Signal Processing: Image Communication 51 (2017) 1–12

Contents lists available at ScienceDirect

Signal Processing: Image Communication

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

Image-based object reconstruction using run-length representation☆

Sung Soo Hwang[1], Hee-Dong Kim, Tae Young Jang, Jisung Yoo, Sujung Kim[2], Kyunghyun Paeng, Seong Dae Kim*

Department of Electrical Engineering, Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Republic of Korea

ARTICLE INFO

Keywords:
3D object reconstruction
Multi-view images
Run-length representation
Visual hull computation
Mesh generation
Texture mapping

ABSTRACT

This paper presents an image-based object reconstruction with a low memory footprint using run-length representation. While conventional volume-based approaches, which utilize voxels as primitives, are intuitive and easy to manipulate 3D data, they require a large amount of memory and computation during the reconstruction process. To overcome these burdens, this paper uses 3D runs to represent a 3D object and reconstructs each 3D run from multi-view silhouettes with a small amount of memory. The proposed geometry reconstruction is also computationally inexpensive, as it processes multiple voxels simultaneously. And for the compatibility with the conventional data formats, generation of polygonal 3D meshes from the reconstructed 3D runs is proposed as well. Lastly, texture mapping is proposed to additionally reduce the amount of memory for object reconstruction. The proposed reconstruction scheme has been simulated using various types of multi-view datasets. The results show that the proposed method performs object reconstruction with a smaller amount of memory and computation than voxel-based approaches.

1. Introduction

Three-dimensional object models, or 3D models, are widely used these days to generate realistic and immersive contents. They are also used for image matching such as landmark search [50]. And as the demand for utilizing 3D models is expected to increase, various researches on 3D models, e.g., transmission of 3D models [51] and 3D object retrieval [52] have been conducted.

These 3D models were, traditionally, generated by utilizing computer graphics techniques. However, a variety of alternatives has been introduced recently. Image-based object reconstruction is one of the popular alternatives to generate 3D models. By utilizing multiple cameras to capture multi-view images of an object, image-based object reconstruction generates a 3D model the object. Unlike the computer graphics techniques that produce 3D model of a virtual object, image-based object reconstruction can generate a 3D model of a real object.

Methods for image-based object reconstruction can be classified depending on the primitives they use. As several primitives have been proposed, (including points, voxels, surfels, and 3D meshes), a variety of researches have been conducted [9–42]. Among them, volume-based approaches, which are traditional, represent a 3D space to be constructed by a set of voxels. Each voxel is then classified as either an object voxel (a voxel inside a 3D object) or a non-object voxel (a voxel outside a 3D object). Compared to other representation methods, volume-based approaches are intuitive and are favorable to several post processing of 3D data such as calculation of volume [1], Boolean operation [2] and collision detection [3]. One of the drawbacks of volume-based approaches, however, is that they generally require a large amount of memory during the reconstruction process. Even though this memory is released after the reconstruction process, memory allocation itself can be a problem when high resolution 3D data are reconstructed. Furthermore, while classification of a voxel as either an object voxel or a non-object voxel is simple, this process should be applied to each voxel. Hence, the computational cost of voxel-based reconstruction is high.

In order to reconstruct a 3D object with a small amount of memory and computation while keeping the merits of volume-based approaches intact, we extend the concept of run-length representation to 3D space and propose the use of 3D runs to represent a 3D object [4] (*i.e., 3D runs* are used as *primitives* for object representation). In run-length representation, a two-dimensional (2D) run is defined as a sequence of pixels that are inside a 2D binary object in a horizontal (or vertical)

☆ Manuscript received February 22, 2016. This work (2011-0016298) was supported by Mid-career Researcher Program through NRF grant funded by MEST.

* Corresponding author.
E-mail addresses: sshwang@handong.edu (S.S. Hwang), khdong98@kaist.ac.kr (H.-D. Kim), tyjang1020@kaist.ac.kr (T.Y. Jang), yjs1018@kaist.ac.kr (J. Yoo), sujung.susanna.kim@naver.corp.com (S. Kim), paeng87@kaist.ac.kr (K. Paeng), sdkim@kaist.ac.kr (S.D. Kim).
[1] Currently with Schools of Computer Science and Electronic Engineering, Handong Global University, Pohang, Republic of Korea.
[2] Currently with Naver labs, Naver Corp., Sungnam, Republic of Korea.

http://dx.doi.org/10.1016/j.image.2016.11.002
Received 21 February 2016; Received in revised form 8 November 2016; Accepted 13 November 2016
Available online 17 November 2016
0923-5965/

direction, and it represents the object as a union of 2D runs. Run-length representation is simple, yet it efficiently and losslessly reduces the data size to represent a binary object. Similar to 2D case, a 3D run is defined as a sequence of object voxels in a certain direction, and a 3D object is represented with 3D runs. By representing a 3D object with 3D runs, the data size of an object can be reduced. Various post processing and 3D object analysis schemes designed for voxels are also applicable, since converting 3D runs to voxels is straightforward. Moreover, previous works proposed several operations of 3D data based on 3D run-length representation [5,6]. Therefore, 3D run-length representation can be useful for compact representation and easy manipulation of 3D object data.

In order to utilize 3D runs as primitives in image-based object reconstruction, it is necessary to generate 3D runs of an object directly from multi-view images of the object (Even though 3D runs are easily obtained from voxel data, this only increases the computational cost to the voxel-based object reconstruction). For the direct generation of 3D runs, this paper proposes a silhouette-based geometry reconstruction method. To this end, we first generate 3D runs considering all the voxels are object voxels. Thereafter, fragments of each 3D run which are determined to be inside the object are produced by using the silhouettes of an object. 3D runs of an object are finally computed as sequence of voxels bounded by these fragments. As multiple voxels are simultaneously identified as either object voxels or non-object voxels, the computational cost of geometry reconstruction can be reduced.

