A Comparative Study of Anisotropic Diffusion and Bilateral Filter for Derivation of Point Cloud Using Fast Marching

Muhamad Yazid Che Abdullah, Nurul ‘Atifah Mohamad Jais, Habibollah Haron
Faculty of Computing
Universiti Teknologi Malaysia
81310 Skudai, Johor Bahru, Malaysia
yazidyt94@gmail.com, nurultifa.jais@gmail.com, habib@utm.my

Abstract- In many applications computers analyse images are often tainted by noise. Also, sometimes the result can be poor. Partial differential equation (PDE) based image processing techniques are mostly used in smoothing, enhance and restoration intent such as anisotropic diffusion and fast marching. In addition, bilateral filter also one of image processing technique that often link with PDE. Anisotropic diffusion and bilateral filter are often use as denoising method and also for edge enhancement. Even though fast marching technique can be used for denoising and enhances like the mentioned techniques but for this paper, we only focus on derivation of point cloud.

Keywords-anisotropic diffusion; bilateral filter; fast marching; point cloud; denoise; edge enhancement

I. INTRODUCTION

Today, image processing techniques has become progressively significant in many types of applications. PDE are mostly used in image processing field especially in denoising and edge enhancement. PDE has become an important branch and give benefits to co -researchers in related fields to become the focus of attention.

There are many techniques in PDE-based image processing technique such as anisotropic diffusion, level set and fast marching [1 - 3]. Bilateral filter also closely related with PDE [4]. Also, there are many techniques for derivation of point cloud based on enhanced image such as fast marching, local mesh method and level set active contours [7 - 9].

In this paper, we focus on the comparative study of modified anisotropic diffusion and directional joint bilateral filter for image processing. They were nonlinear based filtering method. Although Fast Marching techniques is also used for image processing, in this paper we only scope the discussion on derivation of point cloud using extended version of Fast Marching technique named Fast Marching Farthest Point Sampling (FastFPS) for implied surfaces and point cloud.

Modified anisotropic diffusion method is a modified version of Perona-Malik Module [1]. Vertex selection is used to change the original Anisotropic Diffusion into more robust Anisotropic Diffusion.

An extended version of Joint Bilateral Filter (JBF) that combined with Guided Filter (GF) is called Directional joint Bilateral Filter (DJBF). This method is specialized in utilizing the aligned colour image and the depth map. This method proposed to recover the blurring effect form by previous method by treating the noise and holes separately.

Fast marching farthest point sampling (FastFPS) is a method to derive point cloud. It focuses on 3D image. The algorithm of FastFPS is easy to implement and memory effective. Also, it allows multi resolution that represents of the input which is point set.

The outline of this paper is divided into six sections including introduction. In Section II, we describe the modified anisotropic diffusion including the previous work of anisotropic diffusion and the modification of the equation. In Section III, we describe the directional joint bilateral filter which need to go through two process which are filtering and filling. In Section IV, we describes the process of FastFPS for derivation of point cloud. In Section V, we describe the conclusion of this comparative study based on these three mentioned techniques. Finally, in Section VI, we describe our future work to utilise and extend these mentioned techniques in derivation of point cloud.

II. MODIFIED ANISOTROPIC DIFFUSION METHOD

Anisotropic diffusion (AD) is sometime been called as Perona-Malik Module. This method were proposed as a non-linear diffusion to prevent the localization and blurring effects of linear diffusion filtering [2]. This module were replaced the classical isotropic diffusion by introducing the concept of gradient in diffusion constant.

$$AD = div(c(x,y,t)\ \cdot \nabla I) = c(x,y,t) \cdot \Delta \mathrm{I} + \nabla_c \cdot \nabla I \qquad (1)$$

According to [5], from Equation (1), the gradient and Laplacian operator with respect to the space variables which is divergence operator *div* with *r*. Equation (1) is also stated as

$$I_g^{t+1} = I_g^t + \lambda \cdot \sum_{p\in\eta_4(g)} g\,(I_p^t - I_g^t) \cdot (I_p^t - I_g^t) \qquad (2)$$

where I as the image, s and p as the pixel positions, $\eta_4(s)$ as the 4-neighborhood of the pixel at s, λ as the controlling scalar for the rate of diffusion, t as the discrete time step, and *g(x)* as an edge stopping function [5].

