IEEE Second International Conference on Data Stream Mining & Processing
August 21-25, 2018, Lviv, Ukraine

F-transform 3D Point Cloud Filtering Algorithm

Andriy Yerokhin
Computer Science Faculty
Kharkiv National University of Radio Electronics
Kharkiv, Ukraine
andriy.yerokhin@nure.ua

Valerii Semenets, Alina Nechyporenko
Biomedical Engineering Department
Kharkiv National University of Radio Electronics,
Kharkiv, Ukraine
alina.nechyporenko@nure.ua

Oleksii Turuta, Andrii Babii
Department of software engineering
Kharkiv National University of Radio Electronics
Kharkiv, Ukraine
{oleksii.turuta, andrii.babii}@nure.ua

***Abstract* — The paper proposes a new 3D point cloud filtering approach using F-transform. We achieve this by usage of uniform fuzzy partitioning and applying direct and inverse discrete F-transform on a point cloud data. The point cloud was filtered in a depth domain. The performance of developed approach was compared with several common-used statistical-based point cloud filtering methods.**

Keywords* — *point cloud, filtering, fuzzy methods, F-transform

I. Introduction

A technology of 3D scanning have been in active development over past few decades. Various hardware can be used to obtain the point cloud data. This includes implementations based on usage of lasers, structural light, infrared light, stereo and time-of-signal measurement methods [1].

Acquired data can be used in various applications, including medical, environmental, engineering and entertainment. Low-cost 3D sensors become more accessible over past few years [2], but point clouds obtained with such sensors, however, contain noise component and outliers due to different reasons, related with the technology limitation, nature of the measured surfaces and lighting [3]. Therefore, 3D point cloud filtering is an important pre-processing operation of raw data registered by sensor. Filtering approaches can be divided into the following seven groups of algorithms for filtering the 3D point cloud [4]:

- Filtering techniques based on statistic. These methods include various static parameters for filtering: local median filtering, local averaging, kernel-based clustering approaches, different variant of principal component analysis, Bayesian statistics usage for denoising, variations of Least Squares approach, optimization techniques on normal estimation, growing neural gas networks.
- Filtering based on neighborhood analysis. This approach estimates results using similarity metric for a point and its neighborhood. This wide group of methods such as: bilateral filters and its variations, 3D mean shift filter, 3D non-local means filter, filters based on relative deviation of the local neighborhood metrics and the average local neighborhood
- Projection-based approaches. This is a set of methods based on adjusting the position of each point using different projection strategies. It includes a locally optimal projection operator and its modifications, moving least squares and its modifications, plane fitting operation.
- Filtering based on a signal processing methodology. It includes usage of Laplacian operator and its modifications, Fourier transformation based filters, combination of Laplace–Beltrami operator and Wiener filtering.
- Filtering based on partial differential equations (PDE). PDE can be applied to pre-processed neighborhoods, curvatures, weighted arbitrary graphs to filter point clouds.
- Hybrid filtering technique. Usually this is a mix of two or more methods from different groups to filter point clouds.
- Other methods. This class includes groups of methods which are not related to previous points. It includes voxel grid, oriented particles and similar filtering approaches.

In this paper we will focus on F-transform application to the point cloud processing. F-transform is an approximation method based on fuzzy partitioning proposed by Perfilieva in [5] and it has various practical applications. Fuzzy sets are used to deal with uncertain information [6] usually, but fuzzification of the crisp data and usage of fuzzy theory may give additional benefits for data processing. In paper [7] the main principles of F-transform usage for noise filtering were described. Application of fuzzy transformation to image processing were published in paper [8]. Approaches based on usage of fuzzy relations and choosing of optimal granulation level (which were represented by a fuzzy rule) were described in [9]. F-transform were used for image processing in the work [10].

The aim of this work is to apply the F-transform to 3D point cloud filtering.

Section II describes the input data sources and methodology of the point cloud processing. Section III describes an application of the F-transform to 3D point cloud filtering and some aspects related to usage of this transformation. Results of filtering and comparing with several statistical-based 3D filtering methods are demonstrated in section IV. Conclusions and some of further research directions are described in Section V

II. Point Cloud Data Preprocessing

A. Data sources

In the current paper, we used the Stanford Models dataset [11]. These models are scanned by a Cyberware 3030 MS scanner. There are 3D models, such as Stanford Bunny, Drill bit, Happy Buddha, Dragon, Armadillo and etc. For example Fig. 1, the Stanford Bunny model consists of 10 scans and total size is 362,272 points (about 725,000 triangles). The model was aligned by the modified ICP algorithm [12]. The reconstruction size is 35947 vertices and 69451 triangles. The 3D model is zippering and smoothing merged volumetric range of image of the manifold surfaces. The point cloud was processed after 3D scanning. The reconstructed model is point cloud which stored in a PLY file format (ASCII format).

