filtered displacement vector field (nx6)
Test epoch(Mx3)
3DSmoothNet
1
2
3
4
M
Feature descriptors (Mx128)
NN search
1 $x_1,y_1,z_1,x'_1,y'_1,z'_1$
2 $x_2,y_2,z_2,x'_2,y'_2,z'_2$
3 $x_3,y_3,z_3,x'_3,y'_3,z'_3$
4 $x_4,y_4,z_4,x'_4,y'_4,z'_4$
N $x_N,y_N,z_N,x'_N,y'_N,z'_N$
N correspondences (Nx6)

Figure 1: Proposed pipeline for robust estimation of a dense displacement vector field from point cloud data. In the first step, feature descriptors for each point in both point clouds are computed using 3DSmoothNet and are used to establish the putative set of correspondences. Independently, the reference epoch is segmented into supervoxels. The correspondences are then grouped according to the supervoxels and filtered using the proposed outlier detection algorithm. The output of the pipeline is a robustly estimated displacement vector field. Points are colored based on the displacement magnitude (green stable, red moved) but the algorithm outputs the real 3D displacement vectors.

sets that the proposed pipeline, which relies solely on the geometric information intrinsically available in point clouds, is able to correctly estimate the real 3D displacement vectors, even for a real use case point clouds of a landslide in the Swiss Alps.

II. Method

This section describes the proposed pipeline for estimating a dense displacement vector field using point cloud data. The pipeline combines a local feature descriptor (3DSmoothNet), with a supervoxel segmentation algorithm and a novel neural network based outlier detection algorithm, with the goal of establishing true correspondences between points of two or more epochs (Figure 1).

A. *3DSmoothNet – 3D local feature descriptor*

We have recently proposed 3DSmoothNet, a deep learning based 3D local feature descriptor, which has low output dimension, high descriptiveness, and is fully rotation invariant. With 3DSmoothNet, we combine the traditional components of handcrafted descriptors, such as the estimation of the local reference frame for achieving rotation invariance, with the novel smoothed density value voxelization that is amenable to fully convolutional layers of standard deep learning libraries. 3DSmoothNet is a non-linear, learned function that maps the input, i.e. voxelized spherical neighborhood of the point, to a low dimensional feature vector for each point, by solely exploiting the local geometrical properties of the data.

Machine learning and especially deep learning algorithms typically need a large amount of annotated data. Specifically, to train 3DSmoothNet a large data set, consisting of overlapping point clouds with ground truth correspondences was required. Gathering training data of outdoor scenes including the ground truth correspondences for a typical geomonitoring scenario would have been unfeasible. Therefore, we resorted to an indoor benchmark data set denoted as 3DMatch (Zeng *et al*., 2017), which is an RGB-D data set consisting of 62 real-world indoor scenes ranging from offices and hotel rooms to tabletops and restrooms. In Gojcic *et al*. (2019b) we show that the 3DSmoothNet trained only using these indoor RGB-D data can generalize to outdoor point clouds acquired using a laser scanner, without any fine-tuning.

As 3DSmoothNet significantly outperforms the state-of-the-art on indoor as well as outdoor data (Gojcic *et al*., 2019b), we use it hereinafter to infer pointwise local descriptors. A more detailed description of 3DSmoothNet is beyond the scope of this paper and the interested reader is referred to Gojcic *et al.*, (2019b) for more information.

B. *ANN based outlier detection algorithm*

In order to obtain a dense displacement vector field, we do not compute feature descriptors only for some selected points, i.e. keypoints, but rather for each point in the point clouds of both epochs. Under the assumption that for each point from the reference epoch, there is a corresponding point in the test epoch and that correspondence is properly measured by the closeness in the feature space, the displacement vector field can be established by a nearest neighbor search. However, due to the sampling process yielding the point clouds, 6-DOF motion and occlusions, this assumption does not (always) hold. In particular, we also expect that some parts of the area may be deformed too much and may not be recognizable anymore. This, in conjunction with the false correspondences due to repeating structures, changes in point cloud density and noise of the point clouds, results in an initial set of correspondences, which is very noisy and typically contains outliers.