Modified AD was introduced by Adhi Wibowo and Kim [5]. Since non-linear characteristics does not have in a 3D-face point, the AD method is used to enhance it. Based on [5], data input from the Curtin Faces Database were proposed to be used for smooth and enhance the 3D-face point cloud. The data obtained from Microsoft Kinect give result which is red color, green color, blue color, and depth (RGB-D) data.

To make a 3D object point cloud, a 2D and 3D image point cloud need to be use in the same time. I in Equation (2), *g(x)* can be replaced by D_c^k . As the z-coordinate has a huge impact on the quality of a 3D face point cloud, therefore they only need to be handled using AD. However, there are several point cloud values that are computed incompletely using AD due to the cropping process. Thus, the original method require to be transformed as Equation (3) and Equation (4)

$$D_{c,s}^{3,t+1} = D_{c,s}^{3,t} + xr,$$

$$xr = \left[\leftthreetimes \cdot \sum_{p\in\eta_4(s)} g\left(D_{c,p}^{3,t} - D_{c,s}^{3,t}\right)\cdot\left(D_{c,p}^{3,t} - D_{c,s}^{3,t}\right)\right]\cdot VS \quad (3)$$

$$VS(0) = \left\{\leftthreetimes \cdot \sum_{p\in\eta_4(s)} g\left(D_{c,p}^{3,t} - D_{c,s}^{3,t}\right)\cdot\left(D_{c,p}^{3,t} - D_{c,s}^{3,t}\right)\right\}_{p,s=NaN} \quad (4)$$

(4) where *VS* is defined as selected vertices and *VS(0)* is defined as the *VS* that contain value of *NaN* and change it equal to zero.

This modified version only use basic concepts of convolution and filtering that do not require a complicated process. Fig 4 shows the result of this modified method.

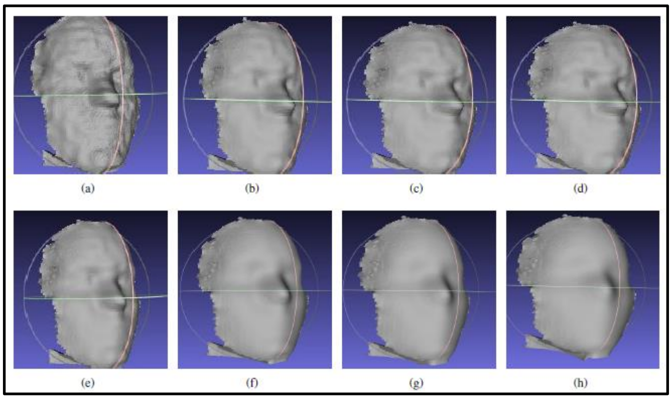


Fig 1. The 3D face point cloud smoothing result arranged by the number of iterations. (a) with two iterations, (b) with four iterations, (c) with six iterations, (d) with eight iterations, (e) with ten iterations, (f) with twenty iterations, (g) with thirty iterations. [5]

Based on the result mention in Fig 1, the modified method is more robust than the previous version of AD. From the result, it proves that this new method give better performance compared to the previous version of AD method. For the result in the smoothing process, the shape of the image and quantity of vertices is remain the same with the original input.

III. DIRECTIONAL JOINT BILATERAL FILTER

Using a cheap Kinect sensor from the depth maps, it commonly produced imperfect depth data, inevitable noise and holes. The edge information is taken to fill the holes. Therefore, a new method named Directional Gaussian Filter were suggested to take the edge information but the accuracy is not enough. Therefore, JBF and GF were introduced so that the utilization can be made for the aligned colour image and the depth map but the distorting effect (blurring) still remain constant when there is no change of intensity difference around depth discontinuities.

There are two processes in this method namely filtering and filling. Filtering process will be used for denoising inside non-hole region, while filling process involved filling to the empty pixels in the hole region.

Directional joint bilateral filter (DJBF) were proposed to recover these imperfections by treating the noise and holes separately [6]. Firstly, each pixel was classified into two, which is a hole pixel and non-hole pixel. Hole pixels as shown in Fig 2 is the Kinect tags "non-available" for pixels that do not return the signal which been labelled as "non-available" pixels. Beside, non-hole region is generally noisy. Therefore, methods in Table 1 are used to fill these imperfections.