The Stanford Bunny model (Fig. 1) is used in pre-processing section.

Fig. 1. Stanford Bunny

B. Point cloud data pre-processing

A high-level overview of fuzzy filtering of a 3D point cloud includes a pre-processing irregular point cloud, preparing patches, fuzzy filtering for noise reduction of patches as a result point clouds. Overview of pre-processing stages is given in Fig. 2.

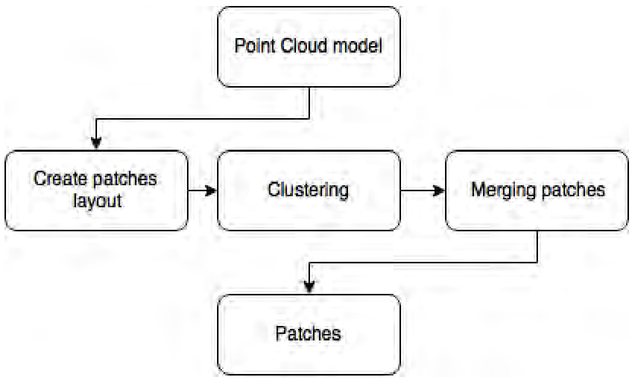

Fig. 2. High-level stages of pre-processing point cloud.

First of all, the irregular point cloud model is divided into a number of points. A set of closed points is collected into raw patches. Criteria of collection points are defined in [12 GEOMETRY]. Such a patches layout describe the surface of this 3D objects presented by point clouds.

Secondly, we cluster points of patches layout into clusters. A binary space partition (BSP) tree samples the set along size of bounding box. We merge the leaves of the tree into clusters with a common parent in the BSP tree. There are potentially many other neighbors so we stop clustering stage when cluster achieves need size (20-100 samples, depending of model) [12].

Thirdly, created patches layout has local neighborhood information. An optimization approach [12] is used for merging patches. Let a local quality metrics Φ and P_i, P_j be neighboring patches such as $P=(P_i,P_j)$ which merges candidate patches.

Φ is defined as:

$$\Phi(P)=\Phi_{Size}(P)\cdot\Phi_{NC}(P)\cdot\Phi_{B}(P)\cdot\Phi_{\mathrm{Re}\,g}(P) \quad (1)$$

where, $\Phi_{Size}(P)$ - estimation of high priority of small patches; $\Phi_{NC}(P)$ - estimation of penalty of increasing width of patch; $\Phi_{B}(P)$ - estimation of boundary of the patches; $\Phi_{\mathrm{Re}\,g}(P)$ - is regularization of the patch distribution.

The raw patches are merged into a connected region of surface.

Finally, we have got the merged patches of point clouds. Visualization of merged patch presented in Fig. 3. We create a mesh of point clouds. It is a part of nose of the Stanford Bunny. In next section we will apply fuzzy transform (FT) approach for smoothing this patch.

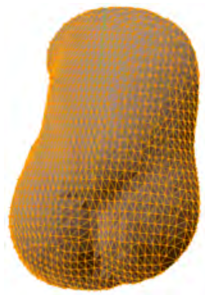

Fig. 3. Visualization of merged patch

III. 3D POINT CLOUD FT-SMOOTHING

Filtering of noise in the patch with point cloud, received in the previous section will be next step in point cloud data processing. Each point cloud element will store information about X-coordinate, Y-coordinate and Z-coordinate - depth. We assume that points with noise in depth domain should be modified. Output will be filtered patch with depth component in Z-coordinate. We propose to apply F-transform [7] method to solve this problem.

A. Fuzzy partition

F-transform technique requires defining of fuzzy partitions. It can be partitions according to Ruspini condition [12], called Ruspini partitions. Each center of this partition is placed uniformly. As a result, we obtain a regular grid of partition centers.

In the paper [13] uniform fuzzy partitions were modified to parametric generalized fuzzy partitions, with some relaxations of Ruspini condition. In this paper we use non-modified uniform fuzzy partitions.

In order to define two-dimension fuzzy partitions we use a generalization of single-dimension partitioning:

You can find an example of a triangle membership function in Fig. 4.

