Fast Normal Approximation of Point Clouds in Screen Space

Daniel Schiffner
Goethe Universität
Robert-Mayer-Strasse 10
D-60054 Frankfurt
dschiffner@gdv.cs.uni-frankfurt.de

Marcel Ritter
University of Innsbruck &
Airborne Hydromapping OG
Technikerstr. 13a & 21
A-6020, Innsbruck, Austria
marcel.ritter@uibk.ac.at

Werner Benger
Center for Computation and Technology,
Louisiana State University
216 Johnston Hall
LA 70803, Baton Rouge, USA
werner@cct.lsu.edu

ABSTRACT

Displaying large point clouds of mainly planar point distributions yet comes with large restrictions regarding the surface normal and surface reconstruction. Point data needs to be clustered or traversed to extract a local neighborhood which is necessary to retrieve surface information. We propose using the rendering pipeline to circumvent a pre-computation of the neighborhood in world space to perform a fast approximation of the surface in screen space. We present and compare three different methods for surface reconstruction within a post-process. These methods range from simple approximations to the definition of a tensor surface. All these methods are designed to run at interactive frame-rates. We also present a correction method to increase reconstruction quality, while preserving interactive frame-rates. Our results indicate, that the on-the-fly computation of surface normals is not a limiting factor on modern GPUs. As the surface information is generated during the post-process, only the target display size is the limiting factor. The performance is independent of the point cloud's size.

Keywords

Normal Reconstruction, Tensor Information, GPU, Point Clouds

1 INTRODUCTION

Huge data sets are nowadays generated by simulations or by observational methods. Point clouds are e.g. the result of particle based simulation codes or laser scans, such as airborne light detection and ranging (LIDAR) scanning. Surface related information, such as the surface normal, can be used to enhance the visualization of point clouds, e.g. for illumination. Traditional methods for reconstruction surface information require an expensive spatial sort operation. Therefore, these are executed during a pre-process. Our method aims at improving the exploration of LIDAR data sets, before applying more expensive approaches.

In our work, we use the large data throughput of modern GPUs to generate a fast estimation of the surface properties within screen space. Therefore, we apply three possible approaches and compare the individual results. The first approach uses the fragment shader specific `dFdx` and `dFdy` functions. The second method calculates the surface normal by computing the cross product in a local neighborhood, which is available through the pixel neighborhood. The third applies a moving-least-squares approach to acquire tensor information. The resulting co-variance matrix is then used to compute the eigenvalues and eigenvectors.

In the next section, we list similar methods to our approach. Then, we present our methods and solutions to encountered issues. These methods are compared to each other and some examples are presented. Finally, we conclude with a summary of our findings and an outlook regarding future work.

2 RELATED WORK

Generic visualization frameworks, such as openWalnut [Walnut] or the visualization shell (VISH) are utilized for data exploration and processing of a large data sets. More expensive approaches to compute visual enhancements of points distributed on surfaces and lines, and geometrical reconstructions of lines have been done in [Bou212], [Rit12b] or [Rit12a].

The calculation of a surface normal is strongly connected to any surface reconstruction method. Especially for point based representations, methods using co-variance techniques [Ber94][Bjö05] are well

 ISBN 978-80-86943-75-6

suited, because no exact neighborhood is available and some noise is to be expected. Alexa defined the so-called point-set surfaces and presented some projection specific calculations [Ale04]. The co-variance matrix allows to assess the quality of the point cloud data set using direct tensor field visualization methods, such as displaying tensor splats [Ben04]. To compute the eigenvalues and eigenvectors from a given co-variance matrix, the analytical approach presented by Hasan [Has01] or one of the methods presented by Kopp [Kop06] can be applied.

Yet, these methods rely on the identification of an accurate neighborhood. To acquire this information, the input data set needs to be sorted. Neighbors are either found by a brute force approach – which is not suitable at all –, by a tree search or by a Morton ordering [Con10]. A tree as well as a Morton order are highly suited for parallelization.

Instead of creating a kd-tree or a Morton order in world space, a neighborhood can also be computed in screen space. Thus, the computation is only performed on the currently visible part of the data set. This is commonly done by splatting the data points and extracting the properties from the frame buffer. Similar to the approach presented by [Sch11] or [Yan06], we use only screen space information for th selection of the neighborhood. The splats are projected using either a fixed or adapted point size, as proposed by Rusinkiewicz [Rus00]. Once the surface information is available, also high quality splatting techniques [Bot05] could be applied.

3 APPROACH

We use the information available in screen space to reconstruct a surface and its corresponding normals. We designed an approach consisting of three individual steps, as illustrated in figure 1.

