Signal Processing: Image Communication 57 (2017) 103–112

Contents lists available at ScienceDirect

Signal Processing: Image Communication

journal homepage: www.elsevier.com/locate/image

A review of algorithms for filtering the 3D point cloud

Xian-Feng Han *, Jesse S. Jin, Ming-Jie Wang, Wei Jiang, Lei Gao, Liping Xiao

School of Computer Software, Tianjin University, 30072, Tianjin, China
National Key Laboratory of Science and Technology on Aerospace Intelligent Control, Beijing 100085, China

ARTICLE INFO

Keywords:
3D point cloud
Filtering methods
Feature-preserving
Noise reduction

ABSTRACT

In recent years, 3D point cloud has gained increasing attention as a new representation for objects. However, the raw point cloud is often noisy and contains outliers. Therefore, it is crucial to remove the noise and outliers from the point cloud while preserving the features, in particular, its fine details. This paper makes an attempt to present a comprehensive analysis of the state-of-the-art methods for filtering point cloud. The existing methods are categorized into seven classes, which concentrate on their common and obvious traits. An experimental evaluation is also performed to demonstrate robustness, effectiveness and computational efficiency of several methods used widely in practice.

1. Introduction

The 3D point cloud [1–3], a new primitive representation for objects, has became increasingly prevalent in many research fields [2], such as object recognition [4] and reconstruction [5,6], due to its simplicity, flexibility and powerful representation capability. In contrast to triangle meshes, the point cloud does not require to store or maintain the polygonal-mesh connectivity [7] or topological consistency [8]. Processing and manipulating point cloud therefore can demonstrate better performance and lower overhead. These prominent advantages make the research on processing point cloud a hot topic.

The rapid development of low-cost sensors, such as Kinect [9–11] and time of flight cameras [5,12], makes it easy to obtain point cloud for growing communities. The point cloud acquired with these sensors, however, inevitably suffers from noise contamination and contains outliers [13,14] due to the limitations of sensors [5], the inherent noise of the acquisition device [15], the lighting or reflective nature of the surface or artifact in the scene [16]. Therefore, it is necessary to perform filtering operations on raw point clouds to obtain accurate point clouds that are suitable for further processing.

In recent years, although a large number of methods contributing to 3D filtering have been proposed, most of these are devised for meshes and only a few approaches directly operate on point cloud. In addition, there is no survey paper giving an insightful analysis of these filtering methods for point cloud.

Compared with the existing literature, the main contributions of this work are as follows: (i) To the best of our knowledge, this is the first review paper in the literature that focuses on algorithms for filtering 3D point cloud at present. (ii) This paper provides readers with a comprehensive review of the state-of-the-art methods covered in early work. (iii) A comparative summary of traits of these methods is demonstrated in table form. (iv) This paper carries out an experiment concerning on performance comparison of several widely used methods.

The remainder of this paper is organized as follows. Section 2 presents an overview of filtering approaches for 3D point cloud. And then experimental results and discussion are illustrated in Section 3. Conclusions are drawn in Section 4.

2. Methods for filtering point cloud

Filtering is an area of intensive research and the crucial step of the processing pipeline for a wide range of applications. The main filtering approaches for 3D point cloud can be categorized into the following seven groups, where four classifications (statistical-based, neighborhood-based, projection-based and PDEs-based filtering) are from [17].

2.1. Statistical-based filtering techniques

In the context of filtering point cloud, many techniques utilize the adaptation of the statistical conceptions, which are suitable for the nature of the point cloud.

* Corresponding author at: School of Computer Software, Tianjin University, 30072, Tianjin, China.
E-mail addresses: hanxianf@163.com (X.-F. Han), jinsheng@tju.edu.cn (J.S. Jin), 18768126670@163.com (M.-J. Wang), jiangweitju@163.com (W. Jiang), thrstone@sina.cn (L. Gao), xlp027@sina.com (L. Xiao).

http://dx.doi.org/10.1016/j.image.2017.05.009
Received 22 November 2016; Received in revised form 28 February 2017; Accepted 15 May 2017
Available online 22 May 2017
0923-5965/

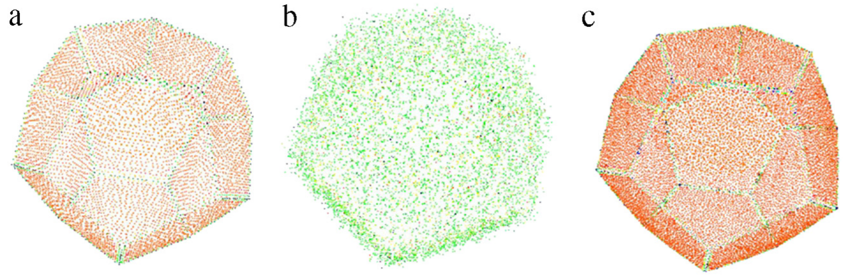

Fig. 1. Denoise point sets via L0 minimization. (a) Original model; (b) input; (c) L0 minimization, primitively shown in [23].