Let X be vector containing fixed nodes on interval $[a,b]$ so that $x_1=a, x_m=b$ and $m\geq 2$. The membership functions $A_1 ... A_m$ of the fuzzy sets should fulfill the following conditions [5] for $i = 1, ..., m$:

1. A_i is continuous;

2. A_i strictly increases on $[x_{i-1}, x_i]$ and strictly decreases on $[x_i, x_{i+1}]$;

3. $A_i : [a, b] \to [0,1], A_i(x_i) = 1$;

4. $A_i(x) = 0 \, if \; x \notin (x_{i-1}, x_{i+1})$, where we set $x_0 = a$ and $x_{m+1} = b$;

5. For all $x \in [a, b]$,

$$\sum_{i=1}^{m} A_i(x) = 1 \qquad (2)$$

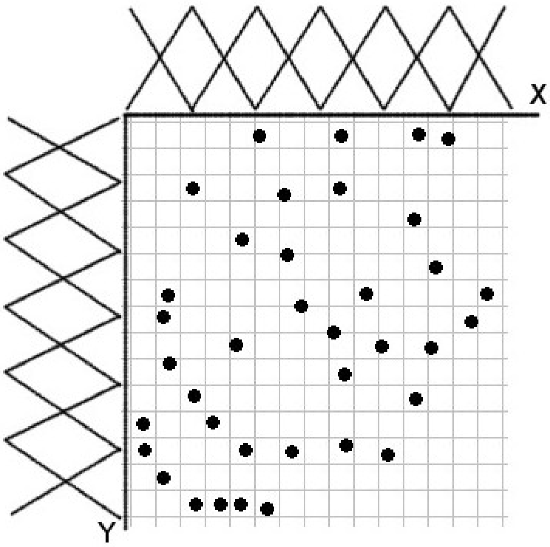

Fig. 4. Triangle membership functions for fuzzy uniform partitioning of 2D projection of point cloud.

The membership function can have different shapes. Shape of the function should be chosen by user. In this paper we use uniform fuzzy partitions.

Shape of the membership function and set of centers should be chosen at the beginning of computation.

B. FT-smoothing point cloud filtering

The membership function should be defined for each axis in order to create F-transform for two dimensions

We consider a set of points in the point cloud which can be represented as a function f , defined on $[M_a, M_b] \times [N_a, N_b]$, where M_a, M_b are boundaries of the patch on X-axis, and N_a, N_b - boundaries on Y-axis.

It is assumed, that values in coordinates (x, y) are the depth values, measured by Z-coordinate axis in point cloud belonging to the set P.

Let $A_1, ..., A_m$ and $B_1, ..., B_n$ be membership functions for partitions on X and Y axes. A point of the point cloud is covered by membership function for two dimensional fuzzy partition with membership functions A_i and B_j if $A_i(x) > 0$ and $B_j(y) > 0$.

Let p_l will be value of coordinate at x-axis, q_k -value of coordinate on y-axis, $f(p_l, q_k)$ will be dense (Z) value at (p_l, q_k) .

We define matrix $[F_{kl}]$ as a $m \times n$ matrix of real numbers, are called F-components.

F-transform of f with respect to $\{A_1, ... A_m\}$ and $\{B_1, ..., B_n\}$ for all $i = 1, ..., m, j = 1, ..., n$ will be based on equations proposed in [15] and can be re-formulated as follows:

$$F_{ij} = \frac{\sum_{l=M_a}^{M_b} \sum_{k=N_a}^{N_b} f(p_l q_k) A_i(p_l) B_j(q_k)}{\sum_{l=M_a}^{M_b} \sum_{k=N_a}^{N_b} A_i(p_l) B_j(q_k)} \qquad (3)$$

Fig. 5. Patch of 3D point cloud (mesh adde for visualisation) with noise.

Fig. 6. Filtered by FT-smoothing patch of 3D point cloud (mesh added for visualization).

The inverse transformation is applied to receive smoothed values:

$$\hat{f}(k, l) = \sum_{i=1}^{m} \sum_{j=1}^{n} F_{ij} A_i(p_l) B_j(q_k) \qquad (4)$$

Example of point cloud filtering represented in Fig. 5, 6.

This function $\hat{f}$ will be approximation of dense component represented in two-dimensional image f .

IV. Implementation And Results

The experiments were carried on several point clouds from Stanford Models dataset. The input point clouds are corrupted by simulated Gaussian noise.

The performance of the filter is measured using peak signal to noise ratio (PSNR).

$$PSNR = 10 \cdot \log_{10} \left(\frac{MAX^2}{MSE} \right) \tag{5}$$

Where MAX – maximum depth value in point cloud, and MSE – mean squared error. Table I contains the results of experiments.