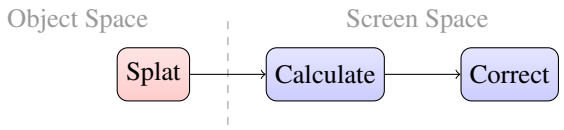


Figure 1: The outline of our screen space normal reconstruction. The first pass consists of splatting the depth values which are used in the consecutive passes. The second pass approximates the surface normal, while the optional third pass smooths the resulting values.

The first pass is a simple splatting of the input data and provides the depth information required by reconstruction. Each pixel is hereby surrounded by neighbor candidates. The second pass uses these depth values and computes surface properties. The candidates are inspected and rejected if the distance is too large, i.e. their interpolation weight is too small. The last pass is optional and allows a further enhancement of the quality of the reconstructed properties.

Splatting the Point Cloud

We draw the point cloud, which will be reconstructed, using either a fixed or an approximate point size. Our approach only requires a depth buffer for computation of the surface information. As the depth-buffer is generated, in general, by all rendering approaches, this method can be applied to all scenarios.

To increase the accuracy, we encourage using a multi-sample depth-buffer. This allows the retrieval of multiple depth values per individual sample. Using a sampling count of 8 means that we are able to capture – at most – 8 individual splat depth values at once. It is, of course, possible that the unprojected world space coordinates are identical or invalid, i.e. the depth value was not set. Still it increases the stability of the following normal calculations. Multi-sampling is only applied within the first post process.

Normal Definition

We calculate the wold space coordinates of the current pixel by un-projecting it based on the multi-sampled depth-buffer. The reconstruction of the surface normal can then be performed in three ways. The first method uses the local derivatives directly available in the fragment shader. The second and third method approximate the surface using a generic neighborhood description.

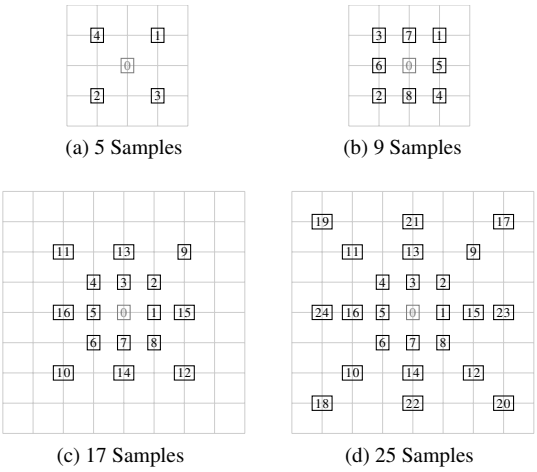
This neighborhood is defined by fixed sampling patterns. The most simple version takes 5 samples within a 3x3 neighborhood, while the most complex version selects 25 samples in a 7x7 neighborhood, see figure 4. The samples are focused on the diagonals, which increase the overall area captured during reconstruction. Note, that we use ascending indices for the opposite sample positions. This enables a simple definition of diagonals within a shader.

In our test, we did not observe any differences between the 5 and 9 sample schemes. This indicates, that the reduced representation is already able to capture the surface properties. The extended schemes, i.e. 17 and 25 samples, further increase stability of the results and are more comparable to off-line methods.

We orient all normals by inverting those, where the z-component is negative. All selected splat samples are visible and, thus, require a normal which is facing towards the camera.

To assure correct identification of possible neighbor candidates, a maximal distance is introduced. Neighboring pixels may not be true neighbors within world space due to the projection. Therefore, we reject every sample that is not within this configurable distance. This is comparable with the maximal distance in the MLS [Ale04] or tensor computations [Rit12a].

 ISBN 978-80-86943-75-6



(a) 5 Samples (b) 9 Samples

(c) 17 Samples (d) 25 Samples

Figure 2: The used sampling schemes for defining the local neighborhood of a fragment. The center point 0 is optional.

Local Derivatives

Shaders support the calculation of local derivatives within the fragment shader since GLSL version 1.10. For reconstruction of the surface normal, the functions `dFdx` and `dFdy` are used. These internally extract neighbor positions from concurrent thread blocks and are only available in the fragment stage. This means that the surface is completely splatted and the individual samples may have overlapped. With c, the current position in clip-coordinates, the surface normal $\vec{n}$ is computed:

$$\vec{n}(c) = dFdy(c) \times dFdx(c)$$

This method is very sensitive to noise or irregularities in the depth buffer and in many cases produces normals not representing a good reconstructed surface. However, if the surface is continuous and the splat size is carefully chosen, this method will suffice.

