A Geometry-based Method for the Spatio-temporal Detection of Cracks in 4D-Reconstructions

Carl Matthes Adrian Kreskowski Bernd Froehlich

Virtual Reality and Visualization Research Group
Bauhaus-Universität Weimar

ABSTRACT

We present a novel geometry-based approach for the detection of small-scale cracks in a temporal series of 3D-reconstructions of concrete objects such as pillars and beams of bridges and other infrastructure. The detection algorithm relies on a geometry-derived coloration of the 3D surfaces for computing the optical flow between time steps. Our filtering technique identifies cracks based on motion discontinuities in the local crack neighborhood. This approach avoids using the material color which is likely to change over time due to weathering and other environmental influences. In addition, we detect and exclude regions with significant local changes in geometry over time e.g. due to vegetation. We verified our method with reconstructions of a horizontal concrete beam under increasing vertical load at the center. For this case, where the main crack direction is known and a precise registration of the beam geometries over time exists, this approach produces accurate crack detection regardless of substantial color variations and is also able to mask out regions with simulated growth of vegetation over time.

1 INTRODUCTION

A timely, accurate and reliable detection of damages of built infrastructure is essential to prevent potential failure and increased costs for maintenance and repair. While monitoring of buildings, bridges and other infrastructure objects by various acquisition and 3D reconstruction technologies is slowly becoming feasible (e.g. [1] [16]), the algorithmic detection of damages such as small-scale cracks is difficult due to changes of the radiometric properties of surfaces under varying lighting conditions, weathering and other environmental influences.

In this work, we developed a crack detection pipeline that relies on at least two 3D-reconstructions of a real-world object from two different points in time. In a first step, we replace the radiometric information of the surfaces by a geometry-derived color, which is based on a local geometric feature invariant under transformations as well as large-scale non-rigid deformations. We establish correspondences between the two time steps through means of optical flow [12] and search for motion discontinuities which may correspond to cracks. We also developed a novel geometric descriptor which helps to identify incorrect correspondences between time steps and allows us to exclude regions that are subject to geometric occlusions e.g. by vegetation. In our current implementation, a precise registration of the geometry over time is required and the main crack direction has to be known or manually found.

Our work is inspired by Chaudhury et al. [8] who suggested to find cracks by exploiting local discontinuities in 2D optical flow along the crack line. Their approach relies on a sequence of images that are acquired under constant lighting conditions and, in their use case, in rapid succession. It is our goal to develop an approach for outdoor environments where radiometric variations of the material due to weathering and lighting conditions as well as occlusions caused by the growth of vegetation are common side effects of longer time spans between acquisitions. In related work, the often image-based crack detection approaches commonly attempt to reduce the amount of false positive matches arising from these real-world influences by a posteriori filtering or connected component analysis. However, substantial detection disturbances cannot easily be removed in post-processing and impose obstacles on the applicability of image-based methods. Therefore, we argue that in many cases the color of the material is not ideally suited for detecting small-scale cracks.

Our geometry-based crack detection approach is centered around the following contributions:

- the development of a local geometric feature to be used in optical flow computations that performs similarly well as the surface color of 3D objects made from concrete if no radiometric changes are present but remains to enable the robust detection of correspondences with radiometric variations when color-based optical flow estimations fail
- an efficient verification of uncertain correspondences in 3D optical flow caused by occlusions using a novel SSIM-based geometric descriptor for point clouds
- a sparse voxel octree-based filtering approach for the detection of small-scale cracks which operates on material displacement, rather than the color offset of the cracks themselves

We integrated all proposed techniques into a framework for the spatio-temporal detection of cracks in a temporal series of large point clouds and demonstrated their effectiveness with 3D-reconstructions of a horizontal concrete beam under increasing vertical load at the center.

2 RELATED WORK

With few exceptions, the detection of cracks has been driven by image-based methods in the past. Combinations of percolation [28] [27], genetic programming [17] and, more recently, machine learning [7] [9] have been applied successfully to detect and measure minuscule-scale cracks in single images. Advances have been made to produce crack detection from image sequences. Benning et al. [3] first proposed to compute deformation of material under stress using photogrammetry. Since then, a number of researchers employed digital image correlation (DIC) to detect micro-scale cracks in 2D displacement fields [6] [13] [20] [22].