The experiments show that FT-smoothing filter is better for Gaussian, Uniform, Rayleigh noise types. Quality of the FT-filter is average for the Poisson noise type. So, we observed that the proposed approach yielded satisfactory results and outperforms the other methods presented in the literature.

TABLE I. Experiment Results

Type of noise	**Denoised point cloud PSNR**				
	FT-smoothing filter	***Median filter***	***Average filter***	***Min filter***	***Max filter***
Gaussian noise	23.16	22.72	22.68	14.44	18.07
Uniform noise	25.20	24.59	24.38	16.23	16.83
Poisson noise	24.50	26.69	24.86	27.18	8.47
Rayleigh noise	22.78	22.33	22.35	18.01	14.68

V. Conclusion and Future Developments

In this paper various point cloud filtering methods have been discussed and analyzed. The proposed FT-smoothing approach was compared with the most common statistical filters. The current study can be extended by including of more numbers of noise types and complex filters from different filtering groups.

The proposed implementation of FT-smoothing filter uses–uniform fuzzy partitions. Further developments should include directions related to usage of non-uniform and generalized fuzzy partitions and its optimal parameter selection.

Optimal settings of membership functions selection for measured scene, using small sample from point cloud can be also defined as a perspective development in this domain.

References

[1] J. Fu, D. Miao, W. Yu, S. Wang, Y. Lu, and S. Li, “Kinect-like depth data compression,” IEEE Transactions on Multimedia, vol. 15, no. 6, pp. 1340–1352, 2013.

[2] J. Han, L. Shao, D. Xu, and J. Shotton, “Enhanced computer vision with microsoft kinect sensor: A review,” IEEE transactions on cybernetics, vol. 43, no. 5, pp. 1318–1334, 2013.

[3] A. Nurunnabi, G. West, and D. Belton, “Outlier detection and robust normal-curvature estimation in mobile laser scanning 3D point cloud data,” Pattern Recognition, vol. 48, no. 4, pp. 1404–1419, 2015.

[4] X.-F. Han, J. S. Jin, M.-J. Wang, W. Jiang, L. Gao, and L. Xiao, “A review of algorithms for filtering the 3D point cloud,” Signal Processing: Image Communication, vol. 57, pp. 103–112, 2017.

[5] I. Perfilieva, “Fuzzy transforms: Theory and applications,” Fuzzy sets and systems, vol. 157, no. 8, pp. 993–1023, 2006.

[6] L. A. Zadeh, “Fuzzy sets,” in Fuzzy Sets, Fuzzy Logic, And Fuzzy Systems: Selected Papers by Lotfi A Zadeh, World Scientific, 1996, pp. 394–432.

[7] I. Perfilieva and R. Valášek, “Fuzzy transforms in removing noise,” in Computational Intelligence, Theory and Applications, Springer, 2005, pp. 221–230.

[8] Y. Nie and K. E. Barner, “The fuzzy transformation and its applications in image processing,” IEEE Transactions on Image processing, vol. 15, no. 4, pp. 910–927, 2006.

[9] K. Hirota and W. Pedrycz, “Fuzzy relational compression,” IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), vol. 29, no. 3, pp. 407–415, 1999.

[10] F. Di Martino, V. Loia, I. Perfilieva, and S. Sessa, “An image coding/decoding method based on direct and inverse fuzzy transforms,” International Journal of Approximate Reasoning, vol. 48, no. 1, pp. 110–131, 2008.

[11] M. Levoy, J. Gerth, B. Curless, and K. Pull, “The Stanford 3D scanning repository,” URL http://www-graphics. stanford. edu/data/3dscanrep, 2005.

[12] G. Turk and M. Levoy, “Zippered polygon meshes from range images,” in Proceedings of the 21st annual conference on Computer graphics and interactive techniques, pp. 311–318, 1994.

[13] E. H. Ruspini, “A new approach to clustering,” Information and control, vol. 15, no. 1, pp. 22–32, 1969.

[14] L. Stefanini, “F-transform with parametric generalized fuzzy partitions,” Fuzzy Sets and Systems, vol. 180, no. 1, pp. 98–120, 2011.

[15] P. Vlašánek and I. Perfilieva, “Image reconstruction with usage of the F-Transform,” in International Joint Conference CISIS’12-ICEUTE’ 12-SOCO’ 12 Special Sessions, pp. 507–514, 2013.