Since polygonal meshes are widely used in computer graphics to represent a 3D object, for compatibility, 3D mesh generation from 3D runs is important. For mesh generation, it requires both the identification of surface voxels which are located on the boundary of the object and the estimation of their connectivity. For that purpose, slices of a volume data are extracted from 3D runs. From these slices, identification of surface voxels and estimation of their connectivity are performed, and 3D meshes are generated by connecting surface voxels based on their connectivity.

Lastly, texture mapping is proposed to reduce the amount of the memory required for object reconstruction. Since textures of an object tend to be homogenous and they do not change rapidly within adjacent frames, similar or same colors are repeatedly used for a certain time. Hence, all the colors used for texture mapping are clustered for a certain interval. Fig. 1 illustrates the flow chart of the proposed system.

This paper is organized as follows. Previous works on image-based object reconstruction are reviewed in Section 2. We explain 3D object representation using 3D run-length representation in Section 3, and we show how to perform geometry reconstruction using 3D run-length representation in Section 4. Surface generation and texture mapping are presented in Section 5, and in Section 6, respectively. Experimental results are shown in Section 7, and we conclude in Section 8.

2. Related works

Since 1990s, a variety of researches have been conducted to generate a 3D model of a moving object using multiple cameras. In most of the previous works, silhouettes of an object were used as a cue for geometry reconstruction [10–12,14,15,17–20,22–25,27,30,31,34]. A silhouette of a 3D object is the projection of the object onto 2D image planes, and it can be extracted from 2D images by segmentation or Chroma-keying. Silhouettes were used mostly to compute a *visual hull*, which is a 3D entity computed by the intersection of the cones generated by back-projecting each silhouette to 3D space [7]. In several works, silhouettes were also used to adjust a priori model of a human actor [19,23,34]. Meanwhile, stereo-based works which first reconstruct 2.5D depth maps for each view and generate a 3D model by combining these depth maps were proposed [9,33]. When stereo cues are used, however, cameras should be densely placed for accurate reconstruction. Recently, several works which fuses silhouette information with stereo cues were also reported [16,21,26,28,29,32,39]. By using a visual hull as an initial estimate, a more accurate 3D model was generated by optimizing the model with stereo information. Since textures of human actors tend to be homogenous, measuring the confidence of stereo information was reported [26]. Furthermore, view dependent optimization [29], and utilization of appearance discontinuities occurred at the clothing boundaries [32] were also proposed to obtain a more accurate 3D model. Lastly, object reconstruction using multiple inexpensive consumer grade depth cameras was also proposed [53]. In this paper, we propose an object reconstruction without using any depth sensors.

Image-based reconstruction methods use various primitives to represent a 3D geometry. Among them, the popular primitive was voxels because representing an object with voxels is intuitive [10,12,13,16,17,20–22,25,27,28,31]. However, voxel-based representation generally requires a large amount of memory. Hence, several works applied octree structure [12] or run-length encoding [27] to reduce memory requirement for voxel-based reconstruction. 3D Meshes were also widely used for 3D object representation, and 3D meshes were either directly computed [18,19,23,30] or generated from voxels [16,17,21,22,25,28,29]. Utilizing irregular 3D points [9,14,15,24,33] or surfels [42], and generating virtual views without explicit 3D model were also reported [11].

For geometry reconstruction, this paper computes a visual hull from multi-view silhouettes. Generally, visual hull computation is classified into volumetric approaches [10,12,16–18,20–22,25,28,31] and polygonal approaches [18,30]. Volumetric approaches use voxels as primitives, and consider a visual hull as the union of voxels whose projections are inside 2D silhouettes. Hence, each voxel is projected into image planes, and a voxel is classified as a part of a 3D object if the voxel is projected inside 2D silhouettes. Volumetric approaches are known to be intuitive and fast. However, the space discretization by regular voxels usually leads to poor precision of 3D modeling. In comparison, polygonal approaches, which utilize 3D meshes as primitives, use the contour of silhouettes rather than the regions of silhouettes. They actually generate visual cones from the contour of silhouettes, and a visual hull is generated by intersecting these cones. Compared to volumetric approaches, polygonal approaches are known to be accurate and memory efficient. However, they often suffer from numerical instabilities which lead to surface models which are incomplete or corrupted. In [55], a polygonal approach which gives additional guarantees was proposed. Even though this method provides a precise model, it is computationally expensive when a large number

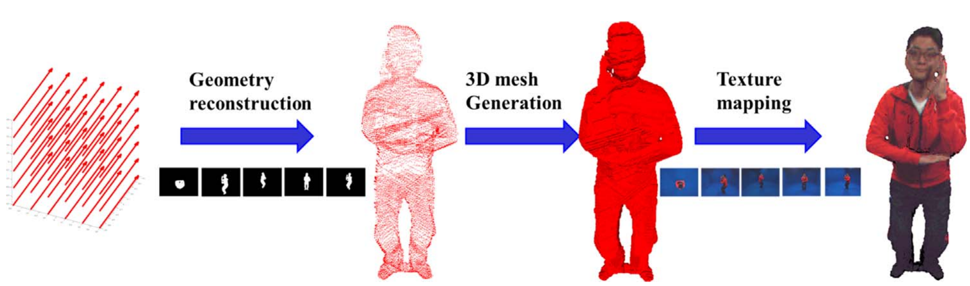

Fig. 1. The flow chart of the proposed system.