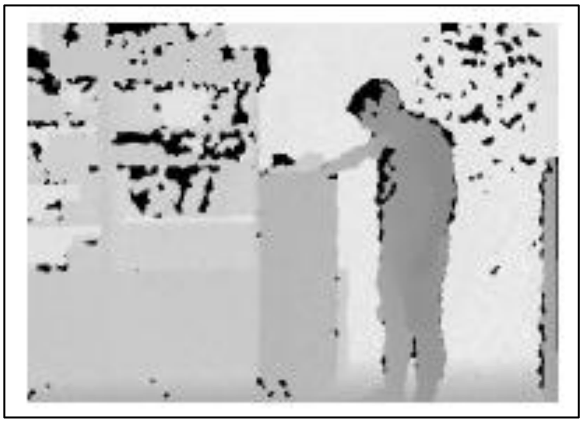


Fig 2. The pixels of the depth map in hole region that are represented by the black pixels ("non-available" Kinect tags.) [6]

TABLE 1. The filters that are used [6]

	Non-Edge	**Edge**
Non-Hole	Joint Trilateral Filter (JTF)	Directional Joint Bilateral Filter (DJBF)
Hole	Partial Directional Joint Bilateral Filter (PDJBF)	Directional Joint Bilateral Filter (DJBF)

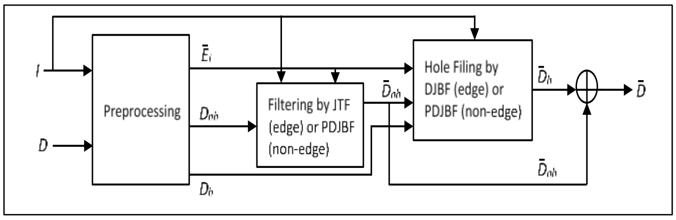

I
D
Preprocessing
$\bar{E}_i$
D_{nh}
D_h
Filtering by JTF (edge) or PDJBF (non-edge)
$\bar{D}_{nh}$
Hole Filing by DJBF (edge) or PDJBF (non-edge)
$\bar{D}_h$
$\bar{D}$
$\bar{D}_{nh}$

Fig 3. The proposed filtering depth image in block diagram [6]

The depth sensor of the Kinect contains two kinds of limitations which is noisy measurements for depth and holes for unmeasured depth. Therefore, a modification is needed for improvement of the incomplete depth image by taking separate filters for hole and non-hole region. Fig 3 above displayed that the depth image, D is categorized into hole region D_h and non-hole region D_{nh}. After that, the filters are used in pixels of the non-hole region to eliminate the noise in depth map, producing $\bar{D}_{nh}$, and after that the hole-filling scheme is applied to fill in the holes region, producing $\bar{D}_h$. For the last depth $\bar{D}$, it merged of $\bar{D}_{nh}$ and $\bar{D}_h$.

The following section explains two processes that are mentioned in paragraph 2 which is Filtering Non-Hole Region and Hole Filling followed by result for the DJBF.

A. Filtering Non-hole Region

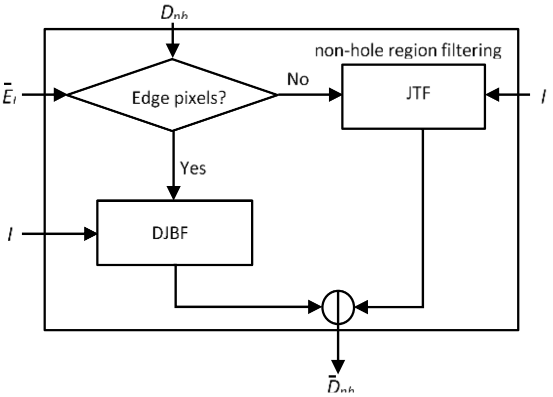


Fig 4. The flowchart for the filtering in the non-hole region. [6]

Fig 4 displays the flowchart for filtering in the non-hole region. Due to the noise and the holes in the depth image that are performed individually, the depth image needs to be classified into non-hole (D_{nh}) and hole (D_h) areas. Equation (5) is the equation for DJBF.

