FREE VIEWPOINT VIDEO SYNTHESIS USING MULTI-VIEW DEPTH AND COLOR CAMERAS

Kazuki Matsumoto, Chiyoung Song, Francois de Sorbier & Hideo Saito

Graduate School of Science and Technology
Keio University
Yokohama, Japan

ABSTRACT

In this paper, we propose an approach for generating free viewpoint videos based on multiple depth and color cameras to resolve issues encountered with traditional color cameras techniques. Our system is based on consumer products such as Kinect that does not provide satisfying quality in terms of resolution and noise. Our contribution is then to propose a full pipeline for enhancing the depth maps and finally improving the quality of the novel viewpoint generated.

Index Terms— Free-viewpoint video, depth camera, upsampling, noise reduction, FTV

1. INTRODUCTION

Over the last years, broadcasting companies have been eager to generate embellished contents for the viewers. Sport or entertaining events can now be displayed with extra information that help the viewer to clarify the program currently watched. Now, those companies are focusing on creating more interactive contents to give the viewer freedom to decide, for example, the best viewpoint to appreciate a video.

Several researches [1] have proposed solutions for creating such interactive contents. They are mainly based on systems made of multiple calibrated color cameras. All the streams are then processed together in order to generate disparity maps or 3D models. However, such approach requires high number of cameras, making it difficult to manage (transportation, calibration, synchronisation). Moreover, the quality of the result is often below the expectations since using only the color information may not be enough to estimate the geometry of a scene.

Recently, depth cameras have become very popular because they can capture the depth information of a scene and the corresponding color image in real time. The most successful one at this moment is Microsoft Kinect based on the structured light technology. In the context of free viewpoint videos, using such depth cameras becomes very interesting since we can reduce the number of devices while increasing the whole quality of the geometry of the scene. However, devices like Kinect have several limitations. First, the resolution of the depth and color image are quite low and do not satisfy the current requirements of HD TV. Also, the depth map captured by Kinect is degraded by the presence of noise, which quickly increases if the number of devices aiming at the same scene is increased. This is caused by the interferences of the identical patterns projected by multiple devices.

In this paper, we describe a system for generating free viewpoint videos based on multiple depth and color cameras. The main contribution is the capability to enhance the depth maps to increase the quality of the result by performing an upsampling with reduced amount of noise. The addition of extra color cameras provides a direct access to high definition images that can be mapped on the geometry. The rest of paper is organized as follows: We first introduce several works related. Then, we give an overview of our capture system with a step by step description of the process. Finally, before the concluding, we present and discuss our results.

2. RELATED WORKS

Even if depth cameras are becoming popular, only few works are using it in the context of free viewpoint TV. Researches like [2] are mostly focusing on the use of dataset[1] providing the color image and the corresponding depth image.

Kuster et al. [3] however proposed to combined two depth cameras with three color cameras to perform a quality foreground reconstruction of a scene. Their main contribution is to propose a refinement in order to improve the quality of the novel viewpoint generation. Few other works like [4] can be mentioned but does not add extra improvements besides increasing the number of depth sensors.

Considering the improvement of the quality of the depth map, Yongseok et al. proposed a depth image superresolution algorithm based on RGB image segmentation [5]. The input depth image is upsampled to the same size as the input color image using a bicubic interpolation. The edges in the reconstructed high resolution depth image are refined by forcing them to match those of high resolution color image and depth values of the image are enhanced by optimizing an energy

[1] like the MSR 3D Video dataset

function based on the Markov Random Field. They, however, apply this approach only to the depth image free of noises provided by the Middlebury dataset and does not provide any information about the computational time. Xuequin et al. presented a simple pipeline to enhance the quality as well as the spatial resolution of range data in real-time with GPU implementation [6]. Moreover, they upsampled the depth information with the data from high resolution video camera and succeeded in improving the sub-pixel accuracy. But, they applied their method to time-of-flight based depth camera only. Although the resolution of the depth map from TOF depth camera is much lower than RGB image, it includes less noise compared with Kinect-like depth cameras; Consequently, it still remains difficult to efficiently upsample the depth data from depth cameras like Kinect.