Plane Approximation

Similar to the computation of mesh surface properties, we approximate face normals within this approach. The normals are accumulated and the resulting vector is normalized. Finally, we impose an orientation and align the vector.

To obtain the needed vectors, we use one of the proposed sampling schemes. Each direction vector is built up either by diagonal or counter-clock-wise (ccw) samples. The diagonals generate smoother results and do not require the center point at sample 0. This is similar to the anti-alias algorithms used in the rendering pipeline. The ccw approach accounts more for local changes and takes the center point into account. In the diagonal case, we obtain the surface normal by using the following formula:

$$\vec{n}(c) = \frac{1}{N} \sum_{i=0}^{\lfloor \frac{N}{4} \rfloor} \vec{d}_{4i} \times \vec{d}_{4i+2}$$

With $\vec{d}_i = s_i - s_{i+1}$. We optimize the sampling schemes for a diagonal pattern, since we intend to create smooth surface normals with minimal noise.

Tensor Information

Using tensor information instead of flat patches leads to a smoother reconstruction. To derive this information, the computation of eigenvalues and eigenvectors is mandatory. We compute the point distribution tensor by deriving the co-variance matrix for the current position c, as presented by [Rit12a] and similar to [Bjö05]:

$$CM(c) = \frac{1}{N} \sum_{k=1}^{N} w_{ik}(d_{ik} \otimes d_{ik}^T)$$

where $d_{ik} = c - S_k$, d_{ik}^T is the transpose, N is the number of samples around center point c, S_k the sample and w_{ik} is a weighting function. Here, we apply a weighting of $w_{ik} = \frac{1}{\|d_{ik}\|^2}$.

The tensor product $\otimes$ is built by the direction vectors pointing from the current fragment's world coordinate to its points in the neighborhood. The weighted sum of these vectors result in the final point distribution tensor.

We compute the eigenvalues with the "Cordano" method presented by [Kop06]. This approach results in more stable vectors than the method proposed by Hasan et al. [Has01]. Similar findings were made by the developers of openWalnut [Walnut]. The eigenvector related to the minor eigenvalue hereby represents the surface normal. The vector is easily oriented, since the calculation is performed in clip-coordinates and the normal vectors have to face the camera.

Smoothing Normals

In a second, optional, screen space pass we correct the computed normals. We extract and scale adjacent normals within a local neighborhood, where the center normal is being favored. The surface normal is yield by accumulation of the weighted vectors.

Different weights and neighborhood sizes can increase the accuracy of the result. However, this does not apply to all situations. Especially, when using the plane approximation method, quality decreases, when the normals contain lots of noise.

4 RESULTS

We implemented a prototype, which has been tested on a i5 670 system with 8 GB RAM and a GeForce 680 running on Windows 7.

 ISBN 978-80-86943-75-6

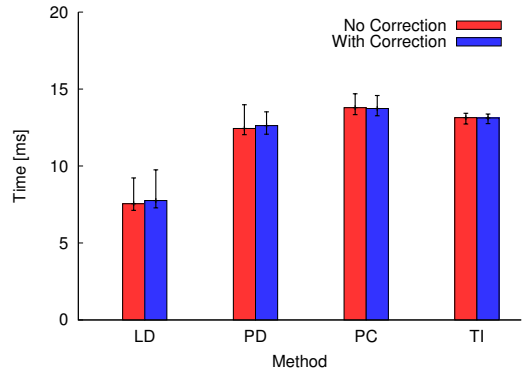


Figure 3: Timing results achieved using a screen size of 1024x768 with 8 multi-samples and the 9 samples scheme. LD denotes the local derivatives, PD the plane approximation using diagonals, PC the plane approximation using counter-clock-wise pattern, and TI the tensor information.

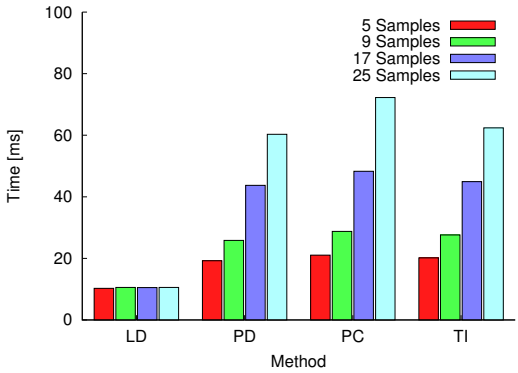


Figure 4: Influence of changing sampling scheme size for the reconstruction methods. Results taken with a screen resolution of 1600x1200. All methods use a 8 times multi-sampled depth-buffer.