More recently, Chaudhury et al. [8] suggested the detection of early stage cracks in image sequences by filtering for discontinuities in optical flow. Similar to DIC-based methods, their approach relies on the surface color of concrete material and is not ideally suited for outdoor environments, when radiometric variations over time other than those along cracks are significant. In addition, potential occlusions over time in photo sequences can lead to an unreliable image-based detection of cracks and were left unattended in previous work.

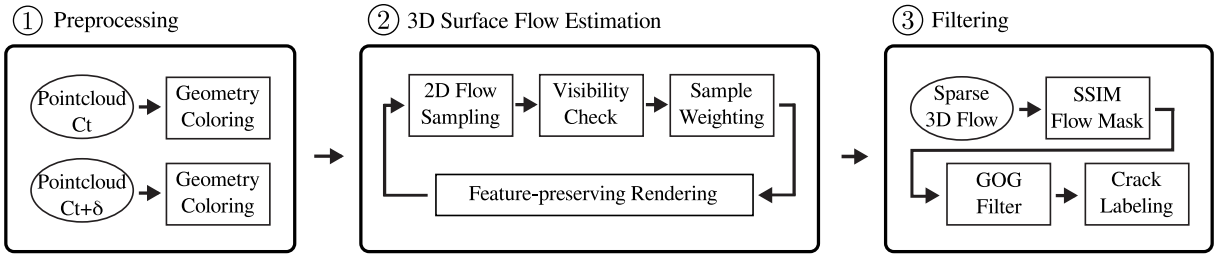


Figure 1: Overview of our proposed crack detection pipeline.

Detection disturbances caused by uneven illumination and shadows in images have been addressed in previous work. Qu et al. [21] eliminate detection noise through percolation processing of images while Li et al. [15] improve detection performance by predicting per-pixel crack probabilities through convolutional neural networks. We propose a distinct and novel approach to perform the crack detection, which is particularly robust against variations of material color, shadows and illumination.

We establish correspondences between two point clouds through means of optical flow. Optical flow estimation is a technique for computing relative motion in image sequences. Especially global methods [12] are useful for the dense and accurate determination of non-rigid displacements. We utilize a GPU-accelerated implementation of a coarse-to-fine variational optical flow algorithm [5] and store the resulting motion information in an auxiliary data structure based on sparse voxel octrees (SVO) [14].

3 Detection of Small-scale Cracks

For the autonomous localization of structural risks on a regular basis, we analyze geometric deformations over time to inform crack detection. Regular 3D acquisition and reconstruction of built infrastructure enables the detection and observation of geometric deformations and changes of the surface since references from the past are available.

In this work, a 3D-reconstruction at a point in time t is represented by a point cloud C_t which is constituted by points residing on surfaces of the reconstructed object. Each point is associated with a color and a normal vector. We detect the formation of small-scale cracks over time by filtering for discontinuities in geometric deformations between C_t and a second point cloud $C_{t+\delta}$, which was acquired at a later point in time.

Figure 1 depicts an overview of our proposed crack detection pipeline. In a preprocessing step, we derive geometric features from both point clouds to establish a basis for optical flow estimation between our data sets. Next, we compute 2D optical flow [5] for a set of viewing points onto the point clouds and combine coinciding samples through a weighted blending based on surface normals and viewing directions to obtain a 3D surface flow field. The 3D surface flow field describes the displacement between surface points in C_t to corresponding positions in $C_{t+\delta}$. Finally, we filter the 3D surface flow to detect motion discontinuities, which correspond to cracks residing on the surface. The detected cracks are already registered into the coordinate system of the two input point clouds for subsequent analysis.

Initially, we replace per-point radiometric information with a geometry-based coloring, which is introduced in section 4. For the 3D surface flow estimation, we synthesize renderings of both point clouds from predefined viewing points generated from the camera extrinsics used for the structure-from-motion-based 3D reconstruction of both point clouds. At each viewing point, we compute 2D optical flow between the two data sets and combine coinciding optical flow samples from all viewing points to estimate an accurate 3D surface flow. The estimation of 3D surface flow is presented in section 5. In a postprocessing step, we verify the established correspondences by considering the geometric similarity in both point clouds using a novel SSIM-based descriptor. To produce our crack labeling, we determine a suitable filter orientation locally and convolve the 3D surface flow using a Gradient of Gaussian kernel. Section 6 describes our postprocessing and filtering in detail. The detection accuracy of the proposed pipeline is evaluated in section 7.

4 Geometry-based Coloring

The material color is often not ideal for detecting small-scale cracks, as radiometric variations due to weather and lighting conditions, graffiti or markings are frequent side effects of longer periods between acquisitions and can cause detection problems. In addition, vegetation growing in the vicinity of concrete surfaces introduces geometric occlusions which may inhibit detection even in areas where cracks remain visible otherwise.

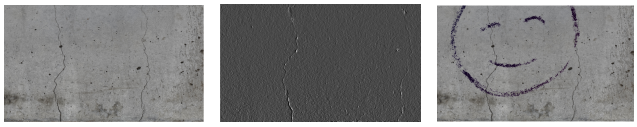
To mitigate these limitations, we replace per-point RGB information with a local geometric recoloring which is not influenced by uneven illumination, shadows or dirt and therefore well-suited to inform the detection of cracks. The geometry-based coloring should be invariant under transformations as well as large-scale non-rigid deformations of our data sets over time. Per-point surface normals provide a suitable starting point for our feature. We compute a best-fitting plane through a sufficiently large neighborhood $p_i \in C_t$, $|x - p_i| < \varepsilon$ of any point x and replace the color associated with point x by a gray-scale value based on ${n_x}^T n_{plane}$, where n_x and n_{plane} are the normals of point x and the plane respectively. Contrast stretching improves the saliency of the feature and leads to a better utilization of the available color range.

Figure 2 depicts our recoloring of a point cloud alongside the original material color for comparison. This pseudo-coloring is exclusively based on a local deviation from planarity of surface normals and enables the estimation of accurate 3D surface flow which is not disturbed by radiometric variations between the point clouds over time.

5 3D Surface Flow Estimation

In our approach, the 3D surface flow serves two purposes. Firstly, it allows us to infer correspondences of points on object surfaces in the presence of non-rigid displacements between our data sets. Secondly, we exploit motion discontinuities alongside crack boundaries to inform our crack detection. Thus, our detection is not based on the color offset of the cracks themselves as in most previous work, but on the displacements of the surrounding material. This is advantageous, because crack detection based on color offset can lead to false positives in outdoor areas, e.g. discoloration due to rust may be classified as a crack.

Since our data sets consist of points residing on object surfaces, we require 3D flow information only for those regions which contain surfaces. Therefore, we do not apply a global variational optical flow algorithm in three dimensions directly. Instead, we esti-



(a) (b) (c)

Figure 2: In previous work, the detection of cracks is based purely on the color of the material (a). We propose to replace the original material colors by a geometry-based artificial coloring (b), which allows us to perform crack detection even in the presence of severe radiometric disturbances (c) that would otherwise introduce a considerable amount of false positive matches and therefore result in unreliable crack detection.

mate the optical flow in 2D, and project the flow vectors back onto the 3D surfaces. The resulting representation can then be filtered in three dimensions to obtain crack labels in the coordinate system of the two input point clouds. We use an auxiliary sparse voxel octree (SVO) [14] to store and validate the computed 3D surface flow in a discretized space. The SVOs spatial resolution is based on the local point density such that it is high close to surfaces and low in empty regions. This representation also allows for adaptive precision when filtering for discontinuities.

To compute a 3D surface flow between two point clouds, we sample both data sets from the set of viewing points used for 3D-reconstruction, which guarantees that all surfaces are covered. First, we render both data sets in two separate passes and use a GPU-accelerated optical flow implementation to determine the 2D flow between our point clouds as seen by a virtual camera positioned and oriented according to a specific extrinsic at a time. The synthesized images used for the estimation of optical flow are based on the artificial recoloring presented in section 4. Then, we project each leaf-level voxel corresponding to point cloud C_t onto the image plane of the current view and interpolate the optical flow at this position. Next, we add the estimated motion vector and unproject the resulting position using the depth buffer corresponding to $C_{t+\delta}$. Thus, we obtain two world positions residing on the surface of the two data sets and the vector connecting them contributes to the per-voxel 3D surface flow. The flow contributions of flow samples from all viewing points are weighted into the adaptive SVO data structure based on the cosine similarity between the current viewing direction and the surface normal at the voxel under consideration, which has the effect of weighting samples taken from viewing directions orthogonal to the local surface higher than oblique ones. For each viewing point, we prevent the sampling of flow for voxels which are occluded by other parts of the data set by intersecting a ray connecting the voxel center and the viewing point with the SVO data-structure.

In our system, the 3D points are rendered as oriented, circular surfels [19] to approximate a closed surface. We achieve a high feature preservation by blending overlapping surfels smoothly using a 2-pass surface splatting approach [4]. For this purpose, we estimate the radii of all surfels such that a visually watertight surface is constructed.

Our data sets consist of hundreds of millions of colored points and their memory footprint can exceed the gigabyte-range. To scale our pipeline to very large, highly-detailed 3D data sets, we suggest a level-of-detail (LOD) approach similar to Goswami et al. [11], designed to effectively reduce the amount of data persistent in memory while optimizing the resolution of the data set for each viewing point. Alternatively, it is possible to split large point clouds into several overlapping sub point clouds before processing.

6 Post-processing and Filtering

Optical flow is prone to artifacts arising from geometric occlusions in particular and therefore, geometric changes in close proximity of the concrete surface (e.g. growth of vegetation) cause unreliable optical flow estimations. To reliably prevent false positive matches caused by erroneous correspondences, we propose to mask out regions where correspondences are uncertain by exploiting geometric information available in our point cloud reconstructions.

6.1 Masking of Uncertain Correspondences

For any 3D surface flow sample s, we compare the geometric and radiometric similarities of two regions $C_t(x)$ and $C_{t+\delta}(x')$ in our point clouds, where $x' = x + s$. We consider the regions similar if $C_{t+\delta}(x') \approx C_t(x)$ according to a suitable similarity measure.

Previous work has invented a number of geometric descriptors for the comparison of point clouds [10] [25] [23] [18]. Commonly, these descriptors consider the local neighborhood of a query point through binning approaches [2] and score the similarity of two queries. Alexandre [2] concludes that descriptors for point clouds should incorporate radiometric information to improve their performance. Palma et al. [18] propose a multi-scale shape descriptor based on a moving least square spherical fit through a local patch of the point cloud. They report strong change segmentation results, however, similar to most point cloud descriptors, their method does not account for radiometric information associated with points.

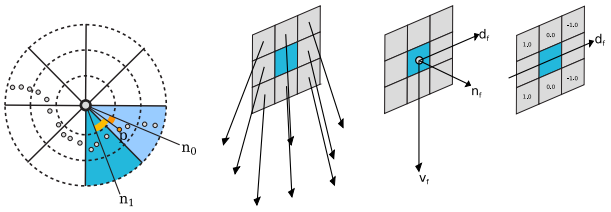
We introduce a customized descriptor based on our geometry-derived coloring to verify correspondences in our pipeline. First, we employ a unique, unambiguous local reference frame established in [24] to orient our descriptors relative to the underlying local surface. This greatly improves the potential similarity of the descriptors even if the orientation of the surface under consideration changed. Next, we divide the neighborhood $p_i \in C_t$, $|x - p_i| < \varepsilon$ and $p_j \in C_{t+\delta}$, $|x' - p_j| < \varepsilon$ of the query points x and x' into spherical bins. We found a multiple of the voxel edge length to be a suitable choice for ε. In our weighting scheme, we first assign a unit-length central axis n_b to each bin b. The contribution of points to each bin b is inverse-distance weighted by $acos({n_b}^T(\frac{p_i - x}{\|p_i - x\|}))$ or $acos({n_b}^T(\frac{p_j - x'}{\|p_j - x'\|}))$, which are the angular deviations between the bin axes n_b and the normalized vectors connecting the query points x and x' with the points p_i and p_j, respectively (Figure 3a). Finally, we perform a weighted averaging of our geometry-derived coloring for each bin.

For our choice of a suitable similarity measure, we use the structural similarity index (SSIM) [26] and we require the number of bins in both descriptors under comparison to be equal. For the comparison of two descriptors, we follow Wang et al. [26]:

$$SSIM(x,y) = \frac{(2\mu_{b_x}\mu_{b_y} + C_1)(2\sigma_{b_x b_y} + C_2)}{({\mu_{b_x}}^2 {\mu_{b_y}}^2 + C_1)(\sigma_{b_x}\sigma_{b_y} + C_2)} \quad (1)$$

In our case b_x and b_y denote the set of bins of two descriptors, μ denotes the average and σ the variance of the per-bin values, and $\sigma_{b_x b_y}$ corresponds to the covariance of b_x and b_y. If the structural similarity of two SSIM-descriptors is too low to support a correspondence, we remove the associated flow information and mask the region for the remainder of our pipeline. We only mask leaf-level voxels of the SVO directly, and propagate masking information to the inner voxels of the hierarchy. If any child voxel is masked, we also mask its parent, but we stop propagating masking information at a voxel edge length of 2 cm. This provides for cleaner, more reliable albeit more coarse masks.

Our masking of voxels prevents false positive matches in regions of the data sets where considerable geometric changes between acquisitions would otherwise introduce erroneous optical flow corre-



(a) (b)

Figure 3: (a) Illustration of our SSIM-based geometric descriptor for point clouds. For simplicity, it is shown in two dimensions but the principle extends in a straightforward manner to three dimensions. We divide the space around the query point in radial bins. The contribution of a point p to any bin is computed through inverse distance weighting of the angular deviations (depicted in yellow) to the nearest central bin axes n_0 and n_1. (b) Illustration of the tangential reference frame used in our filtering approach. For each filtering step, we average per-voxel flow directions in the neighborhood (left) to produce v_f. Our filtering direction d_f is orthogonal to the surface normal n_f and v_f (middle). We filter for discontinuities in flow directions along d_f (right) using a Gradient of Gaussian kernel.

spondences. The resulting masks are used to prohibit crack detection during filtering and crack labeling.

6.2 Filtering and Crack Labeling

To derive a crack labeling from our 3D surface flow, we use the adaptive representation of the sparse voxel octree. Initially, only a subset of leaf-voxels contain flow information. To propagate flow information to all voxels, including the interior ones, we apply a push-and-pull algorithm on the SVO. Starting from the leaf level voxels, we average existing flow information in a bottom-up process to the inner voxels of the hierarchy. When we reach the root-voxel, we start a top-down propagation of flow information from parent voxels to any children which do not contain flow yet. During this process, we avoid the traversal of any masked voxels and their descendants. After this process, all non-masked voxels of the SVO contain flow information.

We use a Gradient of Gaussian filter to detect discontinuities in our surface flow and obtain a per-voxel crack response. To perform filtering on our sparse voxel octree, we choose the size of the filter based on the edge length of the voxel at the center of the filter and use efficient point insertions to look-up values of adjacent voxels.

Our data sets are registered into a joint coordinate system and we focus on the detection of cracks in a concrete beam which is put under increasing vertical load at its center. Thus, the local crack directions roughly agree with the direction of non-rigid displacement of the concrete material. In this case, the most decisive spatial axis d_f to detect cracks lies tangential to the local surface patch and orthogonal to the direction of the crack. The filter response is highest along this axis when cracks are present. Therefore, we construct a local reference frame to orient our filter tangential to the local surface patch for any filtering step (Figure 3b).

7 Results and Discussion

We conducted an evaluation of our spatio-temporal crack-detection method by example of a 240 centimeter long concrete beam which was put under increasing load at its center and captured for 3D-reconstruction using 327 images. We tested our implementation with three point cloud series (Figure 4), each of which consists of two time steps; one without stress and one with high vertical load that shows a moderate amount of vertical cracks. To simulate the application of graffiti and growth of vegetation between acquisitions, we added radiometric and geometric variations to the second

(a) Series 1 (b) Series 2 (c) Series 3

Figure 4: This figure shows the second time step of three point cloud series used in our evaluation. Each series consists of two time steps, the first of which remains unchanged for all three data sets. *Series 1* features no substantial variations of color or geometry over time. We added color variations to the surface of *Series 2* in the second time step to simulate application of graffiti over time and to evaluate our approach when facing significant radiometric variations over time. In addition, we used an ivy generator to simulate the growth of vegetation over time in the second point cloud of *Series 3*.

time steps of our data sets. Each point cloud consists of 579 million points and amounts to approx. 9 GB.

Commonly, ground-truth comparisons are burdened by the difference in width of strokes made during manual annotation and the detected crack labels. This difference is a decisive factor as it can cause considerable disagreement during cell-by-cell comparisons. To alleviate this quality factor, we propose a hierarchical comparison.

We consider precision and recall with respect to the granularity of our comparisons. First, we label all leaf-level voxels in our SVO data structure that contain cracks according to our detection. Next, we propagate labels to the interior voxels of the hierarchy until the root node is reached. This representation of hierarchical labels allows us to perform a ground-truth comparison at different resolutions corresponding to a spectrum of detection accuracy at different scales. Finally, we created a manually annotated ground-truth SVO for comparisons which contains hierarchical labels created in a similar manner.

Figure 5 presents precision and recall scores when comparing our detection to manually annotated ground-truth cracks. We perform the comparison at 8 depths of our SVO data structures corresponding to specific voxel edge lengths in millimeters. Our results show, that we achieve a precision greater than 0.8 for accuracy in the millimeter range. At the sub-millimeter scale, a number of false positive matches decreases the precision. The propagation of crack labels to interior nodes of the SVO allows for coarser comparisons which improve the precision score. Above a voxel edge length of 2 mm, our high recall scores indicate that cracks were completely detected. False negative matches caused by disagreements in stroke widths start to appear below 2 mm and cause recall scores to drop.

Our approach avoids the use of original material color for crack detection and is therefore particularly robust against radiometric variations in 3D-reconstructions (*Series 2*). A complete detection of cracks in *Series 3*, which features considerable geometric occlusions in the second point cloud, is impossible and the corresponding recall score is upper-bounded by 0.85 in our case. However, the consistent precision scores show that the pipeline is able to mitigate false positive matches commonly arising from severe geometric changes between acquisitions.

We implemented the approach described in [8] which operates on a series of photographs in order to derive crack detection from discontinuities in optical flow. To allow for hierarchical comparisons with their work, we used two photographs of our concrete beam with their pipeline and propagated hierarchical crack labels in a quad-tree approach. Table 1 presents a comparison of f1-scores with respect to cell edge lengths chosen for comparison. The comparison suggests that all approaches perform well for *Series 1*, where no significant color variations or geometric occlusions be-

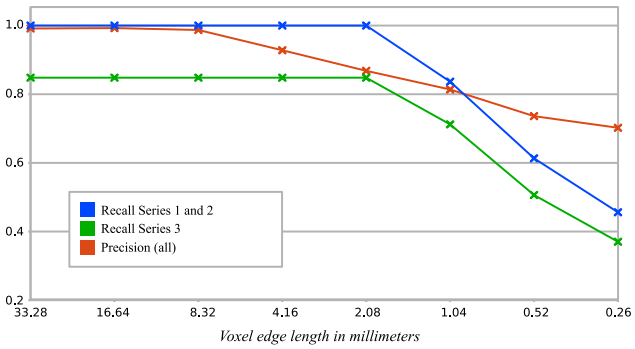


Figure 5: The plots show precision and recall scores of detected cracks in comparison with a ground-truth annotation. We perform a hierarchical comparison and the voxel edge length corresponds to the depth of our SVOs chosen for comparison.

Accuracy	Color-based	Geometry-based	[8]
8.333 mm	0.995	0.995	0.987
4.166 mm	0.963	0.963	0.941
2.083 mm	0.932	0.929	0.865
1.042 mm	0.841	0.821	0.791
0.521 mm	0.687	0.669	0.732
0.260 mm	0.572	0.554	0.695

Table 1: Comparison of f1-scores for *Series 1* with respect to cell edge lengths. All f1-scores decrease at higher comparison accuracy because the ground-truth annotation stroke width does not agree perfectly with the detection, causing lower recall values. Our f1-scores are computed by comparing crack labels at specific depths of a ground-truth SVO with detected crack labels at the corresponding depths. F1-scores for [8] are obtained by analogously comparing 2D-layers of a ground-truth image pyramid of crack labels with a second image pyramid derived from the detected crack labels.

tween time steps exist. The lower f1-score of our method at very high detection accuracy can be attributed to the granularity of the 3D-reconstructions used for evaluation.

When tested with *Series 2* and *Series 3*, the image-based method proposed in [8] yields recall and precision scores between 0.02 and 0.2. These very low scores are caused by false positives due to considerable changes in color and geometry of these data sets over time. In comparison, our results show that the use of 3D point clouds and the inherent geometric information can alleviate false positive matches caused by radiometric variations of the original material color as well as occlusions in overgrown regions.

Before correspondence estimation, we replace material colors by a geometry-based coloring as described in section 4. The similarity between 3D surface flow estimated using this recoloring and 3D surface flow created using the original material color is very high (PSNR $= 56.01$, SSIM $= 0.9978$). This confirms that the flow estimation based on our artificial recoloring results in a very similar 3D surface flow as the original color. In practice, the f1-scores based on flow obtained from the original material color, and flow estimated with our geometry-based coloring differ marginally.

Without taking the 3D-reconstruction into account, our pipeline takes approx. 1 hour and 41 minutes to process two time steps of our concrete beam data set (Figure 6).

In this work, we primarily focused on the detection of cracks caused by vertical bending stress and we assumed crack directions

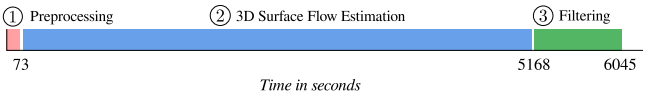


Figure 6: This figure illustrates the overall time in seconds taken to process each of our point cloud series shown in Figure 4.

to agree with the surrounding directions of displacement. The supplementary video shows a GPU-accelerated SVO-filtering to produce crack labels on the fly based on an interactively chosen filtering direction in the local surface plane. In our case, filtering directions orthogonal to the direction of the crack in the surface plane produce stronger crack detection results.

Illumination conditions and shadows may occasionally limit the reconstruction quality of surfaces in practice and we do not expect 3D-reconstructions to be available for an entire building or bridge everywhere in the wild. It would be sufficient to produce 3D-reconstructions in critical regions where cracks are most likely to occur. The accuracy of our method strongly depends on the significance of small-scale discontinuities with respect to the surrounding motion and therefore, a precise registration is required. Further improvements will be necessary to use our method with series of point clouds that are unregistered or deformed substantially between acquisitions.

8 Conclusion

We presented an algorithm for the detection of small-scale cracks in outdoor environments based on motion discontinuities in 3D surface flow. Optical flow estimation based on our local geometric feature produces correct correspondence information even in the presence of significant color variations, provided that 3D-reconstructions from two different points in time are available. As a result, our sparse voxel octree-based 3D filtering technique identifies cracks by exploiting local material displacement information, rather than the color offset of the cracks themselves. Our pipeline also incorporates a novel SSIM-based descriptor for point clouds to mask out poor correspondence estimations that may originate from geometric occlusions over time. We have shown that our filter identifies overgrown regions which allows us to limit the detection to those areas where cracks are visible.

In the future, we plan to generalize our method to enable the detection of all types of cracks in concrete and other materials, including cases where crack directions are not known in advance. Currently, we estimate motion from synthesized image pairs corresponding to a set of view points. An improved pipeline should estimate 3D surface flow directly and geometrically from the point cloud representations.

Acknowledgements

This work was supported by the German Federal Ministry of Education and Research (BMBF) under the project number 13N14657 (Project AISTEC). The concrete beam data set used in this work is courtesy of Bauhaus-Universität Weimar.

References

[1] S. Agnisarman, S. Lopes, K. C. Madathil, K. Piratla, and A. Gramopadhye. A Survey of Automation-enabled Human-in-the-Loop Systems for Infrastructure Visual Inspection. *Automation in Construction*, 97:52 – 76, 2019.

[2] L. A. Alexandre. 3D Descriptors for Object and Category Recognition: A Comparative Evaluation. In *Workshop on Color-Depth Camera Fusion in Robotics at the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vilamoura, Portugal*, volume 1, page 7, 2012.

[3] W. Benning, J. Lange, R. Schwermann, C. Effkemann, and S. Görtz. Monitoring Crack Origin and Evolution at Concrete Elements using Photogrammetry. In *ISPRS Congress Istanbul Commission*, volume 2004, 2004.

[4] M. Botsch, A. Hornung, M. Zwicker, and L. Kobbelt. High-quality Surface Splatting on Today's GPUs. In *Point-Based Graphics, 2005. Eurographics/IEEE VGTC Symposium Proceedings*, pages 17–141. IEEE, 2005.

[5] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert. High Accuracy Optical Flow Estimation based on a Theory for Warping. In *European conference on computer vision*, pages 25–36. Springer, 2004.

[6] H. Bruck, S. McNeill, M. A. Sutton, and W. Peters. Digital Image Correlation using Newton-Raphson Method of Partial Differential Correction. *Experimental mechanics*, 29(3):261–267, 1989.

[7] Y.-J. Cha, W. Choi, and O. Büyüköztürk. Deep Learning-based Crack Damage Detection using Convolutional Neural Networks. *Computer-Aided Civil and Infrastructure Engineering*, 32(5):361–378, 2017.

[8] S. Chaudhury, G. Nakano, J. Takada, and A. Iketani. Spatial-temporal Motion Field Analysis for Pixelwise Crack Detection on Concrete Surfaces. In *Applications of Computer Vision (WACV), 2017 IEEE Winter Conference on*, pages 336–344. IEEE, 2017.