In our knowledge, there still is no work focusing on FTV with depth cameras proposing to reduce the noise of the depth data and an upsampling.

3. OVERVIEW OF THE SYSTEM

In order to generate quality free viewpoint videos, we propose to use multiple pairs of depth and high-definition color cameras. We constrain our setup to an indoor environment since the range of the depth cameras is limited up to 5 meters.

Each element of the system is independent, meaning that after the calibration stage, each stream is processed and the results are combined to render a 3D reconstruction of the scene. The goal of the processing is first to reduce the noise from the depth maps, then to increase their resolution by applying an upsampling. All these stages will be described in the following sections.

4. CALIBRATION OF THE DEVICES

We are using multiple devices, which requires estimating the pose of each of them according to a common referential. We are making two assumptions: the cameras are fixed, and the origin of the referential is the position of one of the depth camera, considering that fixed cameras allows pre-computing the pose of each device without the constraint of real time. Our approach is then to use a calibration pattern to estimate the pose of color and depth cameras.

The pattern is placed in the overlapping fields of view of a pair of devices. For handling two Kinect-like depth cameras, we detect the pattern in the color image and get the 3D position for from the depth information. We then obtain a set of 3D-3D correspondences. The pose is estimated by computing the rigid transformation as described in [7] and minimized with RANSAC. In the case of a depth camera and a color camera, we process in the same manner to detect the calibration pattern and obtain 3D-2D correspondences. The pose is obtained with a Perspective-n-Point camera pose estimation and also minimized with RANSAC.

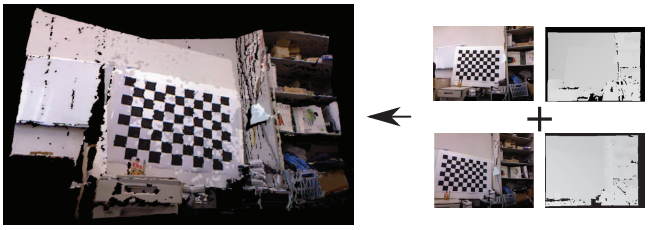


Fig. 1. Result of the calibration of two depth cameras using 3D-3D rigid transformation estimation.

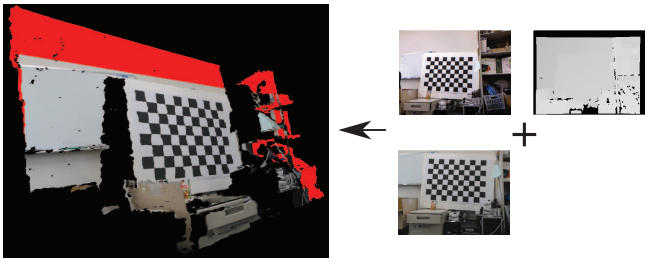


Fig. 2. Result of the calibration of a depth camera with a color camera using the Perspective-n-Point pose estimation. The red parts defined the areas not covered by the color camera.

When using two depth cameras like Kinect simultaneously, the aforementioned interferences degrade the accuracy of the depth estimation. This can be a problem when estimating the rigid transformation between two depth cameras. Since the calibration is an offline process, we suggest to detect and capture depth information at two different moments. While one camera is capturing, the infrared emitter of the other is blocked and vice versa. Two results of the calibration process are presented in Figures 1 and 2.

5. ENHANCEMENT OF THE DEPTH MAPS

As previously described, depth maps obtained with Kinect-like depth cameras have two main drawbacks: the noise, and the low resolution[2]. We then propose a processing flow that jointly reduce the noise of the depth map and upsample it to higher resolutions.

5.1. Noise Reduction

In the captured depth maps we can distinguish two kinds of noise. One is a noise produced by the average accuracy and temporal instability of the depth map estimated by the device, denoted as structural noise. The other appears when more than one depth camera is concurrently used, that we call interference noise. Our goal is then to reduce both kinds of noise in order to improve the quality of the depth map before performing the upsampling.

[2]VGA resolution for Kinect

We propose to reduce the structural noise by applying an accumulation buffer. Each new depth frame is integrated in the buffer based on weights. The data already integrated within the buffer are weighted more. However, if the difference between the previous and the incoming depth is over a given threshold, we consider that the the position of the object in the scene has changed, and replaces the corresponding older data.

The interference noise is reduced by applying a planar fitting method to the point cloud. This approach will be described in the following section since it is also used for the upsampling.

5.2. Upsampling of the depth map

The overall resolution of the depth maps needs to be increased to correspond, for example, to the higher resolution of the neighbouring color cameras. We propose an upsampling approach of the depth map based on a segmentation of the color information and assuming a locally planar geometry in each segmented cluster. All these steps are described in the following sub-sections.

5.2.1. Color image segmentation

The color image segmentation is an important stage since it defines the clusters that will be considered as planar surfaces. In the color image, we assume that a region with smaller gradient corresponds to a potential planar structure of the environment. To find such a region in an efficient manner, a GPU-implementation, with slight modification, of SLIC by Radhakrishna. et al [8] , is used because of its capability to cluster images into regions that conform to the sharp edges of the RGB image, with reasonably uniform distribution and size. Our implementation handles clusters in massively parallel manner thanks to low-cost GPU threads, resulting in throughput boost. An algorithmic modification we made from the original SLIC is that our implementation is single-pass, whereas the original is multi-pass to adjust the cluster centers and guarantee the segmentation of entire image pixels. This modification makes it possible to leave some pixels to be orphaned, but greatly improves the overall runtime of the system. A result of the segmentation is presented in Figure 3

5.2.2. Locally planar surface estimation

After segmenting the color image, we project each 3D data from the depth map onto the segmented image by using the rigid transformation obtained during the calibration stage. For each cluster, we assume a locally planar surface, meaning that the depth values within each cluster of a color image can fit a plane.

For each cluster, we then compute the plane equation. The normal associated with the cluster is estimated by getting the outer product of eigenvectors calculated from the principal component analysis(PCA) of the 3D points belonging to the cluster. Finally, the planar equation is resolved by using the average of the points inside of the cluster.

(a) Original color image (b) Result of the segmentation

Fig. 3. Result of the color based segmentation for defining the potential planar surfaces.

5.2.3. Upsampling

For upsampling the depth map, we use the projected 3D data from the previous stage. After this projection onto the segmented color image, some pixels are left missing the corresponding 3D information because of the difference of resolution. We fill those holes by applying the planar equation related to the cluster. Finally all the points are transformed back in order to obtain the upsampled version of the original depth map.

However, it is possible that some clusters don't necessarily contain planar structures, because a depth discontinuity can still occur without corresponding intensity change. We therefore perform a sanity check of the fitted plane by looking at its eigenvalue obtained by performing PCA, and if it is larger than a threshold, the cluster is thought to be non-planar and we simply apply a linear upsampling based on the original data.

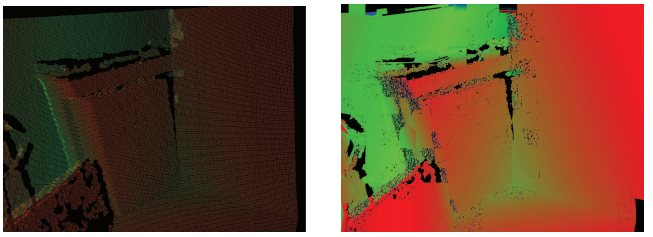
6. RENDERING OF THE FREE VIEWPOINT VIDEO

We base our rendering on OpenGL to take advantage of the capabilities of the GPU for faster performances. Each 3D data related to a depth map is transferred onto the graphic card via a specific data structure and rendered. For correctly displaying the multiple meshes, we apply the transformation computed during the calibration stage as follows: $X' = K \times M \times X$ where K is the OpenGL projection matrix, M the rigid transformation and X the input 3D point.