$$\bar{D}_{nh} = \sum_{q\epsilon\Phi^P} D_{nh}(q) f_{ds}^d(q_x - p_x, q_y - p_y) f_r^c(I^p - I^q) \quad (5)$$

In the depth images, the JTF method operates the best. Besides using the JBF, it is recommended to use JTF that is adopted for the pixels in the non-hole region. Equation (6) is the equation for JTF.

$$D_{nh} = \sum_{q\epsilon\Phi^P} D_{nh}(q) f_s^d(q_x - p_x, q_y - p_y) f_r^c(I^p - I^q) f_r^d(D_{nh}(p) - D_{nh}(q)) \quad (6)$$

B. Filling Hole Region

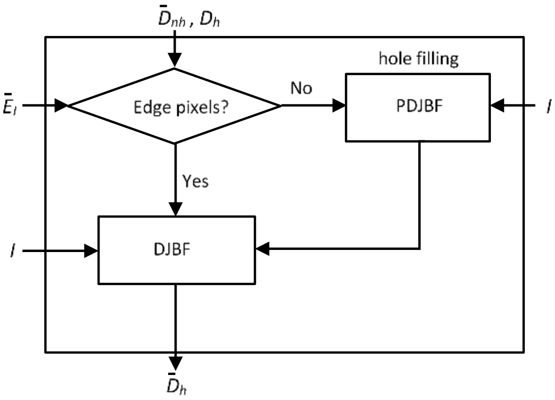


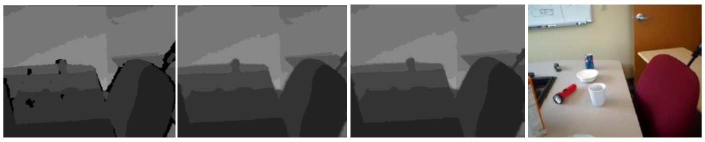
Fig 5. The filling hole region in a flowchart [6]

Fig 5 depicts the filling hole region in a flowchart. Note that, holes usually appear at the object boundaries in the Kinect depth maps. Therefore, in hole filling, the directions of object boundaries should be considered. The direction of the hole filling must be determined first so that the hole pixels can be filled in the non-edge region. Then, a new technique called Partial Directional Joint Bilateral Filter (PDJBF) is introduced. The PDJBF uses the Directional Gaussian Filter that are able to smooth the images while maintaining the details of the edge. The PDJBF can be defined as in Equation (7). The PDJBF is equal with the DJBF except for the filter that supports $\Omega^{p.}$

$$D_f^p = \sum_{q\epsilon\Omega^P} D_m^q f_{ds}^d(q_x - p_x, q_y - p_y) f_r^c(I^p - I^q) \quad (7)$$

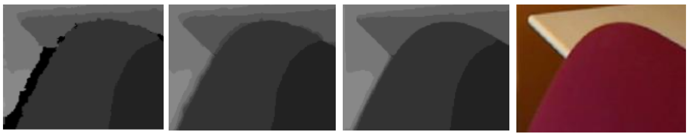
C. Result

In this part, the result acquired using the DJBF is compared to JBF. Fig 7 and Fig 8 shows the result for this experiment.



Original JBF DJBF Colour Image

Fig 7. Results between original, JBF, DJBF and colour image [6]



Original JBF DJBF Colour Image

Fig 8. Results that is zoom-in in Fig 7 which is original, JBF, DJBF and colour image [6]

The DJBF method explained that by the combination of the adaptive directional filter kernel with adaptive filter range provides the best results for hole filling especially for the hole pixels that are located near to the object boundaries. To determine the quantitative of the image, DJBF method was developed using the synthetic test images, while for quantitative, it is developed using the Kinect test images for qualitative.

IV. FAST MARCHING FARTHEST POINT SAMPLING

The previous sections discusses on the image processing and enhancement using two techniques. This section describes the point cloud derivation according to the enhanced image. Fast marching (FM) is a very effective method to solve front propagation problems that can be defined as PDE's boundary value. To compute the distance map across a smooth sampling domain is not an easy task. The problem is that it can be presented in the form of mathematical equation. For instance, PDE and outline of the FM method towards estimation of its solution.