9 Samples Scheme / 8 Multi-samples			
Operation	Min [ms]	Max [ms]	Avg [ms]
Splat	8.963	9.030	9.000
Normal	17.521	18.435	17.968
Correction	0.468	0.717	0.493
17 Samples Scheme / 8 Multi-samples			
Operation	Min [ms]	Max [ms]	Avg [ms]
Splat	8.801	10.654	8.980
Normal	34.278	35.711	34.890
Correction	0.466	5.740	0.702

Table 1: Distribution of the processing times among the individual operations of the proposed method. Results taken with a screen resolution of 1600x1200 using the tensor method.

Timings

On all systems, we observed interactive frame rates with all methods. The fasted method is the local derivatives (LD) approximation, while the tensor information (TI) is the most expensive variant. The plane approximation with diagonals (PD) is slightly faster than the tensor variant. The ccw plane approximation (PC) is worse in terms of performance compared to the PD, due to the definition of the sampling scheme.

In figure 3, the average processing times are shown, including the generation of the depth values. We used a fixed multi-sampling count of 8 in all presented timing results. Thus, the real number of samples taken per pixel needs to be multiplied by 8. For better readability, we continue to use the introduced sampling count.

The splatting of the point cloud requires a significant amount of time. In our tests, it varied in the range of 30% to 50% mainly depend on the used screen and splat sizes.

The used sampling scheme size has a large influence on the performance and quality of the reconstruction, as seen in figure 4. The performance scales linearly with the number of used samples. However, the quality of the reconstruction is not necessarily improved when using a very high sampling count. This is due to the fact that the surface is smoothed and local information is suppressed.

We also measured the contribution of the individual steps performed by our approach. Interestingly, the splatting itself consumes a large amount of the overall processing time, while the correction requires only very little processing time. The larger the number of used samples, the higher the reconstruction times. Table 1 lists the detailed timings of the involved steps: “Splat” represents the splatting of the depth values, “Normal” the reconstruction and “Correction” the final smoothing.

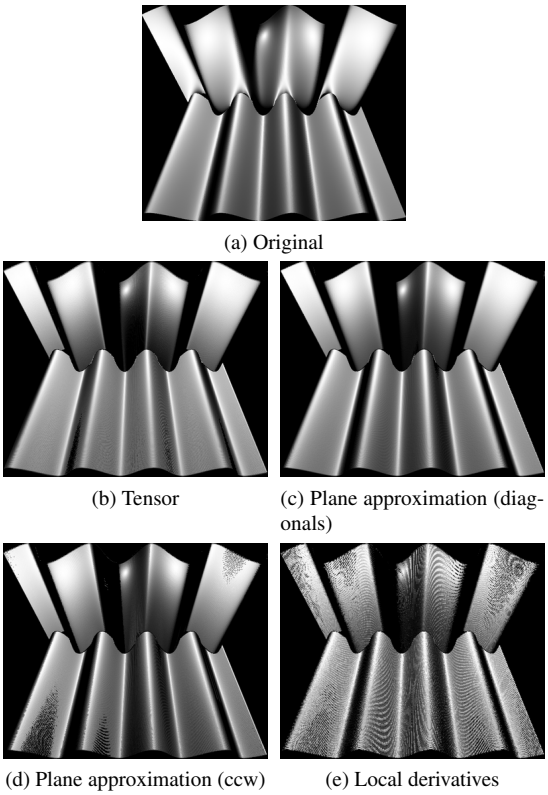
Visual Results

All methods are able to reconstruct both noisy and smooth surfaces. We use several splatted object point clouds as test cases. All point clouds consist of at least 250k points to assure a high sampling density.

The results of the described reconstruction methods are shown in figure 5. These indicate that the TI method provides a stable and accurate reconstruction. The PD approach provides excellent results in smooth data sets. The LD approach always generates large noise. Despite not being suitable for a high quality surface approximation, it is the fasted approach.