Schall et al. [18] filtered point cloud using a kernel based clustering approach. They first accumulated the local likelihoods L_i computed on every point p_i to define the likelihood function L modeling the probability for the noisy point cloud. Next, they moved the points to positions of high probability utilizing an iterative scheme motivated by mean shift technique to smooth the point cloud. This method achieves effectiveness in filtering and robustness in outlier detection. However, sharp features are not treated with emphasis in their method.

Narváez et al. [15] proposed a new weighted variant of the principal component analysis method for denoising point cloud, which used weighting factors assignment by inversely proportional repartition of the sum of distance to the mean. Then, the factors and the weighted mean are used to estimate a weighted covariance matrix. By realizing an eigen-analysis of the matrix, a fitting plane, expanded by the eigen-vectors corresponding to the largest eigen-values, and a normal vector to this plane oriented in the direction of eigen-vector corresponding to the smallest eigen-value are obtained at each point. The variations make the algorithm robust to the noise and outliers. In addition, the operator $p' = p + tn_p$ [19] is applied to shift the mean along the normal direction to preserve shape features.

Jenke et al. [20] first employed Bayesian statistics for denoising point cloud. They found a measurement model $P(D|S)$, which specified the probability distribution of estimated point cloud S agreeing with measured data D. Then, they defined three prior probabilities, such as density priors, smoothness priors and priors for sharp features, to form $P(S) = \frac{1}{Z} P_{density}(S) P_{smooth}(S) P_{discrete}(S) \cdot w(S)$, Z was a normalization constant. Finally, they maximized a posteriori $P(S|D)$ to remove noise while preserving features.

$$S_{MAP} = argmax_S P(S|D) = argmax_S P(D|S) P(S) \quad (1)$$

Kalogerakis et al. [21] provided a robust statistical framework to filter point clouds. In their framework, an Iteratively Least Squares (IRLS) approach estimates curvature tensor and assigns weights to samples at each iteration to refine each neighborhood around every point. The computed curvatures and the final statistical weights are utilized to correct normal. The robustly estimated curvatures and normal can drive the outlier rejection and denoise point cloud in a feature-preserving manner based on a global energy minimization process.

Avron et al. [22] introduced L_1-*sparsity* paradigm to denoise the point cloud. Firstly, a re-weighted L_1 minimization process is used to restore point orientations. Then, point position is reconstructed by assuming a local planarity criterion so as to preserve shape features. Although, this method can achieve reasonable results, points on an edge are sometimes not faithfully recovered [23]. Meanwhile, since L_0 is a sparser solution than L_1, Sun et al. provided an L_0 minimization method, which is directly used to denoise point cloud by applying a similar L_0 optimization procedure to estimate normals followed by repositioning points along the normal directions in order to better maintain the sharp features (see Fig. 1).

Orts-Escolano et al. [24] first used a 3D filtering and downsampling technique based on Growing Neural Gas (GNG) [25–27] network. This is a growth process to produce a GNG network to represent a raw point cloud using a set of 3D neurons and interconnection among them. This method can preserve the topology of point cloud and deal with outliers in point cloud. Therefore, filtered point cloud using GNG can improve keypoints detection performance [28] and yield better recognition results [4,29]. This method produced a GNG network mapping to the point cloud (shown in Fig. 2).

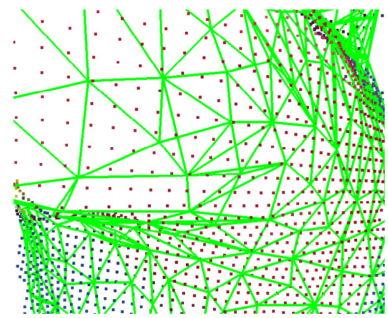

Fig. 2. GNN representation of point cloud, primitively shown in [4].

2.2. Neighborhood-based filtering techniques

Neighborhood-based filtering techniques determine the filtered position of a point using similarity measures between a point and its neighborhood which has a strong influence on the efficiency and effectiveness of the filtering approach [17]. As described in the following methods, the similarity can be defined by positions of points, normals or regions.

The bilateral filter, originally introduced by Tomasi and Manduchi [30], is an edge-preserving [31] smoothing filter, which is extended to 3D meshed denoising [32–34]. However, these methods require a mesh generation process, which itself suffers from noise [35]. In order to tackle this problem, the bilateral filter is applied directly on the point cloud [6,36–38] based on point positions and intensity. The w_s and w_r are the spatial and range weight, respectively,

$$w_s = \exp\left(-\frac{(i-x)^2 + (j-y)^2}{2\sigma_s^2}\right) \quad (2)$$

$$w_r = \exp\left(-\frac{(I(i,j) - I(x,y))^2}{2\sigma_r^2}\right) \quad (3)$$

where (i,j) is the neighborhood of (x,y), $I(i,j)$ presents the intensity at (x,y), σ_s and σ_r are the standards of Gaussian functions.

In order to reduce time complexity, Xu et al. [39] replaced the weight of gray domain in the bilateral filter with a binary function (4) to achieve a better performance. However, this kind of filters deals with the point cloud containing intensity components. As a consequence, normal [35,40–43], being as one of the important attributes of point cloud, is considered in the process of bilateral filter of which the weight