Farthest point sampling (FPS) is a new method that is used to simply the point sampled geometry without any prior surface reconstruction. It is develop by placing the next sample point repeatedly in the middle of the smallest known area of the sampling domain for reducing any reconstruction error [10, 11]..

Based on [7], Moenning and Dodgson design a feature-sensitive simplification or coarse-to-fine uniform algorithm which the density can be control by user. This method had been named as FastFPS. It only focusing on a 3D image point cloud. The algorithm of FastFPS then computes (discrete) Voronoi diagrams which mean it is in the form of weighted distance maps that incrementally right across the input for point set. This is achieved efficiently using Memoli and Sapiro's recent extension of the original FM and level set method without the need for any previous surface reconstruction [7].

A. The Algorithm

The FastFPS can be summarise as follows [7]:

1) In a Cartesian grid, embed the given point cloud sufficiently large so that it can be used for an offset band of size r. Given an initial point cloud subset $S \in \Omega r$, $n = |S| \geq 2$, by using extended FM, compute bounded Voronoi Diagram (*BVD)(S)* by propagating fronts with speed $\dot{F}(p_i)$ from the points towards outwards using extended FM. The Voronoi vertices' arrival times are stored in a max-heap.

2) From the max-heap, extract the root to obtain s_n+1. $S' = S \cup \{s_n+1\}$. By using extended FM and a single min-heap, compute *BVD(S')* by propagating a front locally from s_n+1 towards outwards.

3) In the min-heap, correct the arrival times of updated grid points. After that, insert the vertices of the bounded Voronoi cell of *sn+1*, $BV\ (s_n+1)$, in the max-heap. Then, get rid of the obsolete Voronoi vertices of the neighbours of $BV\ (s_n+1)$ from the max-heap.

4) If the user-controlled point density ρ or the target model size $N_2 < N_1$ has been reached, loop from 2).

B. The Evaluation

The evaluation of Fast FPS is based on density. To allow the user to control the density, the point cloud is simplified uniformly until the density requirement is met. Fig 9 depicts the effect of different density value with user-controlled of the density values.

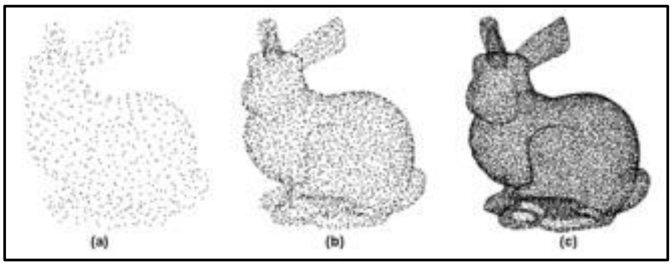


Fig 9. The effect of different density value, ρ, with user-controlled minimum on 3D point set. (a) ρ = 10 (b) ρ = 5 (c) ρ = 2.5. [7]

To simply, the radius of the union of balls that producing the thin offset band Ω_r is set to a constant $r = 2$ [4]. The simplified point sets used by FastFPS algorithm for various levels of detail and the renderings of the corresponding reconstructed triangular meshes (refer Fig 10).

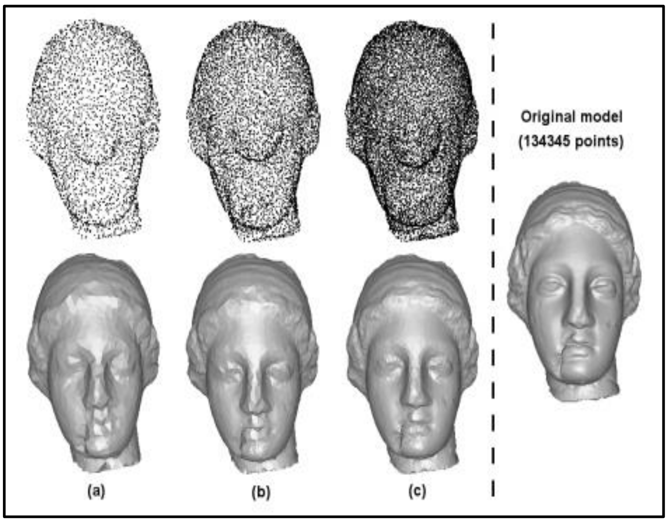


Fig 10. Point sets from the simplified Venus and the representations of their mesh reconstructions used by FastFPS algorithm which is 97.5% simplified (ρ = 5.00) in (a), 95.0% simplified (ρ = 2.50) in (b) and 90.0% simplified (ρ = 1.25) in (c). [7]