To simulate noisy data, we alter the vertex positions within the splat shader. A light source is positioned below the object. The illuminated scene is shown in figure 6. The TI method generates the smoothest result, while the PD method yields more normals that differ widely from the original ones. The LD method provides the worst reconstruction. All methods generate

 ISBN 978-80-86943-75-6



(a) Original

(b) Tensor

(c) Plane approximation (diagonals)

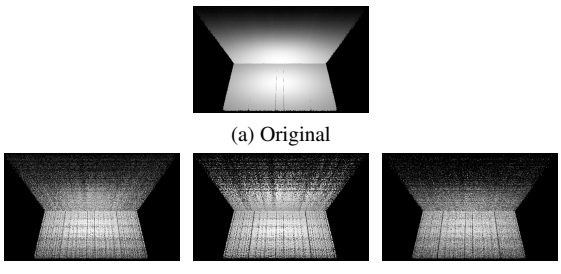
(d) Plane approximation (ccw)

(e) Local derivatives

Figure 5: Reconstruction of the surface normal used for illumination. (a) shows the original object with precomputed normals. (b) to (e) depict the proposed reconstruction methods.

more invalid normals in the low sampled region on the top.

Figure 7 illustrates the influence of the optional correction pass. The corrected normals are smoother and the number of correctly oriented surface normals is higher. The vectors are visualized via colors showing the x-, y-, and z-coordinates as red, green, and blue values.



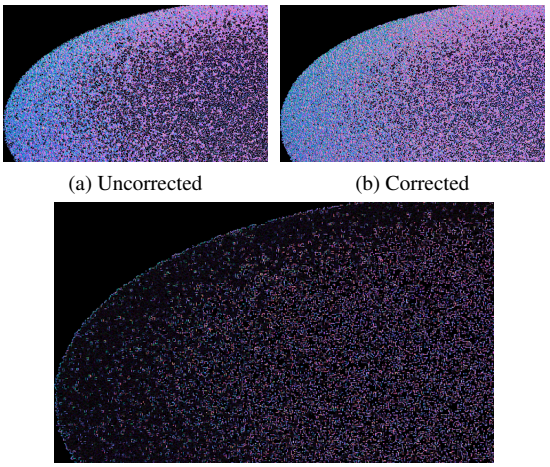
(a) Original

(b) Tensor

(c) Plane approximation

(d) Local derivatives

Figure 6: Reconstructed normals used for illumination in a test scenario with two planes. Noise is added to the input data. Even normals at the edge are well reconstructed, but tend to be smoothed.



(a) Uncorrected

(b) Corrected

(c) Difference Image

Figure 7: The influence of the correction pass applied to an ellipsoidal surface. The surface xyz-normal is illustrated as a rgb-color. The corrected version (b) contains more valid normals. The difference is visualized in (c).

Since the correction pass is very fast and increases the stability of the reconstruction, we always enable this pass in the following tests.

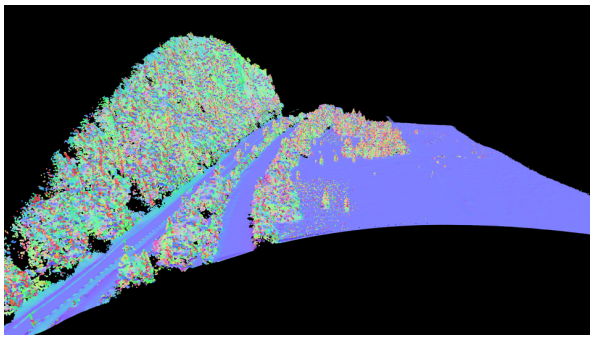
Application to a LIDAR Data Set

A point could stemming from an airborne laser-scan is used for further investigation of the technique and validation of the technique by a real-world application. We chose a small section of a bathymetric scan of the river Loisach in Bavaria (Germany), acquired with the hydrographic laser scanner Riegl VQ-820G [Ste10]. The scan contains different kinds of structures: fields, trees, lower vegetation, a river, a street with cars, power cables and a steep slope partially covered with vegetation. Figure 8 shows a side and a top view of the scan.

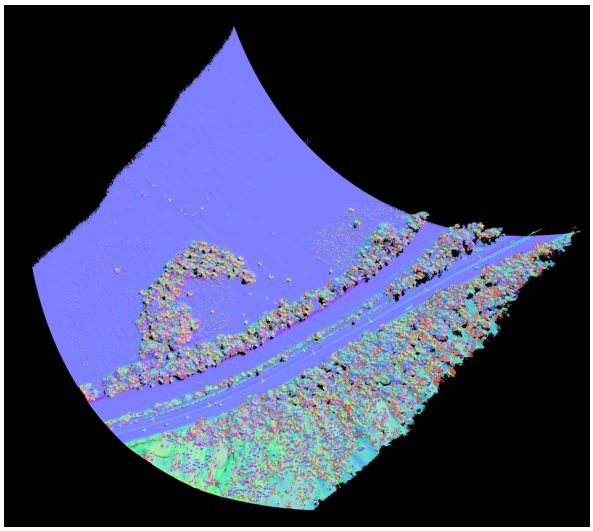
The two million points are colored by the minor eigenvector of the point distribution tensor computed in world-space.