The color information is mapped onto the mesh with projective textures using GLSL shaders. The advantage of this is that the color is not interpolated between the vertices, but defined for each pixel. Consequently, the quality of the color information is preserved.

7. RESULTS AND DISCUSSION

For our experiments, we captured the depth information from two Kinect sensors and two PointGrey cameras with a resolution of 1280x960. The computation is done on a 3.2GHz's Pc with 32GB of memory. We also constrained the captured environment to a closed indoor scene since the range of Kinect is limited and the sensor is sensitive to sunlight.



(a) The raw depth map. (b) The upsampled depth map

Fig. 4. Result of the upsampling of the depth map.

7.1. Results

Figure 4 presents one example of result of our upsampling algorithm. We can notice that planar surfaces contain less amount of noise. However, some artefacts are present and will be discussed in the next section.

7.2. Discussion

In the different presented results we can notice some erroneously upsampled depth map, resulting from using linearly interpolated input image as a backup data for planar-fitting method.This can be avoided by using not only linear plane-fitting but also non-linear plane fittings that can represent multi-dimensional data spread. Also a possibility is to use depth-adaptive segmentation algorithm, instead of RGB-only segmentation. Doing so is expected to result in reduced number of plane fitting failure.

Another drawback of our approach is that it is not in real-time. Even if most of the parts have been coded with CUDA to take advantage of the parallelized architecture, processing multiple streams is still slow. A solution could be to dedicate a processing unit to each stream and another one for the final rendering stage.

8. CONCLUSIONS

We proposed a method for generating free viewpoint videos based on multiple depth and color cameras. We also introduced a technique for upsampling depth data for matching the resolution of TV standard. We also proposed to reduce the noise from depth data to improve the quality of the result.

For future works, we will focus not only on resolving the problems related to artefacts when creating the upsampled version of the depth map, but also on improving the overall quality of the rendering stage. We will also perform more experiments with increasing the number of devices.

8.1. Acknowledgements

This work is partially supported by National Institute of Information and Communications Technology (NICT), Japan, and also partially supported by MEXT/JSPS Grant-in-Aid for Scientific Research(S) 24220004.

9. REFERENCES

[1] Masayuki Tanimoto, "Ftv: Free-viewpoint television," *Signal Processing: Image Communication*, vol. 27, no. 6, pp. 555–570, 2012.

[2] Yuji Mori, Norishige Fukushima, Tomohiro Yendo, Toshiaki Fujii, and Masayuki Tanimoto, "View generation with 3d warping using depth information for ftv," *Signal Processing: Image Communication*, vol. 24, no. 1, pp. 65–72, 2009.

[3] Claudia Kuster, Tiberiu Popa, Christopher Zach, Craig Gotsman, Markus Gross, Peter Eisert, Joachim Hornegger, and Konrad Polthier, "Freecam: A hybrid camera system for interactive free-viewpoint video," in *Proceedings of vision, modeling, and visualization (VMV)*, 2011.

[4] Kai Ruhl, Kai Berger, Christian Lipski, Felix Klose, Yannic Schroeder, Alexander Scholz, and Marcus Magnor, "Integrating multiple depth sensors into the virtual video camera," in *ACM SIGGRAPH 2011 Posters*. ACM, 2011, p. 23.

[5] Yongseok Soh, Jae-Young Sim, Chang-Su Kim, and Sang-Uk Lee, "Superpixel-based depth image super-resolution," in *IS&T/SPIE Electronic Imaging*. International Society for Optics and Photonics, 2012, pp. 82900D–82900D.

[6] Xueqin Xiang, Guangxia Li, Jing Tong, Mingmin Zhang, and Zhigeng Pan, "Real-time spatial and depth upsampling for range data," *Transactions on computational science XII*, pp. 78–97, 2011.

[7] K Somani Arun, Thomas S Huang, and Steven D Blostein, "Least-squares fitting of two 3-d point sets," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, , no. 5, pp. 698–700, 1987.

[8] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Süsstrunk, "Slic superpixels," *École Polytechnique Fédéral de Lausssanne (EPFL), Tech. Rep*, vol. 149300, 2010.