View Generation for Free Viewpoint Video System

Gangyi JIANG¹, Liangzhong FAN², Mei YU¹, Feng Shao¹

¹ Faculty of Information Science and Engineering, Ningbo University, Ningbo, 315211, China
² Ningbo Institute of Technology, Zhejiang University, Ningbo, 315100, China

Abstract: View generation is one of the most time consuming parts at user side in free viewpoint video (FVV) system. A user side oriented framework of FVV system is presented, focus on the requirement and capability of user side. The calculation of disparity map, which is required at the user side, is moved to the server side, based on the fact that the user side is a terminal with constrained resource environment and block based disparity map can be easily obtained at the server side due to the usage of disparity compensation prediction at the MVC encoder. Based on the block based disparity map transmitted from server side, a fast pixel-wise disparity refinement scheme is designed to accelerate arbitrary view rendering by using the disparity consistency test. Experimental results show that compared with the conventional block-based matching method the new method speeds up view generation significantly and the quality of virtual view is improved greatly.

Keywords: Free viewpoint video; view generation; block based disparity map; disparity refinement; ray-space

1 Introduction

It has been recognized that multi-view video signal processing is a key technology for a wide variety of future applications, including free viewpoint video (FVV) system, three-dimensional television, immersive teleconference and surveillance etc [1,2]. FVV can offer a 3D depth impression of the observed scenery and allow interactive selection of viewpoint and direction within a certain operating range, which is regarded as the most challenging audio-video application scenario[3,4].

A classical FVV system, as shown in Fig.1, normally includes the components of multi-view capture, pre-processing, multi-view video coding (MVC) encoder, transmission, MVC decoder, view generation and display[5]. At server side, the most difficult problem is how to improve the coding efficiency of multi-view video. While at user side, the most time-consuming parts are view generation and MVC decoding. Therefore, how to reduce the complexity of these two parts will alleviate the burden of a receiver.

![Fig.1. The framework of classical FVV system](image)

Image-based rendering refers to a collection of techniques and representations that allow 3D scenes and objects to be visualized in a realistic way without full 3D model reconstruction[6,7]. For interactive photo-realistic applications, accurate pixel-wise disparity information is generally used to synthesize high quality novel views[8]. Light field, ray-space and lumigraph are similar ideas which can model radiances of a scene by a 4D plenoptic function[9-13]. A novel view from an arbitrary position and direction can be generated by appropriately combing image pixels from the existing views[4]. However, these benefits have to be paid by dense sampling of the real world with plenty of original view images. In real situation, only sparse multi-view images are compressed by MVC scheme and transmitted to the user side, and dense intermediate view images are generated by some interpolating means[5]. To interpolate dense intermediate view images, accurate disparity map, which determines the quality of synthesized image, will be calculated beforehand.

This paper is focused on arbitrary view rendering using block-based disparity map generated at server side, and it is organized as follows: Section 2 illustrates the main components of our framework. Section 3 describes a refinement scheme for block based disparity map at user side. Section 4 gives arbitrary viewpoint rendering used in our system. Section 5 reports some experimental results on different test images and Section 6 is the conclusion.

2 User Side Oriented Free Viewpoint Video System

In many applications, user side is a terminal with constrained resource environment, such as TV set,
mobile phone or PDA etc. Its capability of memory and processing is not powerful to process complicated operations. To realize free viewpoint navigation in these devices will confront with many technical problems.

This paper aims to view generation at user side with low complexity computation. Compared with the resource limitation at the user side, the complexity of server side can be left out of account. Thus, moving part of time-consuming task to the server side is an alternative scheme. Fig.2 gives the framework of our new system, in which block based disparity map is generated and encoded at the server side. At the user side, the received block based disparity map is refined for view generation, and spatial correlation in block based disparity map is utilized to accelerate the generation of pixel-wise disparity map. By using the refined disparity map, dense ray-space volume is obtained so that novel views can be rendered easily.