The point distribution tensor was computed by using a neighborhood radius of 0.5, 1.0 and 2.0 meters. Two different weighting functions have been tested: constant weight and $\frac{1}{\|d_{ik}\|^2}$ weight. Using a kd-tree for finding neighbors and 6 OpenMP parallel threads on an Intel Xeon X560@2.67GHz the according computation times are 41, 85, and 218 seconds for the three radii. This computation of the tensor is a demanding computational tasks. However, it has been shown, that the tensor can be used for feature extraction, object recognition, and to improve the segmentation of point clouds [Rit12a][Rit12b][Bjö05]. When just looking at the minor eigenvector via color, the fields, the river, the street, the slope and the vegetation can be well distinguished from each other, visually.

 ISBN 978-80-86943-75-6



(a) Side view



(b) Top view

Figure 8: LIDAR laser-scan of a section of the Bavarian river Loisach in Germany. Laser echoes are illustrated as colored points. Color shows the minor eigenvector of the point distribution tensor. Vegetation can be visually distinguished from the ground and the river.

Next, we compare this expensive, fine grain computation in world space with our screen space technique. The results indicate that the approach is able to reconstruct the normals with rather high quality. The normals widely match with the normals calculated in world space, as shown in figure 9. However, differences in the forest areas of the scan are visible.

Also, where the sampling density near the camera position is not high enough to ensure high quality reconstruction in this region.

To compare the results of the different methods, we recorded a series of images from the Loisach data set. The TI method produces the most reliable results, while requiring a high sampling count. The PD method is able to create very smooth normals regardless of small surface changes, e.g. the missing power line in the upper region 10. The PC method includes it, but is more unstable. The LD method is the most efficient approach while yielding the worst quality in comparison to the other methods.

The correction pass increases the quality and the stability of the results by reducing the number of invalid surface normals. Figure 10c illustrates the enabled correction pass and figure 10d .

5 CONCLUSION

Our results show that a fast approximation of the surface normal can be achieved in real-time. Here, the surface is solely reconstructed from the depth-buffer and projection parameters. With our approach a preprocessing of surface information may be delayed until a region of interest has been selected. The results indicate that especially the tensor-based approach to determine the surface normal of a point cloud is a well-working method.

In comparison to the off-line world space method, we are able to create similar results at interactive frame rates. The loss of quality is negligible and is only visible in under-sampled regions. However, this method can only provide an approximation of the real point-cloud's surface information. The tests show that an increase of the neighborhood size decreases the performance linearly. A good quality is already achieved with small neighborhood sizes. The focus on the diagonals in the sampling schemes reduce the number of required samples.

6 FUTURE WORK

We plan to combine this technique with level of detail rendering to provide good visual representations of large airborne LIDAR scans. The surface normals provide important information to control such a level of detail algorithm.

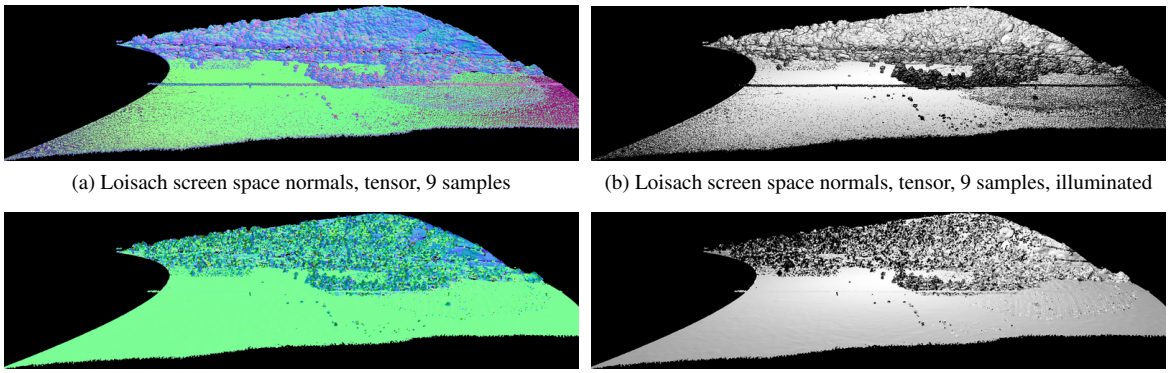
The splatting technique could be enhanced by utilizing more information represented in the point distribution tensor. Extracting some features of the tensor will improve the readability of point clouds without expensive pre-computations.

Additionally, we plan to enhance the reconstruction method by providing more weighting functions besides the $\frac{1}{\|d_{ik}\|^2}$ weight for the computation of the co-variance matrix.

To avoid expensive re-calculations, we plan to employ a caching strategy. A re-computation of the surface normals would only be required when camera location or point coordinates are changed, further increasing the overall performance of the approach.

7 ACKNOWLEDGEMENTS

Special thanks to Frank Steinbacher for providing the LIDAR data set of the river Loisach. This work was supported by the Austrian Science Foundation FWF DK+ project Computational Interdisciplinary Modeling



(a) Loisach screen space normals, tensor, 9 samples

(b) Loisach screen space normals, tensor, 9 samples, illuminated

(c) Loisach world space normals, 1m, squared

(d) Loisach world space normals, 1m, squared, illuminated

Figure 9: The reconstruction of the minor eigenvector using the fast screen space approach.

(W1227), and grant P19300. This research employed resources of the Center for Computation and Technology at Louisiana State University, which is supported by funding from the Louisiana legislatures Information Technology Initiative. This work was supported by the Austrian Ministry of Science BMWF as part of the Uni-Infrastrukturprogramm of the Forschungsplattform Scientific Computing at LFU Innsbruck.

8 REFERENCES

[Con10] Connor, M., and Kumar, P.: Fast Construction of k-Nearest Neighbor Graphs for Point Clouds. IEEE TVCG 16, No.4. pp.599–608, 2010.

[Yan06] Yang, R., Guinnip, D., Wang, L.: View-dependent textured splatting. The Visual Computer 22, pp.456–467, 2006.

[Has01] Hasan, K.M., Basser, P.J., Parker, D.L., Alexander, A.L.: Analytical computation of the eigenvalues and eigenvectors in DT-MRI. J. Magn. Reson. 152, pp.41–47, 2001.

[Ale04] Alexa, M., Rusinkiewicz, S., and Adamson, A.: On normals and projection operators for surfaces defined by point sets. Eurographics Symp. PBG., pp. 149–155, 2004.

[Bou212] Boulch, A., and Marlet, R.: Fast and Robust Normal Estimation for Point Clouds with Sharp Features. Comp. Graph. Forum 31, No.5, pp.1765-1774, 2012.

[Walnut] Open Walnut. `http://openwalnut.org`.

[Ben07] Benger, W., Ritter, G., Heinzl, R.: The Concepts of VISH. 4th High-End Vis. Workshop, pp.26–39, 2007.

[Ben04] Benger, W., Hege, H.-C.: Tensor splats. Conf. on Vis. and Data Analysis, Vol.5295, pp.151–162, 2004.

[Ber94] Berkmann, J., and Caelli, T.: Computation of surface geometry and segmentation using covariance techniques. IEEE TPAMI 16, No.11, pp.1114–1116, 1994.

[Rit12a] Ritter, M., Benger, W., Cosenza, B., Pullman, K., Moritsch, H., Leimer, W.: Visual Data Mining Using the Point Distribution Tensor. IARIS Workshop on Computer Vision and Computer Graphics, VisGra, 2012.

[Rit12b] Ritter, M., Benger, W.: Reconstruction Power Cables From LIDAR Data Using Eigenvector Streamlines of the Point Distribution Tensor Field. WSCG, pp.223–230 ,2012.

[Bjö05] Johansson, B., and Moe, A.: Object Recognition in 3D Laser Radar Data using Plane triplets, technical report LiTH-ISY-R-2708, Dept. EE, Linköping University, 2005.

[Rus00] Rusinkiewicz, S., Levoy, M.: QSplat: A Multiresolution Point Rendering System for Large Meshes, SIGGRAPH '00, pp.343–352, 2000.

[Bot05] Botsch, M., and Hornung, A., and Zwicker, M., and Kobbelt, L.: High-quality surface splatting on today's GPUs. Eurographics VGTC Symposium on PBG, pp.17–24, 2005.

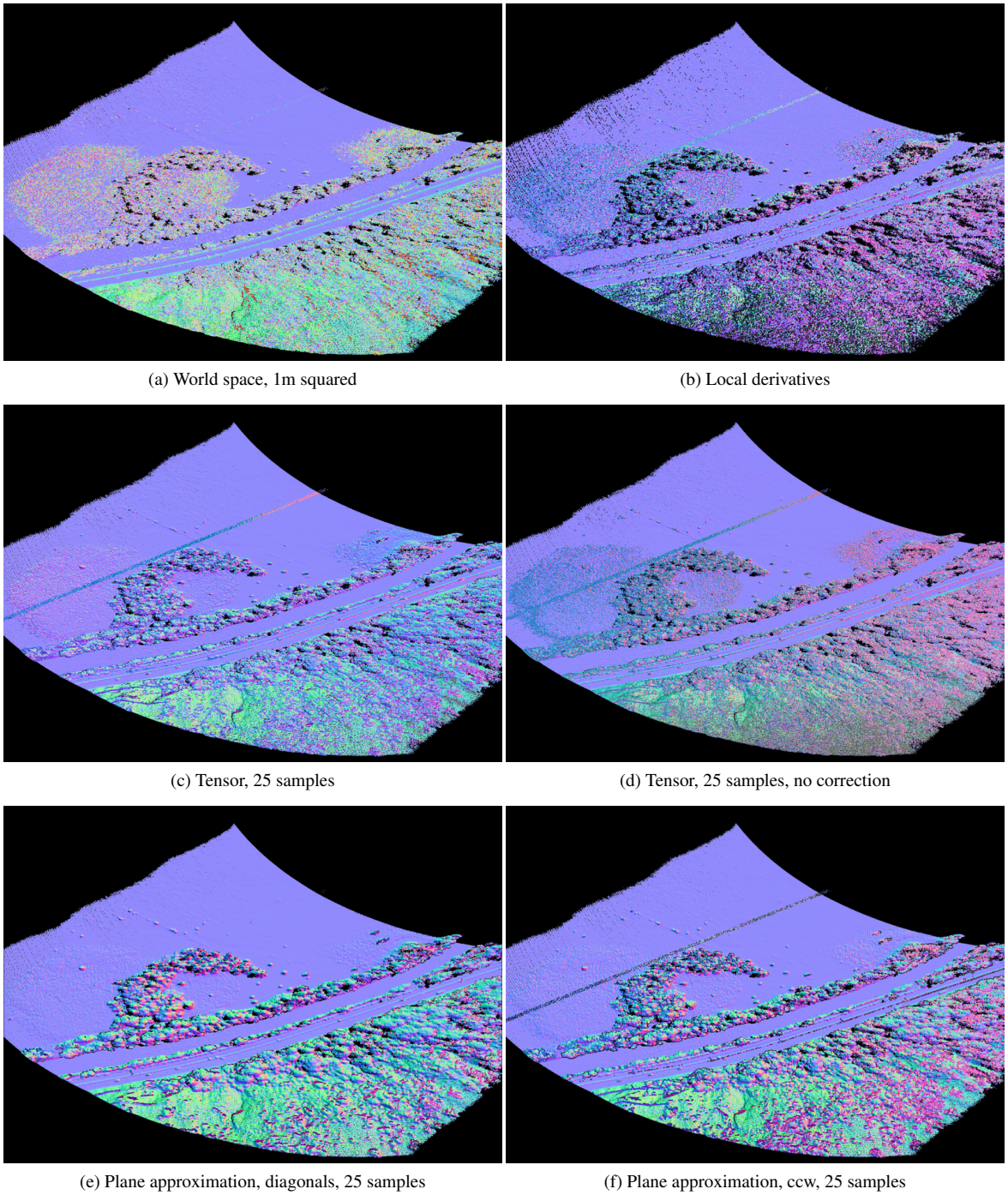
[Sch11] Schiffner, D., Krömker, D.: Three Dimensional Saliency Calculation Using Splatting, 6th ICIG, pp.835–840, 2011.

[Shi09] Shirley, P., and Marschner, S.: Fundamentals of Computer Graphics, 3rd Edition, A.K. Peters Ltd, 2009.

[Kop06] Kopp, J.: Efficient numerical diagonalization of hermitian 3x3 matrices, arXiv:physics/0610206v1, 2006.

[Ste10] Steinbacher, F., Pfennigbauer, M., Ulrich, A., and Aufleger, M.: Vermessung der Gewässersohle - aus der Luft - durch das Wasser, in Wasserbau in Bewegung ... von der Statik zur Dynamik. Beiträge zum 15. Gemeinschaftssymposium der Wasserbau Institute TU München, TU Graz und ETH Zürich, 2010.

 ISBN 978-80-86943-75-6



(a) World space, 1m squared

(b) Local derivatives

(c) Tensor, 25 samples

(d) Tensor, 25 samples, no correction

(e) Plane approximation, diagonals, 25 samples

(f) Plane approximation, ccw, 25 samples

Figure 10: Comparison of the different reconstruction methods used on the Loisach LIDAR data set.

 ISBN 978-80-86943-75-6