We therefore propose a binary classification algorithm to identify the putative correspondences as inliers or outliers. The algorithm is based on the local

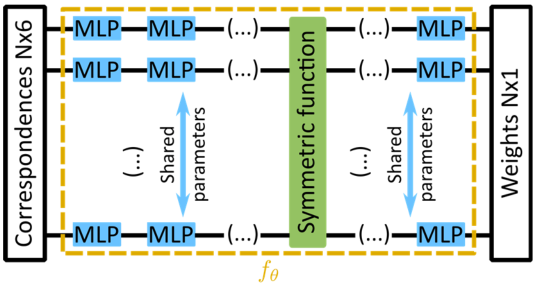


Figure 2: The proposed ANN based filtering algorithm is composed of several MLPs operating on individual correspondences and symmetric functions aggregating local information across the whole input. The parameters of MLPs are shared for all correspondences. In the inference phase, the branches are simply copied to fit the number of points in the input

consistency assumption, i.e. locally the point clouds are assumed to represent a rigid body rather than being significantly deformed. This assumption is also crucial for the good performance of the local feature descriptors. More formally, consider two point clouds $\mathbf{P} \in \mathbb{R}^{N\times 3}$ and $\mathbf{Q} \in \mathbb{R}^{M\times 3}$, representing the test and reference epoch respectively. Let $(\mathbf{P})_i = \mathbf{p}_i = [x_i, y_i, z_i]$ and $(\mathbf{Q})_j = \mathbf{q}_j = [x'_j, y'_j, z'_j]$ represent the coordinate vectors of individual points of the point cloud. If each point of the point cloud $\mathbf{P}$ is matched to its nearest neighbor in the point cloud $\mathbf{Q}$ based on the descriptor distance, a group of N correspondences $\mathbf{c}_i$ is obtained and can be written in matrix form as

$$\begin{aligned} \mathbf{X} &= [\mathbf{c}_1; \cdots; \mathbf{c}_N], \\ \mathbf{c}_i &= [x_i, y_i, z_i, x'_i, y'_i, z'_i], \end{aligned} \tag{1}$$

The matrix $\mathbf{X} \in \mathbb{R}^{n\times 6}$, containing only coordinates of the putative correspondences, represents the input to the filtering algorithm f_θ, which maps $\mathbf{X}$ to a vector of weights

$$\mathbf{w} = \left[w_1, \cdots, w_N\right] \tag{2}$$

where $w_i \in \left[0,1\right]$, with $w_i = 0$ indicating (strongly) that the correspondence $\mathbf{c}_i$ is an outlier and $w_i = 1$ that it is an inlier. We approximate f_θ using an artificial neural network (ANN). Because of the unordered nature of point clouds, the order of the correspondences is arbitrary and permuting the rows of $\mathbf{X}$ should result in the equivalent permutation of the weights $\mathbf{w} = f_\theta(\mathbf{X})$. To achieve the invariance to input permutation the network architecture, adapted from Yi *et al.* (2018), is a composition of Multi-Layer Perceptrons (MLPs), which operate independently on each individual correspondence. Because of this, the individual branches of the network do not obtain the information about the neighboring points explicitly and the local context is implicitly established using suitable symmetric functions, e.g. mean value and standard deviation, connecting the individual layers of the network (see Figure 2).

Loss function In order to optimize the parameters of the ANN, we minimize the hybrid loss function

$$\mathcal{L}(\theta, \mathbf{X}) = \sum_{k=1}^{B} \alpha \mathcal{L}_c(\theta, \mathbf{X}_k) + \beta \mathcal{L}_t(\theta, \mathbf{X}_k) \tag{3}$$

where $\mathcal{L}_c$ is the classification loss and $\mathcal{L}_t$ is the transformation loss. θ denotes the parameters of the network and $\mathbf{X}_k$ represents a set of putative correspondences for point cloud pair k in a mini-batch with B[1] training examples. The contribution of both loss functions is controlled using the hyper-parameters α and β.

The classification loss penalizes both types of error, the false positives as well as false negatives. Given N putative correspondences $\mathbf{c}_i$ with the corresponding ground truth labels y_i it is defined as a binary cross entropy function

$$\mathcal{L}_c = \frac{1}{N} \sum_{i=1}^{N} -y_i \log(h(\mathbf{X}_k)_i) - (1 - y_i) \log(1 - h(\mathbf{X}_k)_i), \tag{4}$$

where h is a sigmoid function

$$h(\mathbf{X}_k) = \frac{1}{1 + \exp(-f_\theta(\mathbf{X}_k))}. \tag{5}$$

Minimizing the classification loss proves robust but can still let some outliers undetected (Yi *et al.* 2018**.** Therefore, additional supervision—supported by the rigid body assumption—is introduced by penalizing the deviations from the ground truth rotation matrix $\mathbf{R}$ as

$$\mathcal{L}_t = \left\| \mathbf{R} - g(\mathbf{X}, \mathbf{w}) \right\|_F, \tag{6}$$

where g is a function, which estimates the rotation matrix based on the singular value decomposition of the weighted covariance matrix $\mathbf{\Sigma}_\mathbf{X} = \mathbf{X}_1^T \mathbf{W} \mathbf{X}_2$, with $\mathbf{X}_1 = (\mathbf{X}_{ij})_{\substack{1\le i\le N \\ 1\le j\le 3}}$, $\mathbf{X}_2 = (\mathbf{X}_{ij})_{\substack{1\le i\le N \\ 4\le j\le 6}}$ and $\mathbf{W} = \mathrm{diag}(\mathbf{w})$ (Sorkine-Hornung *et al.* 2017). $\left\| \bullet \right\|_F$ is the Frobenius norm.

Optimization We optimize the parameters of the network using a variant of gradient descent (Kingma and Ba, 2015) in combination with the same benchmark data set that is used for the training of 3DSmoothNet, i.e. the 3DMatch data set (Zeng *et al.*, 2017). We have empirically discovered that adding the transformation loss from the start can actually harm the convergence. Therefore, we start training by setting $\alpha = 1$ and $\beta = 0$. In a later stage of training, when the network can already correctly classify the majority of the correspondences we change β to 0.1, which enables additional improvement of the performance.

C. *Oversegmentation using the supervoxel approach*

In typical geomonitoring applications, parts of the repeatedly scanned scene may be stable over time,

[1] Herein, a point cloud pair k is represented by all correspondences belonging to a single supervoxel, and a mini-batch denotes a group of supervoxels that fit into the memory of a GPU.

while others change due to the flow of earth and debris. Individual objects may be large enough to be detected as translated and rotated rigid bodies (Gojcic *et al.* 2018). This causes discontinuities in the ground-truth displacement vector field. In order not to violate the local consistency assumption, the point clouds can therefore not be analyzed as one object, but rather have to be segmented into parts that do not cross the discontinuities. The discontinuities of the displacement vector field are not know beforehand, but they predominantly appear at the boundaries of larger objects. We therefore use a segmentation algorithm with boundary preservation (Lin *et al.* 2018) to segment the point cloud of the reference epoch into segments, denoted as supervoxels. The segments are deliberately allowed to be smaller than the expected actual objects. This oversegmentation is preferred to object segmentation because it allows the apparent geometrical changes within an object to be larger than the changes between objects. With supervoxels, only small, similar segments are clustered together thus enabling better boundary preservation. The supervoxels are obtained by minimizing the energy function

$$E(\mathbf{Z})=\sum_{i=1}^{N}\sum_{j=1}^{N} z_{ij}\cdot d(\mathbf{p}_i,\mathbf{p}_j)+\lambda\left|C(\mathbf{Z})-n_s\right| \qquad (7)$$

where $z_{ij}\in\{0,1\}$ with $z_{ij}=1$ if the point $\mathbf{p}_i$ is a representative point of a supervoxel and $\mathbf{p}_j$ belongs to the same supervoxel. λ is the weighting factor, and $C(\mathbf{Z})$ is a function that counts the current number of supervoxels. The boundary preservation is achieved by incorporating the cosine similarity along with the traditional Euclidean distance in the similarity measure

$$d(\mathbf{p}_i,\mathbf{p}_j)=1-\left|\mathbf{n}_{\mathbf{p}_i}\cdot\mathbf{n}_{\mathbf{p}_j}\right|+0.4\frac{\left\|\mathbf{p}_i-\mathbf{p}_j\right\|_2}{r} \qquad (8)$$

where $\mathbf{n}_{\mathbf{p}_i}$ and $\mathbf{n}_{\mathbf{p}_j}$ are the normal vectors of the points $\mathbf{p}_i$ and $\mathbf{p}_j$ respectively and r denotes the approximate size of the supervoxels in terms of a radius, which indirectly sets the approximate number of segments n_s.

Herein, the oversegmentation is performed only for the reference epoch and after the putative correspondences have been determined using the whole point clouds. Following the segmentation, each supervoxel is fed to the ANN based filtering algorithm as an individual example, thus satisfying the local consistency assumption. Finally, the putative correspondences are filtered by rejecting all the correspondences with w_i strictly smaller than the classification threshold $\tau_c=0.5$ and the remaining correspondences are concatenated to form a single point cloud[2]. Such a robustly estimated dense displacement vector field can be used as the final output or as an input to an additional smoothing/interpolation algorithm such as the one given in Gojcic *et al.* (2019a).

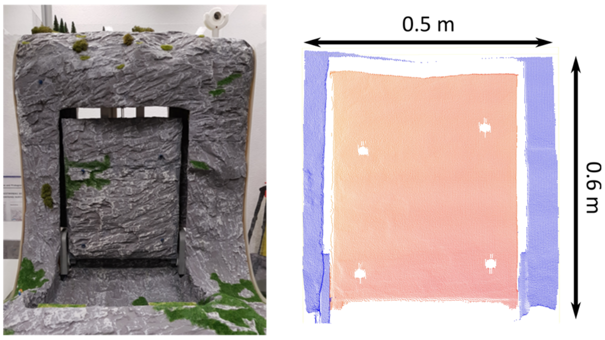


Figure 3: Rockfall simulator (left) and a point cloud acquired using a Leica MS50 (right). Red color shows the moving part and the blue color the stable part of the simulator. Holes in the red part of the point cloud show the location of the four mini prisms.

III. Experiments

In this section, we analyze the performance of the proposed algorithm based on experiments conducted with two data sets. We start with a detailed analysis on a data set of a rockfall simulator, acquired in a controlled environment with per point ground truth, before evaluating the generalization capacity on a real case data set of a landslide located in the Alps. As baseline algorithms, we use the established C2C, C2M and M3C2 algorithms implemented in the open source software CloudCompare[3].

A. *Rockfall simulator*

The rockfall simulator (Figure 3) is a piece of hardware composed by a rigid frame and a part that can rigidly translate vertically and rotate about a horizontal axis by computer control. The surfaces have a similar structure as rocks. The simulator allows mimicking a rockfall and was originally built for educational purposes. The moving part of the simulator (see Figure 3 right) is equipped with four mini prisms that can be used to establish the ground truth. Herein, we consider a scenario in which the moving part is displaced vertically by about 3 cm. The rockfall simulator was scanned from a distance of about five meters in two epochs using a Leica MS50. The mean resolution of the point clouds is 3 mm. The ground truth transformation parameters for the moving part of the simulator were estimated from high precision polar coordinate measurements to the four mini prisms using the same MS50.

[2] Empirically, for more than 95% of the putative correspondences, the $w_i<0.1$ or $w_i>0.9$ in the rockfall simulator example. Therefore, we do not optimize the classification threshold and use the threshold of $\tau_c=0.5$ herein, if not explicitly specified differently.

[3] https://www.danielgm.net/cc/

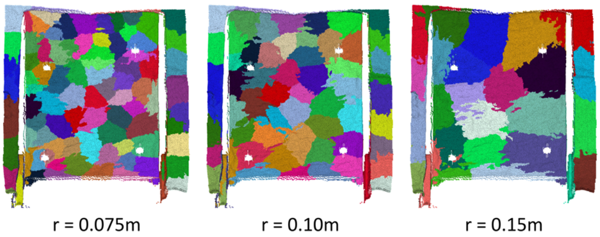


Figure 4: Results of the supervoxel segmentation of the rockfall simulator point clouds in relation to r. When r is too big the supervoxels cross the object boundaries (see for example the top right corner of the right figure). Colors of the supervoxels are selected randomly.

Supervoxel segmentation. The size of the supervoxels, and indirectly also their approximate number, is controlled with the hyper-parameter r of eq. (8). According to Lin *et al* (2018), the object boundaries can be preserved even if the distances between boundaries are smaller than the selected value of r. We evaluate the supervoxel segmentation algorithm on the reference epoch point cloud of the rockfall simulator (Figure 4).

This data set is challenging for the supervoxel segmentation algorithm because individual parts of the simulator are almost parallel (stable and moving parts) and contribute little to the cosine similarity of the normal vectors in the dissimilarity measure (eq. (8)). Therefore, the boundaries can only be partially preserved, when r gets too big (see Figure 4 right). Based on this qualitative result, we use $r = 0.1\,\mathrm{m}$ for the rockfall simulator experiments presented herein.

ANN based outlier detection. The results of the proposed ANN based outlier detection algorithm are depicted in Figure 5. By directly using the model trained on the indoor scenes, more than 94% of the points can be classified correctly as outliers and inliers, while less than 5% are false negatives and about 1% false positives. Here, a false negative means that the ground truth label indicates an inlier i.e., the 3DSmoothNet correspondence is actually correct, but the algorithm classifies the correspondence as wrong. Conversely, a false positive denotes that the 3DSmoothNet correspondence is wrong but the algorithm classifies the correspondence as correct. The ground truth labels for the correspondences were established by considering the deviations between the putative and the ground truth corresponding point. If the deviation was less than 7.5 mm (2.5 times the mean resolution), the correspondence was labeled as an inlier and otherwise as an outlier.

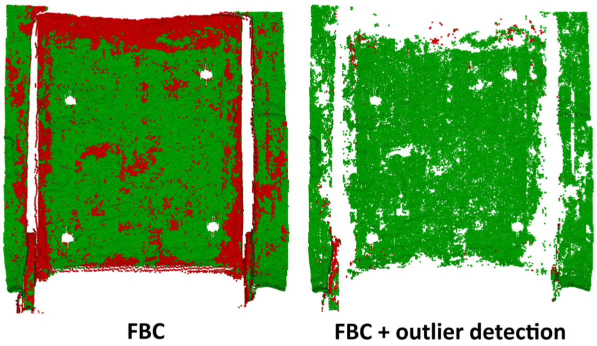


Figure 5: Results of the correspondence search in the feature space, before (left) and after (right) the outlier filtering with the proposed ANN based algorithm. Red color denotes the outlier and the green color the inlier correspondences.

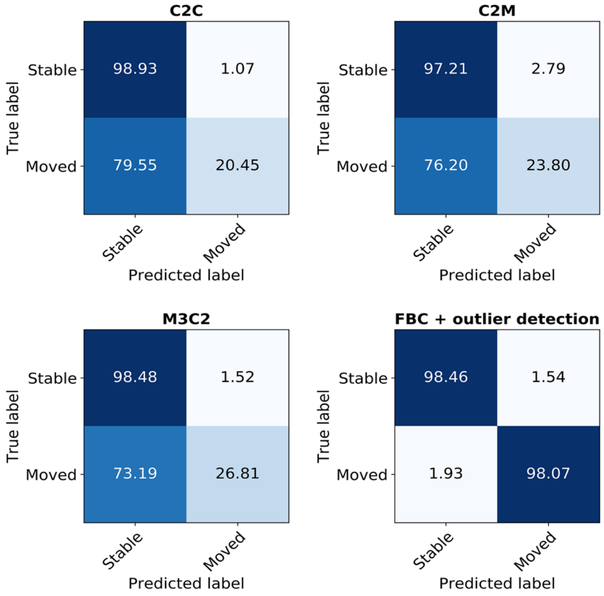


Figure 6: Confusion matrix. We use the algorithms to classify the points of the rockfall simulator into two classes: moved and stable. While the performance of all algorithms for ground truth class stable is comparable, FBC + outlier detection algorithm clearly outperforms the baselines for the ground truth class moved. Results are shown in percentage.

Pointwise classification into stable and moved parts. We start the comparison to the baseline algorithms by evaluating the capability of the algorithms to classify the point clouds into moved and stable parts. The ground truth labels are defined manually and can be seen in Figure 3 right. For the C2C, C2M and MC32 analysis we compute the displacements using the implementation available in CloudCompare and infer the labels based on the magnitude of the displacements. Specifically, all the points for which the estimated displacement is larger than 7.5 mm (2.5 times the point cloud resolution) are classified as moved and the rest are classified as stable. We compare this results to the feature based correspondences (FBC) established using 3DSmoothNet, followed by the proposed ANN based outlier detection algorithm. After the outlier detection, only a subset of points (inliers) are remaining and can be classified as stable or moved based on the same threshold as above. To ensure a fair comparison, we infer the labels for the remaining points (outliers) based on the majority voting inside individual supervoxels. The results are depicted in Figure 6. Whereas all the algorithms achieve high accuracy for the ground truth class "stable", the baselines fail in classifying the moved areas correctly. Indeed, they classify more than 70% of the moved points as stable. On the other hand, FBC + outlier detection

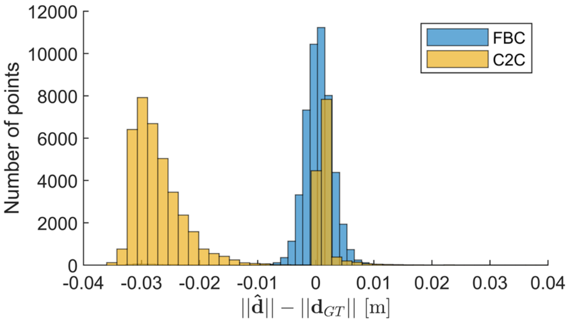


Figure 7: Deviations from the ground truth. For clarity, we show only the results of the FBC and C2C. C2M and M3C2 deviations also show a multimodal distribution similar to the one of the C2C.

algorithm achieves more than 98% accuracy for both classes and its performance is equal for stable and moving parts.

Quantitative analysis of the displacements. The goal of the point cloud based deformation analysis is not only detecting which parts of the scenery have moved/changed and which have remained stable, but also the quantification of the displacements. Because some of the baseline algorithms output only the magnitude of the displacement and not a full 3D displacement vector, we use the residuals between the estimated and the ground truth displacement magnitude as the performance metric. For the proposed pipeline, only the correspondences that were classified as inliers (67.5% of all points) are considered in this analysis.

Figure 7 and Table 1 show the results of the quantitative analysis of the estimated displacements using the rockfall simulator point cloud data. The residuals of the FBC + outlier detection algorithm are normally distributed with the mean value of 0 mm and a standard deviation of 3 mm. On the other hand, the residuals of C2C show a multimodal distribution with one peak centered at 0 mm and one at approximately 3 cm, which corresponds to the actual displacement of the moving part, indicating that this displacement is not detected by the C2C analysis. Table 1 summarizes the results for all the methods, where the precision denotes the ratio between the number of correct correspondences (i.e. the residuals smaller than 7.5 mm) and the number of all correspondences. Similarly, recall denotes the ratio between the number of correct correspondences and the number of all points in the reference epoch.

For most of the methods the precision and recall are the same, because the correspondences established by them are correct only within the stable areas (because of the dominant in-surface component of the actual changes), and the percentage of these within the entire scene is about 26%. In fact, as also indicated by the results printed in Figure 6, the majority of the scenery is identified as stable using these standard algorithms, irrespective of the ground truth motion. Our method instead performs equally well in stable and non-stable areas and achieves a much higher precision and recall, indicating that more than 98% of the identified correspondences are correct. However, this is only possible because our approach rejects putative correspondences identified as outliers. Therefore, the recall cannot achieve 100% but rather indicates the percentage of the point cloud with sufficiently unique features.

Because our method outputs real 3D displacement vectors, we also include the analysis based on the distances between the estimated and the ground truth displacement vectors. Table 1 shows that not only our method reaches a significantly higher precision and recall, the small drop in performance between "Ours" and "Ours (vector distance)" highlights that it can also efficiently estimate the real 3D displacement vectors.

Table 1: Precision and recall of the estimated displacements in comparison to the ground truth, for the rockfall simulator point clouds.

	Precision [%]	**Recall [%]**
C2C	26.2	26.1
C2M	25.8	25.8
M3C2	26.5	25.8
Ours	98.8	66.4
Ours (vector distance)	98.4	66.1

B. *Real case landslide in the Alps*

The second data set used for demonstration herein consists of point clouds of a real landslide located in the Swiss Alps (Figure 8). The point clouds were acquired using a drone based LiDAR system (Riegl RiCOPTER) in two epochs about 2.5 months apart (July and September 2018). The point clouds of both epochs were georeferenced and the registration is hereinafter considered as error free. Based on the GPS and total station measurements, which are available for selected points on the edges of the landslide, displacements in the range of a couple of centimeters (bottom part) to a couple of decimeters (top part) were expected. Due to the large amount of data—the reference epoch has more than one billion points—we focus on two selected areas for which also a ground truth is available through the high precision total station measurements (Area 1/blue and Area2/red of Figure 8). Each of these areas contains approximately one million points per epoch.

Quantitative analysis of the displacements. We follow the same evaluation procedure as in the rockfall simulator case. Specifically, we compare our method to C2C, C2M and M3C2. Additionally we compare the results to the ground truth, which is available in form of a displacement magnitude for a single point in the middle of Area 1 and a single point in the vicinity of Area 2. As we analyze relatively small areas, we assume that the displacement magnitude of these ground truth points is representative for all the points within the respective area. For the supervoxel segmentation we use $r = 1.5\,\mathrm{m}$ for Area 1 and $r = 2\,\mathrm{m}$ for Area 2, which is approximately 30 times the resolution of the point clouds in the respective area and corresponds to the ratio used in the analysis of the rockfall simulator.

The results are depicted in Figure 9 and summarized in Table 2. When compared to the ground truth, traditional methods significantly underestimate the magnitude of the displacements. Even more, Table 2 shows that the estimated magnitude is rather dependent on the resolution of the point clouds, roughness of the surface and possibly the direction and type of the motion than on the actual displacement. On the other hand, our method can accurately estimate the displacement magnitudes with an error smaller than (Area 1) or close to (Area 2) 5% of the actual motion. This error corresponds to less than half of the mean resolution of the point clouds.

Outlier detection analysis. The lower resolution and higher noise of the point clouds, combined with the larger motion, presumably have a negative effect on the correspondence search and on the outlier detection algorithm. This results in a long tail of the distribution of the displacement magnitudes, especially for Area 2, which is to a large extent covered with small gravel with no distinct features. We therefore perform an additional analysis in which we assume that the ground truth displacement magnitude $\|\mathbf{d}_{\mathrm{GT}}\| = 0.414\,\mathrm{m}$ is representative for all the points in the Area 2. Assuming that the standard deviation of the estimated displacement magnitudes is approximately $\sigma_{\|\mathbf{d}\|} \approx 0.1\,\mathrm{m}$ for all the methods in Table 2, we label all correspondences $\mathbf{c}_i$ for which

$$\|\mathbf{d}_{\mathrm{GT}}\| - 3\sigma_{\|\mathbf{d}\|} \le \|\hat{\mathbf{d}}_{\mathbf{c}_i}\| \le \|\mathbf{d}_{\mathrm{GT}}\| + 3\sigma_{\|\mathbf{d}\|}$$

as correct and the rest as wrong.

Considering these labels, 65% of the correspondences established using 3DSmoothNet without outlier detection are wrong for Area 2. After the outlier detection with the approach proposed herein (using $\tau_c = 0.5$), more than 81% of the inliers are actually correct correspondences and less than 19% are wrong, with a recall of 33%. As shown in Figure 10 left, most of the false positives left after the outlier detection lie in the areas covered with gravel (left and right sides of the ridge).

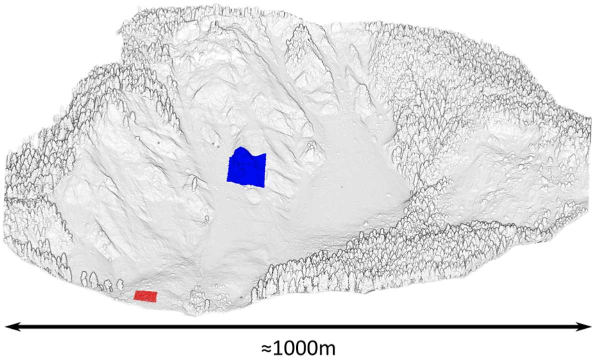


Figure 8: Reference point cloud of the landslide located in the Swiss Alps acquired using a drone based LiDAR system. We perform the deformation analysis for the areas marked with the red (Area 1) and blue (Area 2) color.

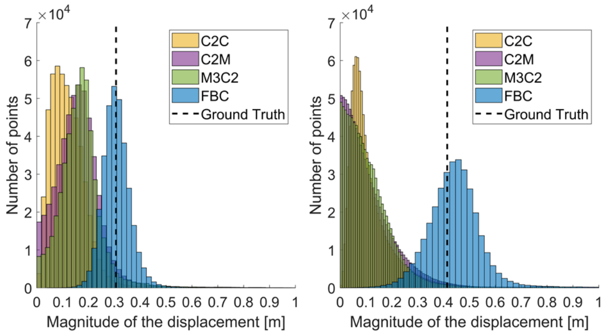


Figure 9: Histogram of the displacement magnitudes for Area 1 (left) and Area 2 (right). Traditional methods greatly underestimate the magnitude of the displacements.

Due to the relative large number of false positives, we analyze the effect of increasing τ_c to 1, i.e. accepting only the correspondences with very high confidence[4]. With this, the percentage of correct correspondences can be increased to 98.5% with a recall of 27.7% (Figure 10 right). This result indicates that a scene specific τ_c should be used and the appropriate choice is a topic for future research.

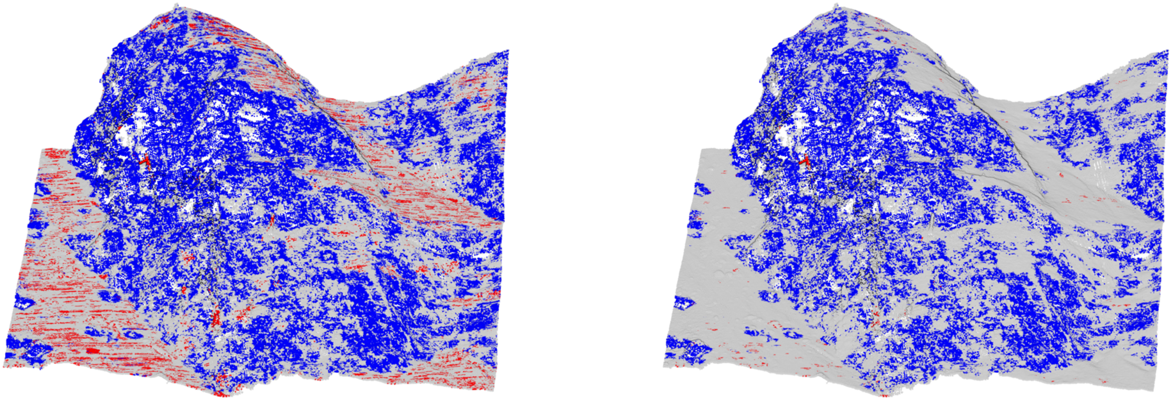
Table 2: Median displacement magnitude. The results of our method ($\tau_c = 0.5$) for Area 2 are slightly biased due to the long tail of the displacement magnitude distribution.

Method	Area 1	Area 2
	Displ. [m]	Displ. [m]
C2C	0.119	0.118
C2M	0.144	0.117
M3C2	0.160	0.113
Ours ($\tau_c = 0.5$)	0.307	0.464
Ours ($\tau_c = 1$)	0.304	0.439
Ground truth	0.306	0.414

IV. CONCLUSION

We have proposed a new pipeline for deformation analysis of natural scenes using point cloud data. We complement 3DSmoothNet with an outlier detection algorithm, which classifies the putative correspondences into inliers and outliers. Using two different data sets, we highlight the shortcomings of the traditional approaches, while showing that our approach can correctly estimate real 3D displacement vectors. In the future work, we will investigate the proper choice of

[4] To avoid the quantization errors we use $1 - 1 \cdot 10^{-5}$ in our code.



$\tau_c = 0.5$ $\tau_c = 1$

Figure 10: Point cloud of the Area 2 after the outlier detection algorithm using $\tau_c = 0.5$ (left) and $\tau_c = 1$ (right). Blue color denotes the true positives, red color the false positives and grey color the negatives. Note how the false positives mostly lie on the flat areas covered by gravel and small cobbles with no distinctive features.

the hyper-parameter for the outlier detection step, and the classification of the displacement vector field into rigidly displaced objects and deformed parts.

V. Acknowledgements

The point cloud data of the landslide in the Swiss Alps used in this paper were collected and georeferenced by ALTAMETRIS. We thank Nicolas Ackermann (Swiss Federal Railways) for providing these data. The Amt für Wald und Naturgefahren of Kanton Graubünden (Andreas Huwiler) has provided the ground truth data of the landslide in the Swiss Alps.

References

Gojcic, Z., C. Zhou and A. Wieser (2018). Learned compact local feature descriptor for TLS-based geodetic monitoring of natural outdoor scenes. In: ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Vol. IV, No. 2, pp. 113-120.

Gojcic, Z, C. Zhou and A. Wieser (2019a). Feature-based approaches for geomonitoring using terrestrial laser scanning point clouds. In: 1st Swiss Workshop on Machine Learning for Environmental and Geosciences (MLEG 2019), 2 p.

Gojcic, Z., C. Zhou and A. Wieser (2019b). The Perfect Match: 3D Point Cloud Matching with Smoothed Densities. In: IEEE Conference on Computer Vision and Patters Recognition (CVPR)

Holst, C., and H. Kuhlmann (2016). Challenges and present fields of action at laser scanner based deformation analyses. Journal of applied geodesy, Vol. 10, No. 1, pp. 17-25.

Holst, C., B. Schmitz and H. Kuhlmann (2017). Eignen sich in Standardsoftware implementierte Punktwolkenvergleiche zur flächenhaften Deformationsanalyse von Bauwerken? Eine Fallstudie anhand von Laserscans einer Holzplatte und einer Staumauer. Zeitschrift für Vermessungswesen (ZFV), Vol. 142, No. 2, pp. 98-110.

Kingma, D. P. and L. J. Ba (2015). Adam: A Method for Stochastic Optimization. In: International Conference on Learning Representations (ICLR), 13 p.

Lague D., N. Brodu, and J. Leroux (2013). Accurate 3D comparison of complex topography with terrestrial laser scanner: Application to the Rangitikei canyon (N-Z). ISPRS Journal of Photogrammetry and Remote Sensing, Vol. 82, pp. 10-26.

Lin, Y., C. Wang, D. Zhai, W. Li, and J. Li (2018). Toward better boundary preserved supervoxel segmentation for 3D point clouds. ISPRS journal of photogrammetry and remote sensing, Vol. 143, pp. 39-47.

Neuner, H., C. Holst, and H. Kuhlmann (2016). Overview on current modelling strategies of point clouds for deformation analysis. Allgemeine Vermessungs Nachrichten (AVN), Vol. 123, pp. 328-339.

Papon, J., A. Abramov, M. Schoeler and F. Worgotter (2013). Voxel cloud connectivity segmentation-supervoxels for point clouds. In IEEE Conference on Computer Vision and Patters Recognition (CVPR), pp. 2027-2034.

Qi, C. R., H. Su, K. Mo and L. J. Guibas (2017). Pointnet: Deep learning on point sets for 3d classification and segmentation. In: IEEE Conference on Computer Vision and Patters Recognition (CVPR), pp. 652-660.

Sorkine-Hornung, O., and Rabinovich, M. (2017). Least-squares rigid motion using svd. Technical report, 5 p.

Yi, K. M., E. Trulls, Y. Ono, V. Lepetit, M. Salzmann and P. Fua (2018). Learning to find good correspondences. In: IEEE Conference on Computer Vision and Patters Recognition (CVPR), 8 p.

Wagner, A., W., Wiedemann and T. Wunderlich (2017). Fusion of laser-scan and image data for deformation monitoring–Concept and perspective. In: Proceedings of the 7th International Conference on Engineering Surveying (INGEO 2017), pp. 157-164.

Wunderlich, T., W. Niemeier, D. Wujanz, C. Holst, F. Neitzel and H. Khulmann (2016). Areal Deformation Analysis from TLS Point Clouds – The Challenge. Allgemeine Vermessungs Nachrichten (AVN), Vol. 123, pp. 340-351.

Zeng, A., S. Song, M. Nießner, M. Fisher, J. Xiao, and T. Funkhouser (2017). 3DMatch: Learning Local Geometric Descriptors from RGB-D Reconstructions. In: IEEE Conference on Computer Vision and Patters Recognition (CVPR), pp. 1802-1811.