![Fig.2. The framework of user side oriented FVV system](image)

The new framework is based on the following considerations:

1) To improve coding efficiency, correlations among multi-view videos are usually exploited. Disparity compensation prediction algorithm is used to calculate the correspondences between adjacent views. Thus the block based disparity map can be easily obtained at MVC encoding step without any extra computational efforts.

2) For view generation at the user side, pixel-wise disparity map is needed. So if pixel-wise disparity map is generated and transmitted to the user side, disparity estimation can be avoided for the receiver. However, in this case, the total bit rate will increase obviously because of the extra pixel-wise disparity map. Taking into account the tradeoff between the total bit rate and the resolution of disparity map, 8×8 block based disparity map is adopted in our scheme. Since the size of the disparity map is only 1/64 of the original image size, and the values of disparity map is restricted in the range of maximum disparity, the increased bit rate is small compared with original multi-view coding. While in the receiver, it is necessary to refine the block based disparity map to obtain pixel-wise disparity map.

3) **Disparity Refinement Scheme**

At the server side, a disparity estimation method considering neighborhood constraints is used. Let \( I_L \) and \( I_R \) represent left image and right image respectively, \((x,y)\) denote the current block to be estimated. An energy function including data item \( E_{\text{data}} \) and smoothness item \( E_{\text{smooth}} \) is defined as

\[
E(x, y, dv) = E_{\text{data}}(x, y, dv) + \lambda E_{\text{smooth}}(x, y, dv)
\]

(1)

\[
E_{\text{data}}(x, y, dv) = \sum_{i=-w}^{w} \sum_{j=-w}^{w} | I_L(x+j, y+i) - I_R(x+dv+j, y+i) |
\]

(2)

\[
E_{\text{smooth}} = \sum_{i=0}^{3} |dv_i - dv_j|
\]

(3)

The smoothness item utilizes neighboring disparity vectors to enhance smoothness of disparity map. The relationship between \( dv \) and \( dv_0 \sim dv_3 \) is illustrated in Fig.3. Factor \( \lambda \) is a constant which controls the influence of the smoothness item on the energy function, the higher value of \( \lambda \) leads to the smoother disparity map.

Fig.4(a) and (b) give the original two views of Akko&Kayo, while Fig.4(c) and (d) show two corresponding disparity maps with block size of 8×8. Fig.4(c) is the result without smooth constraint, while Fig.4(d) is obtained by the above disparity estimation method considering neighborhood constraints.

![Fig.3. Smoothness constraints between disparity vectors](image)

(a) original left view (b) original right view

(c) without smooth constraint (d) with smooth constraint

![Fig.4. Results of block based disparity maps](image)
method can achieve much smoother disparity in the regions belonging to the same object compared with the conventional method.

At the user side, pixel-wise disparity map is necessary for view generation. Based on the block-based disparity map received from the server side, fast disparity refinement scheme can be designed. Similar to normal 2D image, disparity map has strong spatial correlation. If disparity estimation is accurate, objects in a scene with same depth will have same value of disparity. To reduce complexity of disparity estimation, some blocks with accurate disparities are extracted directly from disparity map by using this smoothness assumption so that the re-estimation of these blocks can be avoided.

The disparity refinement algorithm is carried out as the following steps:

Step1. In the disparity map with the resolution of \(8 \times 8\), if disparity \(d_v\) of current block satisfies Eq.(4), where \(d_v(i \in [0, 7])\) is disparity of eight neighborhood of current block, the current disparity \(d_v\) is regarded as an accurate one, so \(d_v\) will be copied to each pixel within the corresponding block of pixel-wise disparity map. Otherwise, the current \(8 \times 8\) block will be split into four smaller blocks with the size of \(4 \times 4\), and they are marked as inaccurate blocks.