V. CONCLUSION

Image processing techniques are mostly used for enhancing and recovery purposes. Here we have reviewed and presented a modified version of anisotropic diffusion named as modified anisotropic diffusion method which use and bilateral filter named as directional joint bilateral filter in denoising the noise present in the image and enhance the edge of the region for further use in image processing. FastFPS also been reviewed as a point cloud derivation method.

The comparative study shows that the different between modified anisotropic diffusion and directional joint bilateral filter is based on the algorithm. In modified anisotropic diffusion, vertex selection and basic concepts of convolution and filtering is used to become more robust that the original version. For the result in the smoothing process, the shape of the image and number of vertices is remain the same as the original input.

For DJBF, it depends on edge's direction and also the position between the edges that are extracted from the filtered pixel and colour image. This method is developed to solve hole filling problem of the Kinect depth images as it uses the adaptive directional filter kernel with adaptive filter range that produces the best hole filling results especially for the hole pixels that are located near to the object boundaries.

FastFPS is an efficient way and easy to implement method to derive a 3D image point cloud. The evaluation of Fast FPS is based on density and also the user can control the density of the point cloud until the requirement is met.

VI. FUTURE WORK

AD and BF has shown good result compared to previous image enhancement method as listed in Section 1. Therefore, to utilise and extend these two techniques in derivation of point cloud, it is suggested to hybrid anisotropic diffusion with fast marching, or bilateral filter with fast marching. The hybridization of these techniques is expected to produce higher density of point cloud. As far as literature concern, the hybrid of these techniques has not yet developed in derivation of point cloud.

ACKNOWLEDGEMENTS

This research is supported by the PDE-GIR project which has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie grant agreement No 778035.

REFERENCES

[1] J.A Sethian, "Level set Methods and fast marching methods," Evolving Computational Geometry, Fluid Mechnanics, Computer Vision, and Material Science, Cambridge Univ. Press, 1999.

[2] P. Perona, J. Malik, "Scale space and edge detection using anisotropic diffusion," Proc. IEEE Comp. Soc. Workshop on Computer Vision, Nov. 30 – Dec. 2, IEEE Computer Society Press, Washington, 16–22, 1987.

[3] J. Weickert, "Anisotropic diffusion in computer vision," Teubner-Stuttgart, 1998.

[4] S. Paris, P. Kornprobst, J. Tumblin, F. Durand, "A gentle introduction to bilateral filtering and its applications," Proc. SIGGRAPH, International Conference on Computer Graphics and Interactive Techniques, August 11-15, 2008.

[5] S. Adhi Wibowo, S. Kim, "Three-dimensional face point cloud smoothing based on modified anisotropic diffusion method," International Journal of Fuzzy Logic and Intelligent Systems,Vol. 14, No. 2, June, 2014, pp. 84-90.

[6] Le A.V, Jung S-W, Won C.S, "Directional Joint Bilateral Filter for Depth Image," Sensors, 2014, 14, 11362-11378.

[7] Moenning C, Dodgson N.A, "A new point cloud simplification algorithm," 3rd IASTED International Conference on Visualization, Imaging, Image Processing (VIIP 2003) Benalmadena, Spain, Sep. 8 – 10, 2003.

[8] Rongjie L., Jiang L., Hongkai Z., "A local messh method for solving PDEs on point cloud," Inverse Problem and Imaging, Vol. 7, 3, 737-755, 2013.

[9] Hon P.H, Yunmei C., Huafeng L., Pengcheng S., " Level set active contours on unstructed point cloud," Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol.2. 2. 655 – 662, July, 2005.

[10] T. F. Gonzalez, "Clustering to minimize the maximum intercluster distance," Theoretical Computer Science, 38, 293–306, 1985.

[11] Y. Eldar, M. Lindenbaum, M. Porat and Y. Y. Zeevi, "The farthest point strategy for progressive image sampling," IEEE Trans. on Image Processing, 6(9), 1305–1315, 1997.