[9] F.-C. Chen and M. R. Jahanshahi. NB-CNN: Deep Learning-based Crack Detection using Convolutional Neural Network and Naive Bayes Data Fusion. *IEEE Transactions on Industrial Electronics*, 65(5):4392–4400, 2018.

[10] A. Frome, D. Huber, R. Kolluri, T. Bülow, and J. Malik. Recognizing Objects in Range Data using Regional Point Descriptors. In *European conference on computer vision*, pages 224–237. Springer, 2004.

[11] P. Goswami, F. Erol, R. Mukhi, R. Pajarola, and E. Gobbetti. An Efficient Multi-resolution Framework for High Quality Interactive Rendering of Massive Point Clouds using Multi-way Kd-trees. *The Visual Computer*, 29(1):69–83, 2013.

[12] B. Horn and B. Schunck. Determining Optical Flow. *Artificial Intelligence*, 17:185–203, 1981.

[13] T. Hutt and P. Cawley. Feasibility of Digital Image Correlation for Detection of Cracks at Fastener Holes. *NDT & e International*, 42(2):141–149, 2009.

[14] S. Laine and T. Karras. Efficient Sparse Voxel Octrees. *IEEE Transactions on Visualization and Computer Graphics*, 17(8):1048–1059, 2011.

[15] Y. Li, H. Li, and H. Wang. Pixel-Wise Crack Detection Using Deep Local Pattern Predictor for Robot Application. *Sensors*, 18(9):3042, 2018.

[16] G. Morgenthal, N. Hallermann, J. Kersten, J. Taraben, P. Debus, M. Helmrich, and V. Rodehorst. Framework for Automated UAS-based Structural Condition Assessment of Bridges. *Automation in Construction*, 97:77 – 95, 2019.

[17] T. Nishikawa, J. Yoshida, T. Sugiyama, and Y. Fujino. Concrete Crack Detection by Multiple Sequential Image Filtering. *Computer-Aided Civil and Infrastructure Engineering*, 27(1):29–47, 2012.

[18] G. Palma, P. Cignoni, T. Boubekeur, and R. Scopigno. Detection of Geometric Temporal Changes in Point Clouds. In *Computer Graphics Forum*, volume 35, pages 33–45. Wiley Online Library, 2016.

[19] H. Pfister, M. Zwicker, J. Van Baar, and M. Gross. Surfels: Surface Elements as Rendering Primitives. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 335–342. ACM Press/Addison-Wesley Publishing Co., 2000.

[20] J. Poissant and F. Barthelat. A Novel Subset Splitting Procedure for Digital Image Correlation on Discontinuous Displacement Fields. *Experimental mechanics*, 50(3):353–364, 2010.

[21] Z. Qu, L.-D. Lin, Y. Guo, and N. Wang. An Improved Algorithm for Image Crack Detection based on Percolation Model. *IEEJ Transactions on Electrical and Electronic Engineering*, 10(2):214–221, 2015.

[22] J. Rupil, S. Roux, F. Hild, and L. Vincent. Fatigue Microcrack Detection with Digital Image Correlation. *The Journal of Strain Analysis for Engineering Design*, 46(6):492–509, 2011.

[23] D. Thanou, P. A. Chou, and P. Frossard. Graph-based Compression of Dynamic 3D Point Cloud Sequences. *IEEE Transactions on Image Processing*, 25(4):1765–1778, 2016.

[24] F. Tombari, S. Salti, and L. Di Stefano. Unique Shape Context for 3D Data Description. In *Proceedings of the ACM workshop on 3D object retrieval*, pages 57–62. ACM, 2010.

[25] F. Tombari, S. Salti, and L. Di Stefano. Unique Signatures of Histograms for Local Surface Description. In *European conference on computer vision*, pages 356–369. Springer, 2010.

[26] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image Quality Assessment: From Error Visibility to Structural Similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

[27] T. Yamaguchi and S. Hashimoto. Fast Crack Detection Method for Large-size Concrete Surface Images using Percolation-based Image Processing. *Machine Vision and Applications*, 21(5):797–809, 2010.

[28] T. Yamaguchi, S. Nakamura, R. Saegusa, and S. Hashimoto. Image-based Crack Detection for Real Concrete Surfaces. *IEEJ Transactions on Electrical and Electronic Engineering*, 3(1):128–135, 2008.