\[
f_{8 \times 8}(x, y) = \begin{cases} 
1, & \text{if } \forall |(d_v - d_v(i))| < 2, i \in [0,7] \\
0, & \text{otherwise}
\end{cases} \quad (4)
\]

Step2. In the disparity map with resolution of \(4 \times 4\), for those inaccurate blocks marked in step1, the disparities are recalculated by the above disparity estimation method considering neighborhood constraints. If \(d_v\) of current block satisfies Eq.(5), where \(d_v(i \in [0, 3])\) is disparity of four neighborhood of \(4 \times 4\) block with respect to \(d_v\), the current disparity \(d_v\) is regarded to be accurate, and it will be copied to the corresponding position of pixel-wise disparity map. Otherwise, they are still marked as inaccurate blocks.

\[
f_{4 \times 4}(x, y) = \begin{cases} 
1, & \text{if } \forall |(d_v - d_v(i))| < 2, i \in [0,3] \\
0, & \text{otherwise}
\end{cases} \quad (5)
\]

Step3. In the pixel-wise disparity map, \(4 \times 4\) disparity vector is applied to accelerate the refinement of these inaccurate regions. Let \(d_v\) denote current disparity of an inaccurate block and it is used as initial position of searching process. A small range for searching the optimal disparity is defined as \([d_v-4, d_v+4]\). Therefore, for position that has initial block disparity, the computation complexity of disparity estimation is decreased greatly.

Fig.5(a) gives disparity map after \(8 \times 8\) block based consistency test. The white areas indicate the inaccurate blocks, while the gray regions are copied directly from the \(8 \times 8\) disparity map. As we can see, the accurate blocks are either background of a scene or at middle part of foreground objects. Fig.5(b) shows the result of \(4 \times 4\) block based disparity consistency test, in which the white areas indicate the accurate blocks while the grey ones need to be verified by the disparity estimation method considering neighborhood constraints. Fig.5(c) gives the verified \(4 \times 4\) block based disparity map, the white pixels are those inaccurate blocks even after the verification.

Fig.6 gives two pixel-wise disparity maps. Fig.6(a) is generated by the conventional block matching method, while Fig.6(b) is acquired by the proposed refinement method. Compared with the block matching method, the proposed method can save time more than 26 times. Moreover, disparities at smooth regions are more accurate than that of conventional method.
4 View Generation Scheme

Intermediate view interpolation is vital for IBR (Image Based Rendering) based FVV system, since it determines the quality of synthesized virtual view. When the correspondences between view \( n \) and view \( n+1 \) are obtained, the dense intermediate disparity map at \( a \) position can be generated through the following steps.

1. For the pixel at position \( x \) in the view \( n \) with disparity \( d \), the position of the corresponding pixel in the intermediate view can be obtained by simply scaling the disparity according to \( a \) which indicates the location of the intermediate view between view \( n \) and view \( n+1 \). Disparity \( d \) is assigned to intermediate view pixel at position \( x + ad \). In this case, the values of the pixels between view \( n \) and view \( n+1 \) are interpolated in term of \( a \). This is illustrated in Fig.7 by the pair of arrows which are opposite with each other.

2. The intermediate disparity map obtained by the above step will have some holes, as shown in Fig.8, in which the holes are indicated by white pixels. For the holes in corresponding intermediate view, the pixels within the holes are regarded as occluded pixels. These pixels can only be seen in one view, and a simple strategy is introduced to estimate their disparities. As we known, the disparities of background objects are normally smaller than that of the foreground objects, and in general, background objects may be occluded by foreground objects. Therefore, for the occluded pixels, the nearest matched pixels at the left and right direction are compared. Let \( d_L \) and \( d_R \) denote the corresponding disparities with respect to the left and right directions, then the disparity \( d \) of current occluded pixel is determined by

\[
d = \begin{cases} 
   d_R, \text{and left occluded,} & \text{if } d_L \geq d_R \\
   d_L, \text{and right occluded,} & \text{otherwise}
\end{cases}
\] (6)

In this case, the pixels are interpolated according to occlusion type. If current pixel is labeled as left occluded pixel, that is, \( d_L \geq d_R \), the pixel at the view \( n \) indicated by the disparity \( d_R \) is directly copied to the interpolating pixel. The similar operation can be done to the right occluded pixel.

Ray-space representation describes the rays in a scene as a 4D function \( f(x, y, \theta, \phi) \), where \( (\theta, \phi) \) denotes the direction of the ray, \( (x, y) \) denotes the intersection of the ray and the reference plane, and \( f(x, y, \theta, \phi) \) represents the intensity of the specific ray. An important feature in ray-space is that an image with respect to a certain viewpoint is given as a sub-space of ray-space. Fig.9 shows the relationship between real space and ray-space. Intermediate views with respect to virtual cameras, denoted as the gray ones in the figure, are required to be rendered. The relationship between the virtual view and the ray-space can be explained as

\[
X = x + z \tan \theta
\] (7)

Therefore, when dense sampling of a scene is acquired, it is comparatively easy to synthesize a novel view.
5 Experimental Results

Experiments are performed to evaluate the efficiency of the proposed algorithm. The test image Xmas with 101 parallel views is used to evaluate intermediate view interpolation without camera parameter. The 3DTV test data Akko&Kayo is used to render arbitrary view image. Akko&Kayo is captured by 3×5 camera array, here, 5 images in the middle row are selected, the maximum disparity between two anchor images is 20 pixels.

Fig.10 shows PSNR and running time of interpolating intermediate view with different camera intervals. In the figures, the proposed method outperforms BMI(Block-based Matching Interpolation) method[14], because it can obtain more accurate disparities. Moreover, the proposed method runs much faster than BMI because it has initial disparity for searching.

For rendering virtual viewpoint image, as illustrated in Fig.11, dense intermediate viewpoint images are firstly interpolated by the proposed method. Here, 19 intermediate viewpoint images are interpolated between two anchor view images according to the sampling theory. Then, these dense multi-view images are projected into ray-space. When given a virtual viewpoint position, the corresponding data can be extracted from the interpolated dense ray-space. Fig.11(a) shows a virtual view at the baseline, while Fig.11(b) and Fig.11(c) show the virtual views at forward and backward positions from the baseline respectively. As we move forward, details of a scene can be seen, meanwhile, the field of view becomes smaller. On the contrary, when we move backward, the field of view becomes larger. The emergence of black regions at top and bottom in Fig.11(c) is resulted from the fact that dense intermediate views are interpolated in horizontal direction but not vertical direction. From the rendered virtual view images, it is seen that the synthesized images have photo-reality without ghosting artifacts.

![PSNR of interpolated intermediate view](image1)

![Run time of interpolating intermediate view](image2)

(a) PSNR of interpolated intermediate view  
(b) run time of interpolating intermediate view

![Original intermediate image](image3) ![BMI](image4) ![8x8 Proposed method](image5)

(c)original intermediate image  
(d)BMI  
(e) 8x8 Proposed method  

(PSNR=28.36dB, TIME=32.8second)  
(PSNR=36.17dB, TIME=1.1second)

Fig.10. Result of intermediate view generation of Xmas (maximum disparity=50)
In this paper, a user side oriented framework of free viewpoint video system is presented, which focus on the requirement and capability of user side. The calculation of disparity map, which is required at the user side, is moved to the server side, based on the fact that the user side is a terminal with constrained resource environment and block based disparity map can be easily obtained at the server side due to the usage of disparity compensation prediction at the MVC encoder. This change will alleviate the burden of view synthesis and it is more suitable for real time applications.

Acknowledgement
This work was supported by the Natural Science Foundation of China (grant 60472100, 60672073), the Program for New Century Excellent Talents in University (NCET-06-0537), Natural Science Foundation of Ningbo (grant 2007A610037).

References: